Nicolas Melo de Oliveira

ABORDAGENS QUÂNTICAS PARA P *VERSUS* NP E SIMULAÇÕES SIMBÓLICAS

Trabalho de Conclusão de Curso

UFRPE

Universidade Federal Rural de Pernambuco
secretaria@preg.ufrpe.br
http://www.ufrpe.br/br/graduacao

RECIFE
2015

Universidade Federal Rural de Pernambuco
Departamento de Estatística e Informática
Bacharelado em Ciência da Computação

Nicolas Melo de Oliveira

ABORDAGENS QUÂNTICAS PARA P *VERSUS* NP E SIMULAÇÕES SIMBÓLICAS

Trabalho de Conclusão de Curso apresentado ao Programa de Bacharelado em Ciência da Computação do Departamento de Estatística e Informática da Universidade Federal Rural de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: *Wilson Rosa de Oliveira Júnior*
Co-Orientador: *Adenilton José da Silva*

RECIFE
2015

MINISTÉRIO DA EDUCAÇÃO E DO DESPORTO
UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO (UFRPE)
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

http://www.bcc.ufrpe.br

FICHA DE APROVAÇÃO DO TRABALHO DE CONCLUSÃO DE CURSO

Trabalho defendido por Nicolas Melo de Oliveira como requisito para conclusão do curso de Bacharelado em Ciência da Computação da Universidade Federal Rural de Pernambuco, intitulado Abordagens Quânticas para P versus NP e Simulações Simbólicas, orientado pelo Prof. Wilson Rosa de Oliveira Júnior e aprovado pela seguinte banca examinadora:

Wilson Rosa de Oliveira Júnior
DEINFO/UFRPE

Adenilton José da Silva
DEINFO/UFRPE

Tiago Alessandro Espinola Ferreira
DEINFO/UFRPE

Dedico este trabalho a todos que, de certa forma, tornaram possível que eu concluísse esta etapa.

Agradecimentos

Agradeço, primeiramente, ao meu professor e orientador Wilson Rosa, pelos anos de ensino e pesquisa, pela dedicação e paixão à ciência e aos estudos e, principalmente, pela paciência e suporte que me fizeram decidir pela carreira acadêmica.

Agradeço, também, aos demais membros do grupo de pesquisa em computação quântica da UFRPE: Adenilton Silva - meu co-orientador, aos professores Tiago Ferreira e Cláudio Cristino, e aos colegas Maigan Alcântara, Vítor Torreão e Fernando Neto (CIn - UFPE).

Aos professores que me acompanharam e me orientaram nas disciplinas de Projeto de Conclusão de Curso e Trabalho de Conclusão de Curso, Obionor Nóbrega e Francielle Santos, respectivamente, que foram bastante atenciosos ao sanar minhas dúvidas para que este trabalho pudesse ter um resultado satisfatório.

À FACEPE (Fundação de Amparo à Ciência e Tecnologia de Pernambuco), que me forneceu apoio financeiro durante boa parte da graduação através de bolsas de Iniciação Científica.

Aos amigos Elmiro, Diana, Clara e Júnior, que estão e sei que sempre estarão junto a mim, pela companhia e amizade que me fizeram sentir mais leve durante os últimos meses.

Ao professor Fábio Hazin (Departamento de Pesca - UFRPE), que, apesar de não me prover conhecimento acadêmico, proporcionou uma brilhante e inspiradora iniciação ao mundo da meditação e do budismo, os quais foram essenciais para a luz, tranquilidade e paz no meu caminho durante a escrita deste trabalho.

Por fim, gostaria de agradecer a todos os outros docentes, discentes e funcionários (destes últimos, em especial, à secretária da coordenação do Bacharelado em Ciência da Computação, Sandra Xavier) que fazem parte do Departamento de Estatística e Informática da Universidade Federal Rural de Pernambuco, por terem feito parte, direta ou indiretamente, dessa minha caminhada.

I think I can safely say that nobody understands quantum mechanics. If you think you understand quantum mechanics, you don't understand quantum mechanics.

—RICHARD FEYNMAN

Resumo

Após espetaculares avanços na forma clássica de realizar computação, especula-se que, talvez, esse modo de computar esteja chegando aos seus limites de capacidade e desenvolvimento. Foi nesse contexto que pesquisadores e cientistas começaram a considerar de modo determinante o uso da Computação Quântica, um paradigma de computação não-convencional. Computar algoritmos em níveis quânticos tem se mostrado aparentemente mais rápido quando comparado aos análogos clássicos. Com isso, o estudo de um dos mais conhecidos problemas em aberto, o da questão $P \times NP$, tem ressurgido na versão quântica $BQP \times QMA$. Além disto, mesmo sem ainda uma prova concreta e definitiva, conjectura-se que os computadores quânticos são, de fato, mais rápidos e conseguem resolver problemas NP-completos para máquinas de Turing determinísticas em tempo polinomial.

Pesquisas tem sido feitas com relação aos problemas da classe NP, porém, em todos os algoritmos clássicos conhecidos, existe apenas um tempo de execução de ordem exponencial com relação ao tamanho da entrada. Alguns desses problemas são: problema do caixeiro viajante, o problema da mochila e o problema da satisfatibilidade (problema abordado no presente trabalho).

Mesmo que tais problemas se configurem como sendo de ordem exponencial, o estudo de soluções alternativas ainda é necessário. Alguns artigos apresentam possíveis soluções quânticas para estes problemas, as quais se baseiam em circuitos quânticos, dinâmica caótica e computação adiabática.

Aqui, neste trabalho, o problema da satisfatibilidade é tratado sob a ótica da Computação Quântica, juntamente com o uso da programação simbólica e elementos quânticos presentes no SymPy, um CAS (*Computer Algebra System*) que compõe a biblioteca para matemática/física simbólica integrada à linguagem de programação Python. A presente pesquisa visa contribuir para o estudo da computação quântica através da implementação de simuladores de soluções que se propõem a resolver os problemas NP-completos em tempo polinomial.

Palavras-chave: computação quântica, problemas np-completos, programação simbólica, python, sympy

Abstract

After spectacular advances in the classical way of realising computing, it is speculated that perhaps this mode of computing is reaching its limits and development capacity. In this context researchers and scientists began to consider in a decisive way the use of quantum computing, a paradigm of unconventional computing. Computing algorithms in quantum levels have been shown to be apparently faster compared to classical analogues. Thus, the study of one of the widely known open problem, the $P \times NP$ problem, has regained interest in its quantum version $BQP \times QMA$. Besides, in spit of not having a concrete formal proof, it is conjectured that quantum computers are indeed faster and are able to solve NP-complete problems for deterministic Turing machine in polynomial time.

Research has been made regarding the problems of NP class, but in all known classical algorithms, there is only an exponential order execution time with respect to the size of the input. Some of these problems are: traveling salesman problem, the knapsack problem and the problem of satisfiability - SAT - (which is the problem to be addressed in this paper).

Even if such problems are configured as being of exponential order, the study of alternative solutions is still needed. Some articles have quantum possible solutions to these problems, which are based on quantum circuits, chaotic dynamics and adiabatic computation.

Here, in this work, the SAT problem is treated from the perspective of Quantum Computing, along with the use of symbolic programming and quantum elements in SymPy, a CAS (Computer Algebra System) that makes up the library for symbolic math/physics integrated to programming language Python. This research aims to contribute to the study of quantum computing by implementing simulation solutions that purport to solve NP-complete problems in polinomial time.

Keywords: quantum computing, *np*-complete problems, symbolic programming, python, sympy

Lista de Figuras

Lista de Quadros

Lista de Pseudocódigos

Lista de Acrônimos

3-SAT	3-Satisfatibilidade
BPP	*Bounded-error Probabilistic Polynomial time*
BQP	*Bounded-error Quantum Polynomial Time*
CAS	*Computer Algebra System*
CQ	Computação Quântica
FNC	Forma Normal Conjuntiva
IQ	Informação Quântica
MA	*Arthur-Merlin Complexity Class*
MQ	Mecânica Quântica
NP	*Non-Deterministic Polynomial Time*
P	*Polynomial Time*
PP	*Probabilistic Polynomial Time*
QCMA	*Quantum Classical Arthur-Merlin Complexity Class*
QMA	*Quantum Arthur-Merlin Complexity Class*
QSAT	*Quantum Satisfiability*
QTF	Transformada de Fourier Quântica
SAT	Satisfatibilidade

Sumário

1 Introdução

1.1 Motivação

Após grandes avanços na forma clássica de realizar a computação em computadores digitais clássicos, tais como o advento e popularização de processadores multi-core e o alto poder computacional presente em unidades de processamento gráfico (GPUs), vislumbra-se um cenário onde não mais será possível transpor as barreiras de capacidade e desenvolvimento dessa forma de computar. Uma destas barreiras é o tamanho dos componentes que formam os chips modernos que em breve chegará a escalas subatômicas. Tal constatação é derivada da *Lei de Moore* (MOORE, 1998). É nesse contexto que surge o interesse pela Computação Quântica (CQ).

Com isso, devido aos importantes resultados obtidos por estudiosos como, por exemplo, o algoritmo de fatoração em números primos em tempo polinomial proposto por Shor (SHOR, 1994) e o algoritmo de busca proposto por Grover (GROVER, 1996), que operam mais rápido do que qualquer correspondente clássico conhecido, a computação quântica vem se tornando um paradigma cada vez mais real e desejável.

A Mecânica Quântica (MQ) é um conjunto de regras matemáticas que servem para a construção de teorias físicas. Desde a sua criação ela tem sido aplicada em diversos ramos, desde a física de partículas, física atômica e molecular, na astrofísica e na matéria condensada. Neste ambiente se desenvolveu a computação quântica, uma proposta de aplicação prática da MQ para a realização de operações e cálculos computacionais.

Neste sentido, a computação quântica surge como um campo de estudo que apresenta grande potencial para solução eficiente de problemas. Entre suas peculiaridades, uma se destaca: a superposição de estados (que implica no paralelismo quântico). Esta permite que se opere sobre uma quantidade exponencial de padrões de qubits (bits quânticos) com relação à forma clássica de computar. Assim, é possível computar, em um único passo em paralelo, todos os padrões de bits clássicos que podem ser atribuídos a uma determinada instância de um problema (MERMIN, 2003). A aplicação da computação quântica em problemas para os quais não se conhece soluções algorítmicas determinísticas em tempo polinomial e que candidatos a soluções

podem ser verificados em tempo polinomial (*Non-Deterministic Polynomial Time* (NP)) tem despertado o interesse de alguns pesquisadores.

Para uma discussão sobre complexidade computacional em computação quântica, pode-se consultar o texto contido em (CLEVE, 1999), o qual provê definições que tratam da complexidade computacional, complexidade de consulta e complexidade de comunicação com respeito à informação quântica, correlacionando estes três cenários e seus algoritmos conhecidos. Características matemáticas da computação quântica e da teoria da informação quântica encontram-se resumidos em (OHYA; VOLOVICH, 2011).

O problema da mochila, o problema do caixeiro viajante, o problema de programação inteira, o problema do isomorfismo de subgrafo e o problema satisfatibilidade (LUCAS, 2014) têm sido estudados há décadas e para os quais todos os algoritmos clássicos conhecidos tem tempo de execução exponencial com relação ao comprimento da entrada. A classe de problemas NP é a classe de todos os problemas de decisão que, sob esquemas de codificação razoáveis, podem ser resolvidos por algoritmos não-determinísticos de tempo polinomial (GAREY; JOHNSON, 1979).

O problema NP-completo escolhido para ser estudado é o da Satisfatibilidade (SAT) de uma fórmula booleana na sua Forma Normal Conjuntiva (FNC). Tal escolha deve-se a dois motivos: primeiramente, o SAT é amplamente explorado tanto em abordagens clássicas como quânticas, o que propicia o acesso a uma grande quantidade de material que serve de base para seu estudo. Em segundo lugar, observa-se o fato de haver redutibilidade entre os problemas NP-completos e os outros problemas da classe NP. Ou seja, uma vez encontrada uma solução polinomial para o problema da satisfatibilidade de fórmula booleana, todos os outros problemas NP poderão ser resolvidos desta mesma maneira com base no resultado obtido.

Neste contexto, considera-se que a CQ (baseada em circuitos, em conjunto com a dinâmica caótica, não-linearidade e através da computação adiabática) exerce uma importante função no estudo de possíveis soluções eficientes para os problemas NP-completos. Portanto, visando explorar os princípios físicos e conceitos envolvidos, alguns dos vários métodos utilizados como proposta de solução para estes problemas serão estudados e alguns destes serão implementados simbolicamente.

Uma vez que os fenômenos quânticos podem ser contra intuitivos (SILVERMAN, 2008), torna-se útil a existência de um software capaz de simulá-los em um computador clássico. Este cenário se concretiza com o uso do SymPy [1], um sistema de álgebra computacional (*Computer Algebra System* (CAS)) presente na linguagem de programação Python [2] que modela e abstrai a estrutura dos operadores e elementos quânticos tornando-os de natureza simbólica. Neste trabalho, será empregada a sintaxe simples e limpa do Python em conjunto com funções, estruturas e classes do SymPy, que caracterizam as peculiaridades da mecânica quântica (presentes no

[1] http://www.sympy.org/en/index.html
[2] https://www.python.org/

pacote para física quântica, *Quantum* [3]). Tal uso resultará em uma representação simbólica de circuitos capazes de computar uma instância do SAT e os operadores responsáveis por concluir sua satisfatibilidade.

1.2 Definição do problema

O problema de pesquisa aqui abordado centra-se em investigar qual o papel desempenhado pela Computação Quântica no que se refere à concepção de soluções que visam resolver os problemas da classe NP-completo em tempo polinomial. A partir disto, o foco torna-se explorar os elementos quânticos presentes no SymPy/Python para a construção de simuladores que visam facilitar o estudo e compreensão de tais soluções.

Importante salientar que o pressuposto teórico desta pesquisa, bem como dos trabalhos que a fundamentaram, consiste nas suposições de que é possível implementar fisicamente o modelo de Computação Quântica de circuitos e que existem operadores quânticos não-lineares, por exemplo.

1.3 Objetivos

1.3.1 Geral

O objetivo geral deste trabalho consiste na investigação de alguns modelos computacionais quânticos que se propõem a resolver a questão *P versus NP* através de circuitos quânticos, bem como na implementação de simuladores simbólicos dos mesmos fazendo uso da biblioteca SymPy presente na linguagem de programação Python.

1.3.2 Específicos

Como objetivos específicos para este projeto, tem-se os seguintes:

- Compreender a formulação de problemas NP-completos;
- Analisar técnicas e modelos quânticos existentes que visam resolver os problemas NP-completos em tempo polinomial;
- Desenvolvimento de simuladores juntamente com o uso da biblioteca simbólica da linguagem de programação Python, o SymPy;
- Examinar os resultados da pesquisa de problemas NP-completos com relação aos modelos quânticos estudados.

[3] http://docs.sympy.org/0.7.6/modules/physics/quantum/index.html

1.4 Metodologia

Pelo fato de o presente trabalho estar contido em um âmbito multidisciplinar envolvendo as áreas de Física, Matemática e Computação, tornou-se imprescindível uma revisão bibliográfica aprofundada a partir de títulos já estabelecidos na literatura. Definições fundamentais sobre computação quântica foram retiradas, essencialmente, dos textos presentes em (YANOFSKY; MANNUCCI, 2008), (MERMIN, 2007), (NIELSEN; CHUANG, 2011), (MCMAHON, 2007) e (JAEGER, 2007). Por sua vez, a formulação da classe de problemas NP-completos, bem como alguns exemplos destes, foram extraídos dos textos contidos em (GAREY; JOHNSON, 1979).

A partir da revisão bibliográfica dos trabalhos relacionados, foram listadas e exploradas de maneira conceitual as abordagens quânticas mais bem estabelecidas que se propõem a resolver problemas NP-completos em tempo polinomial. Tais abordagens incluem, principalmente, o uso de operadores usuais em circuitos quânticos, dinâmica caótica, mecânica quântica não-linear e computação quântica adiabática, e estão enunciadas em (OHYA; MASUDA, 1998), (OHYA; VOLOVICH, 1999), (LEPORATI; FELLONI, 2007), (ABRAMS; LLOYD, 1998) e (FARHI et al., 2001).

Em um segundo momento, foram examinados e avaliados teorias/conceitos de alguns dos diversos modelos de CQ propostos para resolver a questão *P versus NP*. Após essa etapa, simulações simbólicas dos algoritmos foram implementadas com o auxílio da biblioteca simbólica SymPy, da linguagem de programação Python. Para isso, utilizou-se as estruturas e abstrações algébricas e quânticas presentes no SymPy, em especial as contidas no pacote de física quântica *Quantum*. Este possui as noções de espaço, operadores e circuitos quânticos necessários para o desenvolvimento dos simuladores. Ressalto aqui o uso do ambiente de desenvolvimento para Python, chamado *Canopy*[4]. O mesmo foi escolhido por ser um ambiente que provê todas as ferramentas necessárias para o uso científico do Python.

Por fim, em um primeiro instante, foi implementada a solução contida em (OHYA; VOLOVICH, 1999), devido à sua natureza mais próxima à computação clássica (computação via circuitos), juntamente com a utilização da dinâmica caótica acoplada ao computador quântico. Em seguida, este primeiro simulador foi validado com base em testes executados sob várias instâncias do problema da satisfatibilidade a fim de certificar que tal implementação foi, de fato, construída de forma a ter capacidade de analisar e construir o circuito quântico para qualquer fórmula booleana configurada como uma instância do SAT. Posteriormente, foi desenvolvido o segundo simulador, que teve como base o trabalho exposto em (LEPORATI; FELLONI, 2007), o qual faz uso de portas lógicas quânticas Fredkin para a construção do circuito. Novamente, este foi validado com respeito a várias instâncias do problema (neste caso, o 3-Satisfatibilidade (3-SAT)).

Como resultado deste trabalho, são expostos no Capítulo 5 os pseudocódigos das implementações realizadas e no Capítulo 6 diagramas e esquemas dos circuitos derivados das

[4] https://www.enthought.com/products/canopy/

simulações.

1.5 Estrutura do trabalho

Este trabalho está organizado de maneira a apresentar todos os conceitos referentes ao problema em questão para, então, abordar os trabalhos diretamente relacionados e a solução do que foi proposto nos objetivos.

Dessa forma, no Capítulo 2 estão expostos: os conceitos imprescindíveis ao entendimento da Computação Quântica bem como uma revisão bibliográfica do material de apoio utilizado (Seção 2.1), uma visão ampla da formulação dos problemas NP-completos e seu *status* atual (Seção 2.2) e, por fim, segue-se uma apresentação sobre a linguagem de programação Python e sua biblioteca para computação simbólica, o SymPy, bem como a motivação para seu uso, na Seção 2.3. O Capítulo 3 versa sobre os trabalhos diretamente relacionados com a pesquisa aqui proposta, tanto no que diz respeito às soluções quânticas existentes que visam elucidar a questão *P versus NP* quanto no que se refere ao uso de simuladores quânticos em geral e do SymPy para a simulação das peculiaridades presentes na CQ. As soluções escolhidas para serem aprofundadas e servirem de base para os simuladores encontram-se descritas no Capítulo 4, o qual apresenta um resumo dos artigos que propõem tais abordagens. A descrição do desenvolvimento e características dos simuladores, bem como o resultado das simulações feitas são apresentadas no Capítulo 5 e Capítulo 6, respectivamente. Por fim, tem-se os comentários conclusivos deste trabalho no Capítulo 7.

2

Conceitos e Referenciais Teóricos

Objetivando fornecer uma fundamentação teórica suficiente para a compreensão da pesquisa, bem como visando ressaltar os principais aspectos de relevância para este trabalho, este capítulo fornece uma revisão dos textos fundamentais nas áreas de interesse as quais esta monografia está inserida.

Aqui serão abordados os conceitos básicos (e imprescindíveis para a assimilação do conteúdo dos demais capítulos) que permeiam a CQ, os problemas NP-completos e a justificativa do uso do SymPy para o desenvolvimento dos simuladores aqui propostos.

2.1 Computação Quântica

2.1.1 Revisão bibliográfica

Para um estudo sobre os recursos matemáticos inerentes à informação e computação quântica, sugere-se a leitura do texto apresentado por (OHYA; VOLOVICH, 2011). Seu conteúdo engloba desde aspectos relativos às bases da probabilidade clássica e quântica, passando pelos fundamentos da teoria da comunicação quântica, entropia, emaranhamento e algoritmos quânticos, até culminar em aplicações para os problemas NP-completos, criptografia e teleporte quânticos.

Em (SIMON, 1997) os aspectos relacionados ao poder computacional da computação quântica são discutidos em termos análogos às máquinas de Turing probabilísticas. Aqui, as teorias que embasam a probabilidade clássica dão suporte para a definição de máquinas de Turing quânticas. Através de um exemplo de algoritmo que avalia a invariância de uma função, é discutida a ordem de dificuldade para computar tal problema. Também neste trabalho, tem-se uma demonstração de que computadores quânticos são mais eficientes que os computadores clássicos. Em particular, Shor (SHOR, 1994) fornece um algoritmo quântico de tempo polinomial para o problema de fatoração. No entanto, não se sabe se este problema é NP-completo. Além deste algoritmo de fatoração, o poder computacional dos computadores quânticos tem sido explorado em uma série de trabalhos, como em (DEUTSCH; JOZSA, 1992) e (GROVER, 1996).

Existem diversas abordagens para o estudo de computação e informação quântica. Em (MERMIN, 2003) é utilizada a analogia entre bits clássicos e quânticos para, com isso, explanar sobre computação quântica para cientistas da computação. Com a mesma abordagem (MERMIN, 2007) aprofunda conceitos básicos da computação quântica, como algoritmos, criptografia, correção de erros e protocolos de comunicação, e classifica o computador quântico como aquele cuja operação explora certas transformações especiais de seu estado interno, estas, garantidas pelas leis da MQ.

Em (JAEGER, 2007) denomina-se Informação Quântica (IQ) o transporte, armazenamento e processamento de informação utilizando sistemas de MQ. Neste livro é dado um maior foco à teoria da informação quântica - a qual se configura como uma junção entre a mecânica quântica e a teoria da informação. Nele, são tratados temas como: correção de erro, entropia, criptografia, compressão de dados, operações quânticas, emaranhamento, protocolos quânticos, comunicação (clássica e quântica), algoritmos quânticos e teleporte quântico.

Na mesma linha do livro anterior, (NIELSEN; CHUANG, 2011) apresenta um material com um conteúdo abrangente sobre IQ. Entretanto, antes disso, são apontados diversos conceitos sobre MQ, CQ e ciência da computação, abordando, inclusive, perspectivas de realizações físicas para os computadores quânticos.

Outros livros-texto sobre CQ são: (MCMAHON, 2007), que, além dos conceitos usuais da computação quântica, fornece um material que aborda aplicações do emaranhamento, teoria da informação quântica e o modelo de computação quântica adiabática; outro ponto que deve ser ressaltado sobre este livro diz respeito à grande quantidade de exemplos e exercícios presentes em todos os capítulos, o que o torna uma boa alternativa para o ensino/aprendizado inicial da CQ; já em (YANOFSKY; MANNUCCI, 2008) tem-se uma abordagem totalmente voltada para cientistas da computação, onde não é esperado que o leitor tenha conhecimentos avançados prévios sobre matemática ou física; aqui, também são encontrados diversos exemplos ilustrando os conceitos expostos, bem como uma discussão sobre linguagens de programação para CQ.

Em (DEUTSCH; JOZSA, 1992), (GROVER, 1996) e (SHOR, 1994) são apresentados três dos algoritmos mais conhecidos da computação quântica. Deutsch, além de descrever o primeiro computador quântico universal, também desenvolveu o "*algoritmo de Deutsch*" (DEUTSCH, 1985), que consiste em avaliar se uma dada função é contínua ou balanceada. Apesar de uma limitada aplicação técnica, exibe uma amostra de como os computadores quânticos podem ser exponencialmente mais eficientes em comparação aos computadores clássicos. Em (DEUTSCH; JOZSA, 1992) tem-se uma generalização do algoritmo de Deutsch. Por sua vez, o algoritmo de Grover (GROVER, 1996) representa uma ganho quadrático quando comparado ao análogo clássico na busca por elementos em uma lista desordenada. Já (SHOR, 1994) mostra em seu trabalho a descoberta de um algoritmo quântico em tempo polinomial para fatoração de números inteiros grandes em seus fatores primos. Tais explorações do poder computacional quântico possibilitaram o surgimento de estudos que se propuseram a resolver outros problemas computacionais eficazmente, como as propostas de soluções eficientes para os problemas NP-

completos.

Com relação ao estudo da complexidade computacional e de comunicação quântica, (CLEVE, 1999) elaborou uma revisão introdutória baseada na complexidade de circuitos clássicos e quânticos. Através de uma série de teoremas, a ordem da complexidade computacional de alguns problemas, como o da fatoração e o da satisfatibilidade, é dada. Além disso, outros dois tópicos relacionados a complexidade são discutidos: complexidade de consulta, por meio dos algoritmos de Deutsch e de Simon; e complexidade de comunicação, através de um protocolo de comunicação entre duas entidades e dos problemas do *produto interno* e da *intersecção*. Ambos, complexidade de consulta e de comunicação, expostos através da prova de teoremas relacionados com os problemas abordados.

Em (SILVERMAN, 2008) são abordados princípios quânticos que surgem a partir da superposição de estados quânticos e que, em alguns aspectos, torna-se algo contra-intuitivo. Características como coerência, emaranhamento e interferência são discutidas com base em alguns fenômenos quânticos, tais como: a natureza ondulatória da propagação de partículas, efeitos topológicos de campos magnéticos e interações não-locais de partículas correlacionadas.

2.1.2 Definições

Aqui será exposta uma revisão dos conceitos que permeiam a computação quântica e que são imprescindíveis para um entendimento básico do assunto. Para uma abordagem completa e mais detalhada, sugere-se os textos contidos em (YANOFSKY; MANNUCCI, 2008), (MERMIN, 2007), (NIELSEN; CHUANG, 2011), (MCMAHON, 2007) e (JAEGER, 2007). Tais conteúdos foram usados como suporte para as definições que seguem.

Um computador quântico consiste numa máquina capaz de executar cálculos e operações computacionais baseando-se em propriedades e fenômenos da mecânica quântica. Em computadores clássicos a unidade de informação é o *bit*, o qual pode assumir os valores lógicos "0"ou "1". Analogamente, a unidade de informação quântica é o bit quântico, ou *qubit*. A um qubit podem ser atribuídos os valores lógicos "0", "1", ou qualquer superposição destes. Esta superposição consiste de uma combinação linear dos estados da base computacional descrita por um vetor com a seguinte forma básica, onde, para cada posição desse vetor, um dos estados está associado:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

O vetor $|\psi\rangle$ é chamado de ***ket*** e seu conjugado transposto, $\langle\psi|$, de ***bra***. No estado quântico acima, α e β são amplitudes probabilísticas associadas aos estados, representadas por números complexos e que obedecem à seguinte regra de normalização:

$$|\alpha|^2 + |\beta|^2 = 1$$

Outra forma de interpretar um estado quântico é por meio da sua parametização através

de ângulos θ e ϕ:

$$|\psi\rangle = cos\frac{\theta}{2}|0\rangle + e^{i\phi}sen\frac{\theta}{2}|1\rangle$$

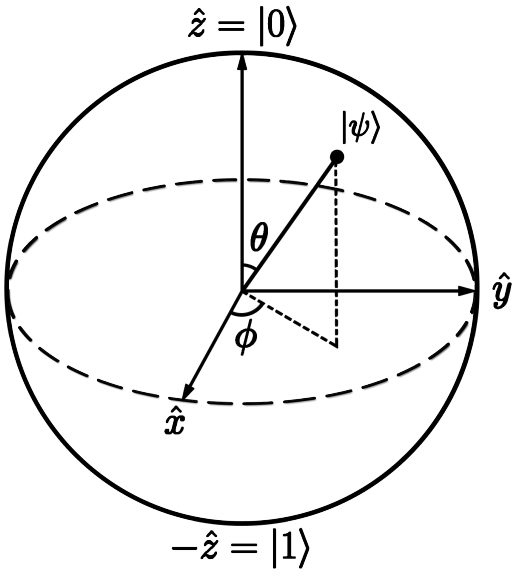
Com o que foi descrito acima, pode-se observar que em um sistema quântico é possível obter um alto grau de paralelismo computacional através da superposição de diversos estados, o que torna um computador quântico uma alternativa eficiente aos computadores clássicos no que se refere à execução de algoritmos de ordem exponencial (classicamente) em tempo polinomial. Tal paralelismo é expresso do seguinte modo:

$$|\psi\rangle := \sum_{i=0}^{2^n-1} a_i|i\rangle,$$

onde tem-se n qubits e a_i amplitudes probabilísticas associadas a i estados. Dessa forma, n qubits podem representar 2^n combinações de estados.

Fisicamente, qubits são representados por qualquer objeto quântico que possua dois auto-estados bem definidos. Na figura abaixo é mostrada uma interpretação gráfica geométrica de como podem ser representado um qubit através da *esfera de Bloch*:

Figura 2.1: Representação geométrica do qubit através da *esfera de Bloch*



Fonte: O autor

O sistema quântico descrito até aqui está associado a um espaço vetorial complexo chamado *espaço de Hilbert*, o qual é equipado com os produtos interno e externo. Nele, seus elementos são vetores complexos que representam o estado físico do sistema. Aqui, o sistema composto é construído através do *produto tensorial*.

O produto interno entre dois estados quânticos resulta em um número complexo e é simbolizado por $\langle\phi|\psi\rangle$, para $|\phi\rangle, |\psi\rangle \in \mathbb{C}^n$, onde o símbolo † representa o conjugado transposto e $*$ representa o conjugado complexo:

$$\langle\phi|\psi\rangle = (|\phi\rangle)^{\dagger}|\psi\rangle = \sum_{i=1}^{n} \phi_i^* \psi_i$$

Já o produto externo dos mesmos estados quânticos $|\phi\rangle$ e $|\psi\rangle$ em $\mathbb{C}^n$, denotado por $|\phi\rangle\langle\psi|$, tem a forma:

$$|\phi\rangle\langle\psi| = \begin{bmatrix} \phi_0\psi_0^* & \phi_0\psi_1^* & \cdots & \phi_0\psi_{n-1}^* \\ \phi_1\psi_0^* & \phi_1\psi_1^* & \cdots & \phi_1\psi_{n-1}^* \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{n-1}\psi_0^* & \phi_{n-1}\psi_1^* & \cdots & \phi_{n-1}\psi_{n-1}^* \end{bmatrix}$$

Como já mencionado acima, caso haja necessidade de representar mais de um qubit no espaço de Hilbert, é feito uso do produto tensorial, que se caracteriza como uma operação bilinear entre espaços vetoriais. A operação de dois ou mais qubits pelo produto tensorial é representada pelo símbolo $\otimes$. Como exemplo, tem-se a seguir o produto tensorial entre 2 qubits expresso em termos simbólicos e de forma matricial:

$$|0\rangle \otimes |0\rangle = |00\rangle; |0\rangle \otimes |1\rangle = |01\rangle; |1\rangle \otimes |0\rangle = |10\rangle; |1\rangle \otimes |1\rangle = |11\rangle$$

$$|00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}; |01\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}; |10\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}; |11\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Generalizando o produto tensorial de vetores representados por matrizes, sejam A e B vetores do espaço complexo, sendo $A \in \mathbb{C}^{m\times n}$ e $B \in \mathbb{C}^{p\times q}$, com $a_{1,1}$ até $a_{m,n}$ sendo os elementos que compõem A, $A \otimes B$ é dado por:

$$A \otimes B = \begin{bmatrix} a_{1,1}B & a_{1,2}B & \cdots & a_{1,n}B \\ a_{2,1}B & a_{2,2}B & \cdots & a_{2,n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1}B & a_{1,1}B & \cdots & a_{m,n}B \end{bmatrix},$$

e resulta em uma matriz de dimensões $mp \times nq$.

Contudo, nem sempre um estado quântico de 2 ou mais qubits pode ser representado dessa forma. Quando este estado não pode ser descrito como um produto tensorial é dito que o mesmo encontra-se *emaranhado*. Dado um sistema representado pelo estado $|\psi\rangle$, se não for possível decompor o mesmo através do produto tensorial de dois ou mais vetores, então há um *emaranhamento*. Como exemplo:

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}},$$

o qual não pode ser representado por meio do produto tensorial de 2 qubits. Tal particularidade faz com que, uma vez realizada uma operação sobre um qubit desse sistema, mesmo que os outros qubits deste estejam fisicamente separados por uma grande distância, estes também sofrerão um efeito da operação aplicada.

Outro aspecto característico da CQ diz respeito a seus operadores. As operações sob um estado quântico são feitas por operadores unitários. Dado um operador $U : H \to H$, com H denotando o espaço de Hilbert, U é referido ser unitário quando seu inverso é igual ao seu conjugado transposto:

$$U^{-1} = U^{\dagger} \qquad ou \qquad UU^{\dagger} = U^{\dagger}U = I,$$

onde I designa o operador *identidade*.

Dessa forma, um operador agindo sobre n qubits precisa estar em um espaço complexo de Hilbert de dimensão 2^n. Devido a este caráter unitário, os diagramas dos circuitos construídos por operadores quânticos apresentam o mesmo número de linhas (qubits) na entrada e na saída.

Agora, considere um registrador com n qubits no estado:

$$|\psi\rangle := \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$$

A aplicação de uma matriz unitária U de dimensão 2^n sobre este registrador, produz:

$$U|\psi\rangle = U\left(\sum_{i=0}^{2^n-1} \alpha_i |i\rangle\right) = \sum_{i=0}^{2^n-1} \alpha_i U|i\rangle$$

Isto é, uma única aplicação de U realiza um número exponencial de operações em estados básicos. Este fenômeno é chamado de *paralelismo quântico*. Usualmente, a matriz de *Hadamard* é utilizada para gerar, num registrador, um estado quântico contendo a superposição de todos os estados básicos, todos com a mesma amplitude.

Uma restrição que é aplicada ao modelo de computar da CQ se refere à *medição* necessária para extrair qualquer informação resultante de um estado de qubits. Ao realizar esta operação, o estado colapsa e apenas um dos qubits que estavam em superposição será observado. Então,

dado um estado quântico:

$$|\psi\rangle := \sum_{i=0}^{2^n-1} \alpha_i |i\rangle,$$

após ser realizada a medição, a probabilidade de se obter a saída $|i\rangle$ é $p_i = |\alpha_i|^2$. O tipo de medição mais comum é a medição *projetiva*, a qual, dado um sistema com estados mutuamente exclusivos, é usada para determinar em qual estado o sistema está. Então, seja um sistema quântico $|\psi\rangle$ de dimensão n, e seja $\{P_1, P_2, ..., P_n\}$ um conjunto de operadores projeção, uma medição nesse sistema resultará na seguinte probabilidade de encontrar a i-ésima saída:

$$Pr(i) = |P_i|\psi\rangle|^2 = (P_i|\psi\rangle)^\dagger (P_i|\psi\rangle) = \langle\psi|P_i^2|\psi\rangle = \langle\psi|P_i|\psi\rangle,$$

uma vez que um operador projeção se caracteriza por:

$$P = P^\dagger \qquad e \qquad P^2 = P$$

De um modo mais geral, as medições podem ser caracterizadas assim:

$$Pr(i) = \langle\psi|M_i^\dagger M_i|\psi\rangle,$$

onde M_i é o operador de medição, i é o índice do estado $|\psi\rangle$ a ser medido e $Pr(i)$ é a probabilidade de encontrar i como resultado.

A representação no circuito quântico de uma medição feita sobre um estado é dada pelo diagrama abaixo:

Por último, tem-se a concepção de *operadores/matrizes densidade*, as quais são uma alternativa à formulação dos estados quânticos através de vetores vista até o momento. Estes operadores proporcionam uma descrição de sistemas quânticos e são definidos como:

$$\rho = |\psi\rangle\langle\psi|,$$

para um dado estado $|\psi\rangle$. Caso o sistema quântico seja formado por um número i de estados, tem-se:

$$\rho = \sum_{i=1}^{n} p_i |\psi_i\rangle\langle\psi_i|,$$

onde p_i é a probabilidade de encontrar o sistema no estado $|\psi_i\rangle$.

Este operador também pode ser caracterizado como uma *matriz densidade*. Em (MC-

MAHON, 2007) tem-se o seguinte exemplo: dado um sistema no estado $|\psi\rangle = \frac{1}{\sqrt{3}}|u_1\rangle + i\sqrt{\frac{2}{3}}|u_2\rangle$, onde $|u_1\rangle$ e $|u_2\rangle$ constituem uma base ortonormal (ou seja, o produto interno de dois elementos iguais da base resulta em 1; caso contrário, resulta em 0). O operador densidade para este estado é:

$$\rho = |\psi\rangle\langle\psi| = \left(\frac{1}{\sqrt{3}}|u_1\rangle + i\sqrt{\frac{2}{3}}|u_2\rangle\right)\left(\frac{1}{\sqrt{3}}\langle u_1| - i\sqrt{\frac{2}{3}}\langle u_2|\right)$$

$$= \frac{1}{3}|u_1\rangle\langle u_1| - i\frac{\sqrt{2}}{3}|u_1\rangle\langle u_2| + i\frac{\sqrt{2}}{3}|u_2\rangle\langle u_1| + \frac{2}{3}|u_2\rangle\langle u_2|$$

E a representação matricial é dada por:

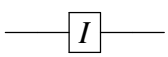
$$[\rho] = \begin{pmatrix} \langle u_1|\rho|u_1\rangle & \langle u_1|\rho|u_2\rangle \\ \langle u_2|\rho|u_1\rangle & \langle u_2|\rho|u_2\rangle \end{pmatrix}$$

$$= \begin{pmatrix} 1/3 & -i\sqrt{2}/3 \\ i\sqrt{2}/3 & 2/3 \end{pmatrix}$$

Tendo em vista esses conceitos, aqui será feita uma abordagem sobre como computar problemas e algoritmos em computação quântica baseada em circuitos, os quais são a base de estudo da CQ. Provavelmente por ser o modelo computacional presente nos computadores clássicos, este paradigma torna-se mais facilmente absorvido por aqueles que desejam aprender sobre CQ. São os circuitos que determinam quais e em que ordem os operadores lógicos são aplicados a um ou mais qubits. São compostos por linhas (uma para cada qubit) e símbolos, que representam os operadores (descrevendo um conjunto de operações quânticas a serem aplicadas a um ou mais qubits). A seguir, os operadores mais comuns encontrados na CQ. Serão fornecidas: suas formas matriciais, o resultado da ação sobre qubits e a representação no circuito:

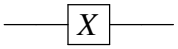
- Identidade - o mesmo qubit de entrada é obtido na saída do circuito:

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \begin{aligned} I|0\rangle &= |0\rangle \\ I|1\rangle &= |1\rangle \end{aligned}$$



- Not - o qubit inicial tem seu valor invertido:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \begin{aligned} X|0\rangle &= |1\rangle \\ X|1\rangle &= |0\rangle \end{aligned}$$



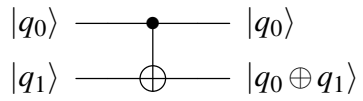
- Hadamard - cria uma superposição dos estados da base:

$$H = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}, \quad \begin{aligned} H|0\rangle &= \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ H|1\rangle &= \tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{aligned}$$

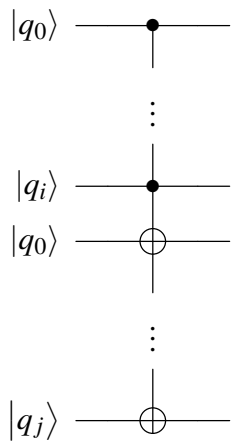
- Controlled-Not - porta lógica de 2 qubits na qual o qubit alvo, representado pelo sinal $\oplus$ no circuito, tem seu valor alterado caso o outro qubit (o de controle) tenha valor 1; caso contrário, nada é alterado:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\begin{aligned} CNOT|0\rangle|0\rangle &= |0\rangle|0\rangle \quad CNOT|1\rangle|0\rangle = |1\rangle|1\rangle \\ CNOT|0\rangle|1\rangle &= |0\rangle|1\rangle \quad CNOT|1\rangle|1\rangle = |1\rangle|0\rangle \end{aligned}$$

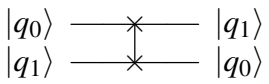


- Controlled-Gate - uma generalização de portas lógicas com bits controladores que pode incluir mais de um qubit como função de controle (i) e mais de um qubit como alvo (j):



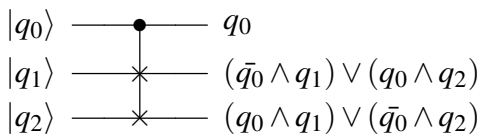
- Swap - porta lógica de 2 qubits que realiza uma troca entre os valores dos mesmos:

$$SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



- Fredkin - porta lógica de 3 qubits que atua como uma *troca controlada*; dessa forma, caso o bit de controle (•) tenha valor 1 os outros dois qubits (×) tem seus valores trocados entre si:

$$Fredkin = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$



Por fim, cabe ressaltar que existem outros modelos de CQ que não seja o de circuitos. Um destes diz respeito à computação quântica adiabática, que é usada em (FARHI et al., 2001) para prover uma solução eficiente para os problemas NP-completos.

A computação quântica adiabática consiste numa abordagem alternativa ao modelo de circuitos da computação quântica e se baseia na evolução do tempo de um sistema quântico (MCMAHON, 2007). Assim, a computação quântica adiabática fundamenta-se no *teorema adiabático*, o qual diz que, dado o *Hamiltoniano H* (um operador que representa a energia total do sistema), se um computador quântico inicia no estado fundamental de H_0, então precisa

terminar no estado fundamental de H_1, tal que a transição de H_0 para H_1 seja lenta o suficiente, objetivando encontrar um Hamiltoniano cujo estado fundamental corresponda à solução do problema de interesse.

A velocidade de evolução de um sistema quântico adiabático em um estado $|\psi(t)\rangle$ no tempo t é descrita pela *equação de Schrödinger*:

$$\hat{H}(t)|\psi(t)\rangle = i\hbar\frac{\partial}{\partial t}|\psi(t)\rangle,$$

onde $\hat{H}(t)$ é o Hamiltoniano do sistema, $\frac{\partial}{\partial t}$ é a derivada no instante t determinada pelo estado $|\psi\rangle$ no mesmo instante, i é o *número imaginário* e $\hbar$ é a *contante de Plank* dividida por 2π.

O texto contido em (AHARONOV et al., 2007) mostra que o modelo de computação quântica de circuitos pode ser eficientemente simulado com um modelo adiabático quântico em tempo polinomial e, inversamente, qualquer algoritmo adiabático pode ser implementado em um circuito quântico. Porém, o objetivo do artigo é caracterizar o poder computacional da computação adiabática e, através de uma série de provas de teoremas e corolários, isto é feito mostrando sua equivalência com a abordagem padrão de computação quântica (de circuitos).

Para uma discussão sobre as capacidades computacionais providas pela evolução adiabática, aconselha-se os textos presentes em (PINSKI, 2011), (DAM; MOSCA; VAZIRANI, 2001) e (FARHI et al., 2000). Neles, é explicitado como se dá a evolução através do teorema adiabático, a abordagem adiabática para a classe de problemas NP e análises de complexidade. Particularmente, em (PINSKI, 2011), são encontrados dados resultantes de simulações executadas com base no modelo quântico adiabático de computação.

Já em (ANDRECUT; ALI, 2004) é feita uma discussão acerca da implementação de portas lógicas quânticas por meio do modelo adiabático. A partir de conceitos como o do Hamiltoniano, o autor se propõe a mostrar que é possível construir as portas lógicas Hadamard, Controlled-Not e Toffoli com respeito ao teorema adiabático.

2.2 Problemas NP-Completos

Visando compreender a formulação dos problemas NP-completos a fim de estabelecer uma descrição concisa dos mesmos, esta seção se propõe a explorar brevemente alguns dos trabalhos que os tem como base. Além disto, uma caracterização formal será dada, bem como serão citados alguns dos problemas mais conhecidos e estudados na literatura. Diversos textos podem ser encontrados nos quais se propõem abordagens para soluções clássicas de problemas NP-completos. Mesmo não sendo o objetivo deste trabalho, tem-se em (MÉZARD; PARISI; ZECCHINA, 2002) um texto que expõe soluções algorítmicas e analíticas para tais problemas, as quais são baseadas em sistemas físicos. A escolha da citação deste texto se deu devido ao mesmo exibir soluções para o problema da satisfatibilidade, estudado na forma geral de $k-$SAT, onde as cláusulas são construídas com k variáveis booleanas.

2.2.1 Revisão bibliográfica

No texto contido em (LUCAS, 2014) tem-se uma formulação dos problemas NP-completos e NP-difíceis baseada em um modelo matemático chamado *Ising*. De acordo com (GAREY; JOHNSON, 1979), um problema é NP-difícil se é um problema de decisão para o qual existe um problema NP-completo que pode ser transformado nele, de forma que o problema NP-difícil não pode ser resolvido em tempo polinomial a menos que $P = NP$ (informalmente é um problema dito ser pelo menos tão difícil quanto os problemas NP-completos). Além de listar e descrever todos os *21 problemas NP-completos de Karp* (os quais são separados por categorias como: problemas em árvore, isomorfismos de grafos, problemas de coloração e de cobertura, entre outros), o autor sugere que tal formulação matemática pode ser útil na construção de algoritmos quânticos adiabáticos de otimização.

A classe de problemas NP-completos configura-se como de extrema importância devido a questões como criptografia, provas matemáticas, organização de metodologia na manufatura, entre outros (FORTNOW, 2009). Caso seja provado que $P = NP$, novas perspectivas de possibilidades computacionais poderão ser exploradas.

Porém, (GAREY; JOHNSON, 1979) mostram em seu estudo sobre intratabilidade que tais problemas são difíceis de computar e provavelmente possuem tempo de execução exponencial com relação ao tamanho da entrada. Mesmo se assim o for, soluções eficientes são ainda necessárias e, neste trabalho de conclusão de curso, considera-se uma abordagem a estes problemas com base em soluções propostas para serem executadas em um computador quântico juntamente com o uso de operadores não-lineares (ABRAMS; LLOYD, 1998), computação de circuitos (OHYA; MASUDA, 1998) e (LEPORATI; FELLONI, 2007), computação quântica adiabática (FARHI et al., 2001) e dinâmica caótica (OHYA; VOLOVICH, 1999). Tais concepções, com seus respectivos algoritmos e sistemas, se propõem a resolver a questão se $P = NP$.

No trabalho apresentado por (FORTNOW, 2009), argumenta-se que, à medida que o poder computacional cresce, com ele também aumenta a importância dos problemas NP e seu contraponto com os problemas que possuem solução de tempo polinomial em uma máquina de Turing determinística (*Polynomial Time* (P)). É feita uma revisão das abordagens mais conhecidas para solucionar *P versus NP*, focando nas tentativas de provar que $P \neq NP$ (as quais tem se mostrado falhas). Abordagens alternativas que se propõem a resolver o problema, como computação quântica e geometria algébrica são citadas, porém este trabalho não aprofunda a pesquisa em tais modelos/técnicas.

Em (KENDALL; PARKES; SPOERER, 2008) foi desenvolvida uma abordagem investigativa baseada na análise de *puzzles* (jogos para uma pessoa), para os quais encontrar uma solução tem uma ordem de complexidade que os enquadra como problemas NP-completos. Para isto, foi feito um levantamento de 24 *puzzles*, com atenção especial dada ao *cubo de Rubik* e à *torre de Hanói*. O autor considera que o estudo da complexidade de alguns problemas a partir de jogos é uma área de pesquisa *fascinante*, porém, o mesmo pensamento parece não

ser compartilhado pela comunidade científica, fato este justificado pela escassez de trabalhos relacionados ao tema. Dessa forma, os 24 jogos listados são definidos com relação às regras e à complexidade, além de ser mostrado como estes jogos são caracterizados como problemas NP-completos. Alguns destes *puzzles* são: *Clickomania*, *Lemmings*, *Minesweeper*, *Peg Solitaire*, *Mahjong Solitaire*, *Solitaire*, *Sudoku*, *Rubik's Cube* e *The Towers of Hanoi*.

A dissertação presente em (FUX, 2004) discute, exclusivamente, o problema NP-completo SAT e como este pode ser trabalhado sob a ótica da lógica multivalorada. O cerne deste trabalho trata da investigação de algumas heurísticas que objetivam encontrar soluções aproximadas para SAT. Alguns algoritmos são apresentados e as heurísticas GRASP (*Generic Search Algorithm for Satisfiability Problem*), SATO (*Satisfiability Testing Optimized*), Zchaff e Berkmin-Berkeley-Minsk são detalhadas com exemplos e os códigos que as implementam. A partir destas, simulações e comparações foram realizadas para determinar a mais eficiente. Por fim, tem-se a conceituação da lógica multivalorada e a apresentação de dois algoritmos implementados (*Davis-Putnam Binário Estendido* e *CAMA*) que encontram soluções para instâncias multivaloradas do SAT.

Em (GAREY; JOHNSON, 1979) é encontrado um texto completo no que se refere a NP-completude. Este trabalho aborda desde complexidade e intratabilidade até a teoria dos problemas NP-completos. Embora levemente desatualizado (por não abranger conceitos mais recentes como o teorema PCP - do inglês, *Probabilistically Checkable Proofs*, o qual afirma que cada problema na classe NP pode ter sua prova checada probabilisticamente), este livro possui conceitos fundamentais para a compreensão de NP-completude, além de reunir uma lista com a caracterização dos problemas NP-completos.

Por fim, (AARONSON, 2005), em uma análise da concepção física dos problemas NP-completos, argumenta que, para tentar resolver essa classe de problemas em tempo polinomial, torna-se necessário estudar não só computação, mas também as características físicas que envolvem os sistemas que se propõem a tal. Aqui, são listados diversos modelos de possíveis soluções computacionais não usuais (computação quântica, algoritmos adiabáticos, variáveis ocultas, dilatação relativista do tempo, gravidade quântica, entre outros) e uma discussão acerca da realidade física de cada um é feita. Importante ressaltar que o autor acredita que nenhuma das propostas expostas por ele seja capaz de resolver de modo eficiente os problemas NP-completos, porém, a argumentação aqui é de que tal estudo contribui tanto para a computação quanto para a física envolvida nesta.

2.2.2 Definição

A partir de (GAREY; JOHNSON, 1979), os problemas da classe NP são definidos como problemas de decisão os quais podem ser resolvidos em tempo polinomial por um computador não-determinístico. Porém, em uma máquina determinística, os problemas desta classe tem apresentado apenas um tempo de ordem exponencial para a descoberta de uma solução. Outro

ponto a ser frisado é que, dada uma possível solução para o problema, é possível verificar deterministicamente em tempo polinomial se esta é uma real solução ou não para o problema.

Também em (GAREY; JOHNSON, 1979) é dito que os problemas classificados como NP-completos são aqueles para os quais, dado um outro problema B da classe NP, é possível realizar uma transformação de redutibilidade em tempo polinomial do problema NP-completo A para B. Dessa forma, uma solução algorítmica de tempo polinomial para um problema NP-completo pode ser convertida em um outro algoritmo, também de tempo polinomial, que soluciona o problema B. Assim, dados dois problemas A e B da classe NP e uma transformação T, se:

$$T : B \xrightarrow{tempo\ polinomial} A,$$

então existe um T' que transforma uma solução de tempo polinomial para A em uma solução de tempo polinomial para B:

$$T' : Sol(A) \rightarrow Sol(B),$$

e, assim, A é classificado como NP-completo. Portanto, um problema NP-completo é um problema de decisão que pertence à classe de problemas NP e, para todos os outros problemas NP, estes são redutíveis polinomialmente a ele.

Para este trabalho, as pesquisas e simulações foram feitas com base no problema NP-completo da *Satisfatibilidade* - **SAT**. Esta escolha se deu devido ao mesmo ser um problema de lógica booleana, sendo, dessa forma, mais próximo ao cenário de circuitos clássicos dominante no que se refere à computação clássica; outro motivo centra-se na grande quantidade de trabalhos que usam o SAT como alvo para o estudo de soluções eficientes para os problemas NP-completos.

Cabe lembrar que, dado um problema que pertence à classe de problemas NP-completos, uma vez encontrada uma solução em tempo polinomial para este, é possível resolver todos os outros problemas da classe NP através de uma redução do primeiro para todos os outros.

2.2.2.1 SAT

O problema SAT, como definido em (OHYA; MASUDA, 1998), consiste em um conjunto $X \equiv \{x_1, x_2, ..., x_n\}$ onde x_k e sua negação $\bar{x_k}$ são chamados de literais. $\mathscr{F}(X')$ é usado para representar o conjunto de todos os subconjuntos de $X' \equiv \{x_1, \bar{x}_1, ..., x_n, \bar{x}_n\}$, e $C \in \mathscr{F}(X')$ constitui uma cláusula. Tomando os elementos de C, se for possível ter uma atribuição verdadeira a pelo menos um deles, então o valor $t(C)$ também é verdadeiro. Uma vez que este problema figura como uma fórmula booleana na forma normal conjuntiva, para que $\mathscr{C} = \{C_1, C_2, ..., C_j\}$ seja satisfatível, faz-se necessário que todos ao valores de $t(C)$ sejam verdadeiros para todo $C_j (j = 1, 2, 3, ..., m)$, ou seja, $t(\mathscr{C}) \equiv \wedge_{j=1}^{m} t(C_j) = 1$, onde $t(C) \equiv \vee_{x \in C} t(x)$ e $\vee$ e $\wedge$ são os habituais operadores 'OR'e 'AND'. Assim, o problema SAT, fundamentalmente, resume-se a perguntar se é possível atribuir um conjunto de valores verdade a $\mathscr{C}$ que o faça satisfatível.

Definição formal SAT: Dado um conjunto $X' \equiv \{x_1, \bar{x}_1, ..., x_n, \bar{x}_n\}$ de literais e suas negações e um conjunto $\mathscr{C} = \{C_1, C_2, ..., C_j\}$ de cláusulas, determinar se $\mathscr{C}$ é satisfatível ou não.

Já o 3-SAT é definido como sendo um problema de satisfatibilidade onde existem, precisamente, 3 literais por cláusula.

Definição formal 3-SAT: Dado um conjunto $X' \equiv \{x_1, \bar{x}_1, ..., x_n, \bar{x}_n\}$ de literais e suas negações e um conjunto $\mathscr{C} = \{C_1, C_2, ..., C_j\}$ de cláusulas que contém, exatamente, 3 elementos de X', determinar se $\mathscr{C}$ é satisfatível ou não.

2.3 Python e SymPy

2.3.1 Motivação do uso

O Python é uma linguagem de programação de alto nível que visa descomplicar o uso e a aprendizagem. Para promover esta finalidade, a linguagem suporta múltiplos paradigmas de programação, incluindo funcional, imperativa e orientada a objetos. Dessa forma, em posse de sua sintaxe limpa e simplificada, Python torna-se fácil de ler. Por ser uma linguagem interpretada, os códigos escritos em Python podem ser executados diretamente no interpretador, sem a necessidade de compilação prévia. Dotada de diversos módulos que possibilitam cálculos matemáticos e científicos como o NumPy [1] e o SciPy [2], Python dispõe de uma biblioteca para computação simbólica, o SymPy.

Portanto, uma vez que os fenômenos quânticos podem ser contra-intuitivos, torna-se útil a existência de um software capaz de simulá-los em um computador clássico. Em se tratando de simulações quânticas de um sistema com n estados em computadores clássicos, são necessários, nestes últimos, 2^n estados. Essa razão exponencial faz com que se torne comumente impraticável a simulação de tais sistemas. Este cenário, então, se concretiza com o uso do SymPy, um CAS presente na linguagem de programação Python que modela e abstrai a estrutura dos operadores e elementos quânticos tornando-os de natureza simbólica. Neste trabalho, será empregada a sintaxe simples e limpa do Python em conjunto com funções, estruturas e classes do SymPy, que caracterizam as peculiaridades da mecânica quântica. Tal uso resultará em uma representação simbólica de circuitos capazes de computar instâncias do SAT e os operadores responsáveis por concluir sua satisfatibilidade.

Em (ARI; MAMATNAZAROVA, 2014) a computação simbólica é definida como uma *"computação na qual os objetos matemáticos são tratados simbolicamente, ou seja, são representados exatamente como são (não aproximadamente) e as variáveis não avaliadas são deixadas na forma simbólica"*. Também neste trabalho são apresentadas particularidades do SymPy para a solução simbólica de problemas matemáticos como derivadas, integrais, limites e sistemas de equação. Aqui, ressalta-se a importância de o SymPy ser *open source* (código aberto) e ter

[1] http://www.numpy.org/
[2] http://www.scipy.org/

seu código inteiramente escrito em Python, o que lhe confere um certo nível de simplicidade da sintaxe. Outros pontos argumentados em favor do uso do SymPy são: possibilidade de realizar diversos tipos de cálculos simbolicamente, ser gratuito (diferente das alternativas existentes como o Maple [3] ou Mathematica [4]), além de ser de fácil instalação. Por fim, diversos exemplos de códigos são apresentados envolvendo variados conceitos matemáticos.

Para uma síntese que engloba desde o surgimento do SymPy, passando pela sua licença, instalação, desenvolvedores e documentação, até exemplos de uso e capacidades computacionais, tem-se a pesquisa desenvolvida em (JOYNER et al., 2012). Além dos aspectos já mencionados, este trabalho explana de modo breve sobre o pacote quântico do SymPy usado nesta monografia.

Como já exposto nos parágrafos anteriores, o SymPy possui a capacidade de lidar com diversos problemas matemáticos e tratá-los de maneira simbólica, facilitando, assim, o entendimento dos mesmos. Conforme também já foi mencionado, o SymPy é dotado de um pacote para a realização de física quântica, o *Quantum*. O mesmo possui em sua estrutura diversas classes que servem de base para aplicações de funções quânticas, estados, operadores e computação quântica, e algumas de suas características serão expostas na subseção seguinte.

2.3.2 Módulo *Quantum*

Objetivando facilitar a compreensão dos conceitos relacionados à mecânica/computação quântica, o SymPy surge como uma ferramenta capaz de abstrair e simplificar os detalhes algébricos e aritméticos que podem desestimular e dificultar o processo de obtenção do conhecimento. A biblioteca apresenta em sua composição um pacote específico para CQ, chamado *Quantum*. Este possui as noções de espaço de Hilbert, ket, bra (o conjugado transposto do ket), operadores que agem sobre estados, produto interno/externo, produto tensorial e várias outras funcionalidades que fornecem amplas possibilidades para a implementação de algoritmos quânticos, alguns destes previamente disponíveis para estudo como Shor e Grover. Desse modo, os códigos provenientes das simulações implementadas em SymPy possuem potencial para serem usados como uma ferramenta para ensino e pesquisa de CQ.

2.4 Observações finais

Este capítulo visou fornecer os conceitos básicos para o entendimento do que será exposto a partir daqui. Sabendo que se trata apenas de um resumo, aconselha-se a leitura dos textos referenciados, os quais foram usados como suporte para embasar as definições apresentadas.

Embora um dos pressupostos teóricos deste trabalho seja o de que o modelo de computação quântica por circuitos possa ser implementado de alguma forma, considera-se válida a menção aos trabalhos desenvolvidos em (SARKAR; BHATTACHARYYA; PATWARDHAN, 2006),

[3] http://www.maplesoft.com/products/maple/
[4] https://www.wolfram.com/mathematica/

(MONROE; KIM, 2013), (LANTING et al., 2014), (GIOVANNETTI; LLOYD; MACCONE, 2008) e (DICARLO et al., 2009), onde investiga-se possíveis implementações de hardware quântico.

Mais especificamente, em (SARKAR; BHATTACHARYYA; PATWARDHAN, 2006) tem-se realizações que conduzem à tentativas de desenvolver processadores quânticos lógicos no *interferômetro de Mach-Zehnder*, onde portas lógicas de 1 qubit poderiam ser obtidas através da manipulação do spin dos elétrons e portas lógicas de 2 qubits através do grau de liberdade dos elétrons.

Em (MONROE; KIM, 2013) tem-se uma revisão do desenvolvimento de tecnologias que visam prover escalabilidade ao modelo de construção de hardware quântico baseado em íons atômicos presos - tal escolha deve-se ao fato de que íons presos possuem características peculiares (relativas à eficiência, emaranhamento e coerência) que os tornam uma escolha interessante para este propósito.

O trabalho experimental realizado em (LANTING et al., 2014) demonstra que, em um processador quântico arrefecido (do inglês, *quantum annealing processor*) - o qual também é descrito no trabalho - é possível obter o emaranhamento de qubits e os mesmos se manterem em estado de equilíbrio, o que, segundo os autores, qualificaria o arrefecimento quântico como um caminho viável para a construção de processadores quânticos. Há, ainda, uma descrição matemático-experimental do chip que compõe o processador quântico proposto, além de diversos gráficos que ilustram os experimentos realizados.

Já em (GIOVANNETTI; LLOYD; MACCONE, 2008) é feita uma implementação óptica objetivando a construção de memórias RAM (*Random Access Memory*) quânticas. A realização de tal implementação se dá através da composição de íons ou átomos aprisionados como os elementos fundamentais, além de fótons compondo os registradores e que constituem uma arquitetura capaz de reduzir exponencialmente as requisições de chamadas à memória.

Por outro lado, o trabalho experimental de (DICARLO et al., 2009) tem em sua pesquisa o foco em um dispositivo quântico supercondutor onde é possível executar algoritmos com 2 qubits. São mostradas as particularidades físicas deste dispositivo a partir das portas lógicas quânticas e de operações presentes no algoritmo de busca de Grover. As representações físico-matemáticas necessárias para a implementação dos conceitos quânticos neste dispositivo também são dadas.

Tais propostas de desenvolvimento de hardware quântico podem ser vislumbradas como alternativas para implementação dos modelos quânticos de soluções para *P versus NP* aqui apresentados.

O Quadro 2.1 destaca as principais classes do *Quantum* e suas respectivas ações empregadas para o presente estudo da computação quântica bem como para o desenvolvimento dos simuladores aqui propostos.

Quadro 2.1: *SymPy/Quantum* - resumo das classes e funcionalidades

Classe	**Ação**
Dagger	A partir de um dado estado/operador simbólico ou uma matriz que o represente, retorna seu conjugado transposto
TensorProduct	Dados dois estados, retorna uma instância simbólica de TensorProduct ou, para matrizes, retorna uma matriz resultante do produto tensorial entre estas através do método `matrix_tensor_product`
Qapply	Uma expressão contendo operadores e estados é passada como parâmetro e a classe retorna a expressão original com a aplicação dos operadores aos estados feita
Qubit	Provê uma representação dos qubits para computação quântica. Também possui os métodos: ▪ `qubit_to_matrix`: converte um qubit para sua representação matricial ▪ `matrix_to_qubit`: realiza o inverso do método anterior; dada uma matriz, retorna o qubit que esta representa ▪ `matrix_to_density`: a partir de uma matriz, encontra o operador densidade que esta caracteriza. Retorna um objeto da classe `Density`
Gates	Possui as implementações das portas lógicas quânticas mais usuais, tais como: ▪ `HadamardGate` ▪ `CGate` - Usada para construir as portas lógicas Controlled-Not (em conjunto com `XGate`), Fredkin (em conjunto com `SwapGate`) e demais portas de controle ▪ `XGate` ▪ `IdentityGate` ▪ `SwapGate` As funcionalidades de tais portas lógicas estão explicitadas na Seção 2.1.2
Density	Recebe um sistema quântico como parâmetro e retorna uma caracterização das probabilidades dos estados que o compõe
CircuitPlot	A partir de uma sequência de operações com portas lógicas quânticas em conjunto com os índices dos qubits que receberão sua ação, essa classe constrói um diagrama expondo o circuito e as representações das portas lógicas

Fonte: O autor

3

Trabalhos Relacionados

3.1 Considerações iniciais

O presente capítulo trata dos trabalhos diretamente relacionados à pesquisa aqui desenvolvida, de forma a contemplar algumas das mais diversas abordagens presentes na literatura. Assim, são apresentados textos que objetivam mostrar soluções quânticas que discutem e visam elucidar a questão *P versus NP*. Além destes, publicações que abrangem o uso geral do SymPy e sua aplicação para a simulação das particularidades da CQ também serão expostos.

O problema aqui abordado consiste em duas etapas. A primeira diz respeito ao que se tem produzido quando do assunto referente ao uso da CQ para a solução da classe de problemas NP-completos em tempo polinomial. A segunda parte trata de simulações quânticas (simbólicas ou não) que objetivam representar fenômenos quânticos em computadores clássicos.

Por fim, para este trabalho, tem-se a junção dessas duas fases de pesquisas para alcançar os objetivos propostos.

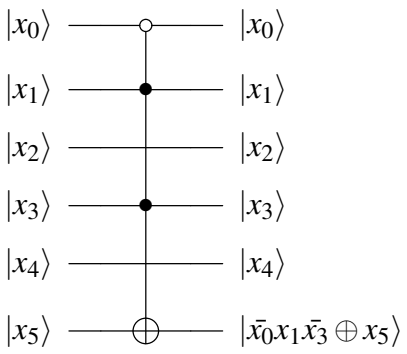
3.2 Levantamento bibliográfico

3.2.1 Soluções quânticas para problemas NP-completos

Em (ARAÚJO; FINGER, 2011) é dada uma definição do problema SAT em termos quânticos (*Quantum Satisfiability* (QSAT)) em comparativo com a versão clássica do mesmo problema. A partir desta definição, o autor descreve uma versão lógica para o QSAT, chamada QSAT^l, a qual é baseada na álgebra linear e é mostrado que esta versão quântica do SAT não consiste em uma extensão do problema da satisfatibilidade clássico, dessa forma, não podendo haver uma comparação de complexidade direta entre as classes NP e o análogo quântico da classe *Arthur-Merlin* (do inglês, *Arthur-Merlin Complexity Class* (MA)) - chamado *Quantum Arthur-Merlin Complexity Class* (QMA). Assim, os autores optaram por converter as atribuições de valores para a fórmula no SAT em matrizes densidade no QSAT^l para, com isso, estabelecer uma relação de associação entre o SAT clássico e o quântico.

Outro método proposto para testar a satisfatibilidade de uma fórmula booleana é encontrado em (WANG et al., 2012). Nesta abordagem são usadas operações lógicas usuais, tais como AND, OR e XOR em conjunto com um circuito que implementa o algoritmo de Deutsch-Jozsa (DEUTSCH; JOZSA, 1992) com portas lógicas Toffoli estendidas (do inglês, *Extended Toffoli Gate* - ETOF). É mostrado como converter o problema SAT para portas lógicas quânticas ETOF e como sucessivas medições da saída em conjunto com a aplicação do circuito Deutsch-Jozsa podem prover, através de uma análise probabilística do algoritmo, uma solução para o problema da satisfatibilidade em tempo quadrático em relação ao seu análogo clássico, tendo a vantagem de manter a complexidade mesmo quando não for possível determinar um conjunto de atribuições que torne a fórmula satisfatível. A Figura 3.1 ilustra uma porta lógica quântica ETOF.

Figura 3.1: Porta lógica ETOF



Fonte: (WANG et al., 2012)

Masanori Ohya tem desenvolvido algoritmos e explorado os problemas NP-Completos em alguns de seus trabalhos. Em (OHYA, 2012) é exposta uma revisão sobre o algoritmo quântico utilizado em conjunto com as propriedades da dinâmica caótica que resolve o problema NP-Completo SAT em tempo polinomial. É dada, além da formulação do problema da satisfatibilidade, uma caracterização do algoritmo quântico que avalia uma fórmula booleana representando uma instância do problema. Uma consideração feita consiste no uso de um amplificador dinâmico caótico em conjunto com o computador quântico. Essa escolha se dá devido ao seu comportamento caracterizado por uma sensibilidade exponencial às condições iniciais. Assim, a natureza caótica presente no mapa logístico clássico seria usada para distinguir entre uma certa saída $q = 0$ e outra muito pequena, porém, diferente de 0. Seguindo o mesmo algoritmo do trabalho citado acima, em (OHYA; VOLOVICH, 2003) o autor fornece provas matemáticas e uma descrição mais detalhada dos teoremas utilizados.

Anteriormente, em (OHYA; VOLOVICH, 1999) a mesma abordagem com uso da dinâmica caótica já havia sido empregada, porém com sua execução realizada em um modelo de computador atômico, o qual permitiria o uso de portas lógicas quânticas não-lineares descritas pelas equações *Hartree-Fock*. Aqui, são usadas matrizes densidade representativas do estado

quântico como o dado inicial para os cálculos do mapa logístico no qual se baseia o amplificador dinâmico caótico (responsável por amplificar a saída a ser medida). Este trabalho será detalhado na Seção 4.1.1.

Outro modelo proposto para a solução eficiente do problema NP-completo SAT é mostrado em (OHYA, 2005), onde é sugerido o uso de um algoritmo quântico adaptativo em conjunto com entropia mútua quântica, a qual é aplicada para fornecer uma descrição dos processos de comunicação quânticos. Com base nesses conceitos, alguns teoremas são citados objetivando fornecer uma base para a justificação da entropia.

Em (OHYA; MASUDA, 1998) foi proposto um algoritmo que, a partir da aplicação de um operador que polarizasse a saída do circuito que computa uma instância do SAT seguida da aplicação de um operador de projeção, seria capaz de resolver o problema da satisfatibilidade em tempo polinomial. Esses operadores são, respectivamente:

$$V_\theta \equiv \otimes_1^{n+l}(A|0\rangle\langle 0| + B|1\rangle\langle 1|) \otimes e^{i\theta f(C)} I$$

e

$$E \equiv \otimes_1^{n+l} I \otimes |1\rangle\langle 1|,$$

onde $n+l$ é a quantidade total de qubits necessários, $f(C)$ representa o resultado do circuito que analisa a fórmula booleana, θ é uma certa constante que descreve a fase do vetor $|f(C)\rangle$ e:

$$A \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad B \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$$

Com exceção deste, os outros trabalhos deste autor não apresentaram um exemplo de computação prático que demonstrasse o resultado esperado. Assim, a partir de uma dada instância do SAT é explicitado como construir o circuito quântico que avalia a fórmula na FNC e um diagrama do mesmo é mostrado. Além disso, é feita uma análise de complexidade do algoritmo apresentado.

Uma outra abordagem para o tema é encontrada em (SONG, 2014), onde o problema *P versus NP* é explorado segundo a ótica dos processos físicos. Neste sentido, os problemas P são considerados como uma classe de processos físicos determinísticos de tempo polinomial, enquanto NP é visto como seu análogo não-determinístico. Uma revisão de Máquinas de Turing é feita, seguida da construção de um modelo computacional não-determinístico particular. Assim, uma discussão sobre a possibilidade de computar processos físicos não-determinísticos em tempo polinomial em máquinas determinísticas é concebida seguindo o modelo quântico de computação.

A solução apresentada por (FARHI et al., 2001) fundamenta-se na concepção de um algoritmo evolutivo em um sistema quântico adiabático. É considerado que o comportamento

quântico adiabático constitui uma base para novos algoritmos na computação quântica. Essa abordagem baseada na evolução adiabática foi aplicada a aleatórias instâncias do problema NP-completo *"Cobertura Exata"*. A partir de pequenas instâncias é dito que o algoritmo foi simulado em um computador clássico. Uma vez que os resultados são probabilísticos, sugere-se que repetidas aplicações do mesmo algoritmo poderiam resultar em uma probabilidade de sucesso de 0.99997 para a solução do problema. Lembrando que os estados quânticos evoluem de acordo com a equação de Schrödinger:

$$\hat{H}(t)|\psi(t)\rangle = i\hbar\frac{\partial}{\partial t}|\psi(t)\rangle$$

A equação acima permite que o estado $|\psi(t)\rangle$ seja expresso em qualquer instante de tempo. Logo, dado um Hamiltoniano $H(t)$ e um estado quântico inicial $|\psi(0)\rangle$, a evolução adiabática deverá variar lentamente o Hamiltoniano para que $|\psi\rangle$ alcance o estado fundamental que codifica a solução para o problema. A questão central aqui é determinar o quão lento deverá ser essa variação.

Continuando na linha de investigação de soluções para NP baseadas no teorema adiabático, (BOLOTIN, 2014) argumenta que, uma vez que as características da teoria quântica permitem que se possa encontrar soluções para a equação de Schrödinger em um tempo razoável, então $P = NP$. A partir deste cenário, o autor realiza uma análise crítica partindo de uma prova construída para demonstrar que, sendo a equação de Schrödinger tratada como um problema computacional, esta é um problema NP. Aqui, também considera-se para estudo a possibilidade de se obter propriedades completas de objetos macroscópicos a partir de seus microestados com base em um sistema regido por esta equação. Algumas considerações são feitas sobre a teoria de complexidade computacional, a fim de se investigar como a citada equação se enquadra dentro desta.

Na proposta apresentada por (FELDMANN, 2012) tem-se a teoria probabilística Bayesiana. É apresentada uma revisão das bases da computação baseada em algoritmos, seguida das descrições e definições acerca de computação quântica, probabilidade de Bayes e complexidade de algoritmos. A partir da formulação Bayesiana de algoritmos, o problema 3-SAT é resumido em um problema de programação linear e sua estrutura é explorada com base em um sistema auxiliar de Bayes. A partir da implementação de tal sistema, é provado que, uma vez que 3-SAT só é satisfatível se for possível criar um sistema auxiliar de Bayes, tal viabilidade é obtida em tempo polinomial, o que é mostrado com exemplos. Portanto, utilizando o argumento de que computação clássica e quântica são casos especiais de raciocínio probabilístico, o problema 3-SAT é reduzido a um problema de programação linear. Segundo o autor, de acordo com a teoria de complexidade algorítmica, isto seria suficiente para provar que $P = NP$.

São propostos em (LEPORATI; FELLONI, 2007) três algoritmos para resolver eficazmente, de forma específica, o problema NP-completo 3-SAT. Para isso, são apresentados três

meios diferentes: o primeiro consiste na construção de um circuito quântico composto apenas de portas lógicas Fredkin, as quais seriam capazes de computar, em superposição, todos os valores possíveis para uma dada instância do problema; o segundo algoritmo faz o mesmo, porém utilizando registradores em uma máquina de registradores quânticos; por fim, têm-se a solução para o problema baseada em níveis de energia de uma membrana em um sistema quântico P (do inglês, *P system* - não confundir com a classe de problemas *P*). Este trabalho será detalhado na Seção 4.1.2.

Na proposta exposta por (ABRAMS; LLOYD, 1998) a não-linearidade apresentada durante a evolução no tempo dos estados quânticos é explorada através de portas lógicas que possuam tal característica. A partir do modelo de Weinberg de mecânica quântica não-linear, este trabalho propõe três algoritmos capazes de resolver problemas P e NP-completos em tempo polinomial através da aplicação de portas lógicas construídas com base no argumento de que, virtualmente, qualquer teoria quântica determinística não-linear incluirá portas lógicas que agem não-linearmente. Os primeiros dois algoritmos iniciam no seguinte estado quântico:

$$\psi = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i,0\rangle,$$

que é obtido a partir da realização de rotações de $\pi/2$ em cada um dos n primeiros qubits, onde n representa os qubits de entrada e $0 < n < 2^n - 1$. Após o uso do oráculo (uma espécie de *caixa-preta* responsável por computar a solução para o problema), o estado quântico será:

$$\psi = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i,f(i)\rangle,$$

onde $f(i)$ representa a resposta da função para o i-ésimo qubit. A partir disto, operações não-lineares (as quais diferem entre os dois algoritmos) são feitas neste estado para permitir que uma medição no mesmo seja capaz de retornar a solução correta. Já para o terceiro algoritmo proposto, a rotação é de $\phi < 45^\circ$ e os próximos passos são baseados na evolução no tempo do sistema de acordo com um Hamiltoniano não-linear. Assim, a quantidade de aplicações destes algoritmos está relacionada com a extensão angular da região de não-linearidade.

Em (FREEDMAN, 1998) alega-se que a Teoria Quântica de Campo Topológica (TQFT) pode, em algum sistema físico com um termo topológico não-Abeliano, manipular tal sistema para que este se comporte como um computador análogo capaz de resolver os problemas NP. A dificuldade aqui encontrada concentra-se no processo de extração de informação do sistema, o qual seria feito através de medições com acurácia limitada. Para a proposta apresentada, algumas teorias matemáticas são citadas com o objetivo de embasar a construção do sistema. Entre elas está a *Teoria dos Nós*, através do *polinômio de Jones* que, conectado com a TQFT e associado a outras teorias matemáticas topológicas, tornaria o sistema capaz de fornecer informação suficiente para a resolução de problemas NP. Por fim, uma discussão sobre a complexidade dos

sistemas físicos e a dificuldade no processo de medição, acentuam a necessidade de se construir estruturas que tornem o sistema confiável e eficaz.

No trabalho apresentado por (LIMA; ISIDRO; LULA JÚNIOR, 2007) têm-se uma síntese das abordagens existentes que se propõem a resolver *P versus NP*. São considerados a dinâmica não-linear, o algoritmo quântico adiabático e a dinâmica não-unitária. Embora aborde algumas das diversas técnicas para a solução de problemas NP-completos em computação quântica, o texto não apresenta algo que contribua diretamente no tema, servindo, assim, como um resumo do que foi feito até aquele momento.

Em (AHARONOV; NAVEH, 2002) é feita uma revisão sobre QMA, o análogo quântico da classe de problemas MA, que, por sua vez, é o análogo probabilístico dos problemas NP. Dessa forma, após as definições referentes à classe QMA, o trabalho concentra-se em provar a analogia feita entre estas e outras classes de problemas clássicos e quânticos. Além disso, uma rápida revisão sobre análise de complexidade computacional é feita antes de relacionar o problema do "Hamiltoniano local" como uma extensão do 3-SAT e provar sua pertinência à classe QMA. Questões como redução de problemas e completude são discutidas e algumas outras em aberto, relacionadas com a complexidade do Hamiltoniano local e com a classe QMA, são listadas. Com relação às classes de complexidade, são citadas as classes: *Bounded-error Quantum Polynomial Time* (BQP), que consiste na classe de problemas os quais podem ser resolvidos em tempo polinomial em um computador quântico com uma certa probabilidade de erro máxima; *Bounded-error Probabilistic Polynomial time* (BPP), análogo clássico da classe BQP; *Probabilistic Polynomial Time* (PP), o mesmo que a classe BPP, porém, sem a probabilidade de erro associada; QMA e *Quantum Classical Arthur-Merlin Complexity Class* (QCMA). Esta última se refere à uma variação da classe QMA, onde a solução para os problemas de decisão da mesma podem ser verificados em um computador quântico, porém, com uma prova clássica. O seguinte teorema é mencionado:

$$BPP \subseteq BQP \subseteq QCMA \subseteq QMA \subseteq PP$$

Uma proposta de solução quântica para problemas NP pode ser encontrada em (OLIVEIRA, 2015), onde é considerado que, dada uma árvore com um parâmetro constante que modela a estrutura de um problema NP, é possível resolvê-lo em tempo polinomial. Alguns teoremas e definições acerca de circuitos quânticos, problemas NP e conceitos sobre árvores são dados para possibilitar a construção da solução apresentada.

Em (GAITAN; CLARK, 2014) é usado como exemplo o problema do isomorfismo de grafo e é mostrado como convertê-lo em um problema de otimização combinatória que pode ser resolvido para instâncias arbitrárias usando a evolução quântica adiabática. Uma simulação numérica do algoritmo quântico é mostrada, bem como uma discussão sobre uma implementação experimental do mesmo.

3.2.2 Simuladores quânticos e programação simbólica com Python/SymPy

A proposta apresentada por (RADTKE; FRITZSCHE, 2005) consiste na descrição de um programa simulador (chamado de FEYNMAN) capaz de prover as ferramentas necessárias para definir e lidar com registradores (em termos de vetores de estados e como matrizes densidade) e operadores quânticos (restritos à transformações unitárias). Inicialmente é fornecido um resumo do ambiente de desenvolvimento, bem como uma sinopse do funcionamento e características do programa. O mesmo foi desenvolvido com a finalidade de facilitar a simulação de sistemas quânticos gerais de n-qubits, utilizando, para isso, uma codificação em conjunto com o framework Maple, devido a este proporcionar ferramentas para computação simbólica e numérica. São listados comandos que podem ser aplicados em 1 único ou em 2 qubits, além de procedimentos quânticos auxiliares com a finalidade de representar estruturas básicas de dados como qubits e registradores. Outros comandos disponíveis tratam da aplicação de operadores, normalização de vetores, plotagem de gráficos, entre outros. Aqui, o autor ressalta que o propósito de seu programa é tornar-se útil para o ensino de elementos básicos da computação quântica bem como para o estudos de suas possíveis realizações físicas (apesar de o mesmo não ter sido desenvolvido com base em nenhuma possibilidade de sistema físico em particular).

No trabalho apresentado por (BARBOSA, 2007) são expostos os aspectos relativos ao desenvolvimento de um CAS próprio para a representação de circuitos quânticos, sendo este uma extensão de outro simulador, o *Zeno*[1]. O autor argumenta que tal simulador permite que as peculiaridades quânticas sejam manipuladas de um modo fácil, objetivando simplificar o entendimento dos algoritmos quânticos. Após uma revisão sobre circuitos quânticos e álgebras computacionais (em geral e específicas para a CQ), tem-se as características do desenvolvimento e da aplicação em si, incluindo as linguagens de programação utilizadas e o processo de integração com o *Zeno*. Além disto, a estrutura do CAS é detalhada de forma a prover uma visão geral das capacidades matemático-simbólicas disponíveis.

Com o objetivo de diminuir a complexidade dos circuitos lógicos devido às redundâncias matemáticas existentes, (CURRY, 2011) apresenta regras de simplificação de circuitos quânticos utilizando a computação simbólica presente no SymPy. É fornecida uma base teórica sobre circuitos lógicos quânticos e, posteriormente, são mostrados exemplos de como usar a sintaxe do SymPy para criar bits quânticos, operadores, portas lógicas, registradores e circuitos gerais. Além destes, é mostrado como encontrar circuitos equivalentes a outros através de funções da própria linguagem.

Ainda em (CURRY, 2011) são dadas as relações de simplificação para otimização de circuitos quânticos, tais como: duas portas lógicas idênticas e Hermitianas podem ser removidas ou substituídas pela porta lógica *identidade* para reduzir o custo computacional do circuito; relações de comutação; busca por sequências não-triviais específicas de portas lógicas que, juntas, resultam na simplificação para a porta *identidade* ou que ocasionam o aparecimento de

[1] http://www.dsc.ufcg.edu.br/ iquanta/zeno/

novos relacionamentos entre as mesmas. Vale ressaltar que, apesar de mostrar alguns comandos essenciais para simplificação de circuitos quânticos, não foram dados exemplos mais práticos do uso do SymPy para a realização do modelo de circuitos da computação quântica.

No trabalho apresentado por (CUGINI, 2011) é mostrado como usar o SymPy para construir uma simulação de um computador quântico com foco no operador densidade da mecânica quântica estatística usando, para isso, matrizes densidade presentes no próprio SymPy. Tomando como verdade que a mecânica quântica seria de complexo entendimento e tem sua dificuldade algébrica elevada, este trabalho mostra como usar a sintaxe simplificada do SymPy como facilitador no aprendizado. Uma discussão sobre as estruturas de dados presentes na linguagem e seu uso na modelagem da mecânica quântica em geral é feita (tais como as subclasses **Mul**, **Add** e **Pow** para a representação de expressões da classe **Expr**). Por sua vez, são expostos exemplos de como simular estados, operadores e portas lógicas quânticas, bem como suas representações no circuito. Códigos de uso para as classes **State**, **Gate** e **Operator** são apresentados (em conjunto com a classe **CircuitPlot**) para demonstrar a construção de alguns circutos e operações elementares, como a Transformada de Fourier Quântica (QTF) - na sua forma diagramática e matricial. Por fim, são utilizados os conceitos de operador densidade, através da classe **Density**, para uma melhor descrição do sistema quântico e sua eventual modelagem, bem como a entropia de *von Neumann* para simular fenômenos físicos presentes na computação quântica, como o emaranhamento e a concepção de matrizes densidade reduzidas. Considerações sobre outros aspectos relativos à construção de circuitos quânticos para simulações, como as medições, não foram contemplados neste trabalho.

Vale acrescentar a contribuição do presente autor na área na forma do artigo (OLIVEIRA; OLIVEIRA, 2015), em conferência nacional de CQ, resultado do trabalho em PIBIC, sob orientação do Prof. Wilson de Oliveira (DEInfo-UFRPE), em que fazemos estudos das soluções de problemas NP-completos, em particular do SAT, através da representação/simulação simbólica de elementos e operadores quânticos presentes no SymPy, usando a abordagem e os operadores propostos em (OHYA; MASUDA, 1998).

3.3 Considerações Finais

Neste capítulo foram apresentados estudos relevantes fundamentais para o desenvolvimento deste trabalho de pesquisa. Temas relacionados com o tema aqui desenvolvido (computação quântica, problemas NP-completos, simulação com computação simbólica) foram abordados de maneira a identificar variáveis e métodos imprescindíveis para a caracterização do problema.

O Quadro 3.1 exibe uma listagem resumida das abordagens apresentadas na Seção 3.2.1.

Quadro 3.1: Resumo dos trabalhos relacionados

Trabalho	**Escopo**	**Método**
(ARAÚJO; FINGER, 2011)	Definir SAT em termos quânticos	Comparação com SAT clássico; uso de álgebra linear
(WANG et al., 2012)	Testar quanticamente a satisfatibilidade de uma fórmula booleana	A partir do algoritmo Deutsch-Jozsa modificado com portas lógicas Toffoli estendidas
(OHYA; MASUDA, 1998)	Algoritmo quântico para resolver SAT em tempo polinomial	Circuito quântico composto por portas lógicas usuais; operadores projeção
(OHYA; VOLOVICH, 1999)	Algoritmo quântico para resolver SAT em tempo polinomial	Computação quântica em conjunto com a dinâmica caótica
(OHYA, 2005)	Algoritmo quântico para resolver SAT em tempo polinomial	Algoritmo quântico adaptativo em conjunto com entropia mútua quântica
(SONG, 2014)	Estudo de *P versus NP*	Sob a ótica de processos físicos determinísticos
(FARHI et al., 2001)	Resolver instâncias aleatórias de problemas NP-completos	Algoritmo quântico baseado na evolução adiabática
(FELDMANN, 2012)	Provar $P = NP$	A partir do uso da teoria probabilística Bayesiana
(LEPORATI; FELLONI, 2007)	Algoritmos quânticos para resolver 3-SAT eficientemente	Circuitos quânticos Fredkin com operador não-unitário; máquina de registradores quânticos; níveis de energia de uma membrana em um sistema quântico
(ABRAMS; LLOYD, 1998)	Solução eficiente para problemas P e NP-completos	Uso de operadores não-lineares
(FREEDMAN, 1998)	Relação entre os problemas P, NP e a Teoria Quântica de Campo Topológica	Através da manipulação de sistemas físicos
(AHARONOV; NAVEH, 2002)	Revisão de classes de complexidade	Relacionando NP, MA e QMA
(GAITAN; CLARK, 2014)	Solução polinomial para o problema de isomorfismo de grafos	Através da conversão do problema para um de otimização combinatória junto com o uso da evolução quântica adiabática

Fonte: O autor

4

Abordagens utilizadas

Aqui serão explicitadas as abordagens tidas como base para o desdobramento da pesquisa e desenvolvimento dos simuladores propostos.

4.1 Circuitos Quânticos

Devido à sua natureza análoga ao paradigma computacional clássico mais bem difundido, foi feita uma escolha por desenvolver este trabalho fundamentado no modelo de computação quântica de circuitos. Tal modelo se baseia na aplicação das portas lógicas quânticas, descritas na Seção 2.1.2, ao estado quântico inicial para a construção do circuito em conjunto com alguma operação capaz concluir a satisfatibilidade de uma instância do problema SAT.

4.1.1 Comentários sobre *Quantum Computing, NP-complete Problems and Chaotic Dynamics*

A primeira das abordagens que serviram de base para a pesquisa contida neste trabalho, bem como para o desenvolvimento dos simuladores simbólicos aqui propostos diz respeito ao texto contido em (OHYA; VOLOVICH, 1999). Nele, o autor argumenta que o problema da satisfatibilidade pode ser resolvido em tempo polinomial caso haja o uso em conjunto de um computador quântico com um amplificador dinâmico caótico baseado no mapa logístico.

Outro ponto debatido neste mesmo trabalho é relativo à uma possível implementação de um computador quântico caótico usando, para isso, a descrição de um computador atômico com portas lógicas descritas pelas equações *Hartree-Fock*. A discussão acerca da possível realização de tal computador está fora do escopo do presente trabalho e, por isso, não será contemplada pelo mesmo.

Neste seção serão expostos os principais aspectos do artigo citado acima, bem como serão extraídos trechos de definições do mesmo. Porém, para a construção do circuito quântico que analisa uma instância do SAT foram usadas os detalhes explicitados em (OHYA; MASUDA, 1998).

- **Motivação:** Comumente, a decoerência quântica e a dinâmica caótica são vistas como efeitos indesejados em um sistema, os quais conduzem o mesmo a produzir um aumento na taxa de erros com relação ao tamanho da entrada. Para o trabalho de (OHYA; VOLOVICH, 1999), é considerado que tais efeitos podem desempenhar um papel favorável à resolução dos problemas NP-completos em tempo polinomial. Isso seria possível através da amplificação dos vetores de saída do computador quântico que representam a análise da satisfatibilidade, uma vez que tais vetores podem ser muito pequenos e difíceis de detectar.

- **O problema SAT:** Dado um conjunto de literais $\{x_1, \bar{x_1}, \cdots, x_n, \bar{x_n}\}$ e uma fórmula booleana na forma de produto de somas, essa fórmula é dita satisfazível se existe uma atribuição de valores aos literais tal que a mesma tenha valor 1. Dessa forma, é dada uma formulação analítica para tal problema:

  Seja f_α uma família de polinômios booleanos, onde:

  $$\alpha = \{S_1, ..., S_N, T_1, ..., T_N\},$$

  e

  $$S_i, T_i \subseteq \{1, ..., n\}.$$

  Então, f_α é definida como:

  $$f_\alpha(x_1, \cdots, x_n) = \prod_{i=1}^{N} \left(1 + \prod_{a \in S_i} (1 + x_a) \prod_{b \in T_i} x_b\right)$$

  Aqui é assumida a *soma módulo 2*. O problema é determinado pela existência ou não de atribuições à x_n tal que $f_\alpha = 1$.

- **Algoritmo quântico:** No espaço de Hilbert $H \equiv \otimes_1^{n+1} \mathbb{C}^2$ que abrigará o vetor $|x_1, ..., x_n, y\rangle = \otimes_{i=1}^{n} |x_i\rangle \otimes |y\rangle = |\mathbf{x}, y\rangle$, tem-se que $x_1, ..., x_n, y = 0$ ou 1. A aplicação do operador Hadamard para produzir a superposição de todas as possíveis atribuições para as variáveis booleanas resulta em:

  $$|v\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}} |\mathbf{x}, 0\rangle$$

  O operador unitário responsável por avaliar a superposição acima é definido pela matriz unitária U_f e gera:

  $$|v_f\rangle = U_f |v\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}} |\mathbf{x}, f(\mathbf{x})\rangle$$

Ao aplicar o projetor $P = I \otimes |1\rangle\langle 1|$ ao estado $|v_f\rangle$, ou seja, uma medição no último qubit, é obtida a probabilidade de encontrar $f(\mathbf{x}) = 1$, que é $||P|v_f\rangle||^2 = \frac{r}{2^n}$, onde r é o número de raízes que satisfaz a equação para $f(\mathbf{x}) = 1$. Porém, quanto menor r, menor é a probabilidade de obter alguma informação do sistema quântico.

Após a aplicação de U_f, o computador quântico estará no estado:

$$|v_f\rangle = \sqrt{1-q^2}|\varphi_0\rangle \otimes |0\rangle + q|\varphi_1\rangle \otimes |1\rangle,$$

onde $|\varphi_0\rangle$ e $|\varphi_1\rangle$ são estados de n qubits normalizados e $q = \sqrt{\frac{r}{2^n}}$.

De um modo reduzido, o problema se torna de 1 qubit:

$$|\psi\rangle = \sqrt{1-q^2}|0\rangle + q|1\rangle,$$

e concentra-se em distinguir entre os casos $q = 0$ e $q > 0$. Sugere-se que esta distinção pode ser alcançada de maneira satisfatória com o emprego da dinâmica caótica.

- **Circuito quântico:** O operador unitário U_f é construído a partir de várias portas lógicas quânticas usuais, tais como: *Not*, *Controlled-Not* e *Controlled-Controlled-Not*. A matriz de U_f é formada a partir da combinação das matrizes unitárias das portas lógicas citadas.

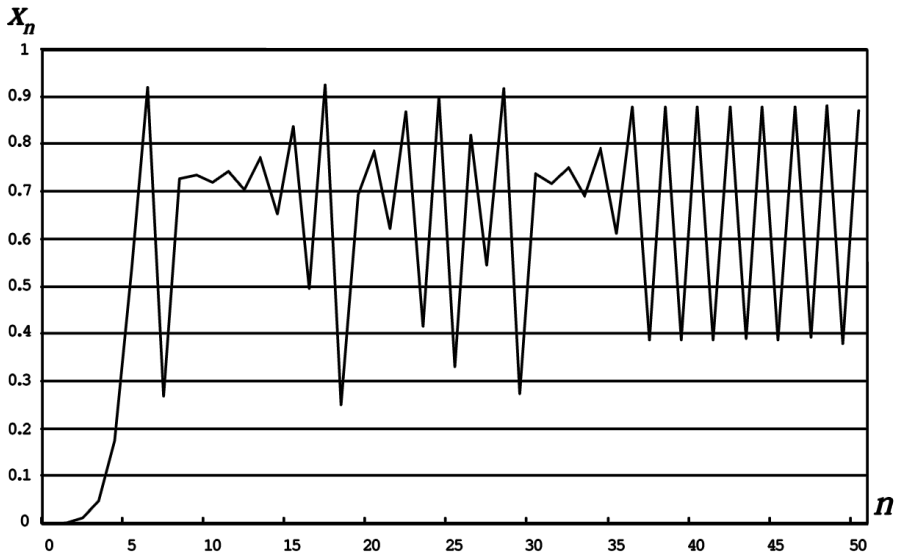
  Um ponto a ser ressaltado é que a topologia do circuito quântico que avalia uma fórmula boolena é diferente para cada fórmula. Outra questão diz respeito à necessidade de qubits adicionais. Assim, o espaço de Hilbert onde U_f opera é $\otimes^n \mathbb{C}^2 \otimes^l \mathbb{C}^2 \otimes \mathbb{C}^2$, onde n é a quantidade de variáveis booleanas, l é o número de qubits adicionais e o último qubit armazenará a saída $f(\mathbf{x})$.

- **Dinâmica caótica:** Dado que o comportamento caótico em um sistema representa uma sensibilidade às condições iniciais, essa característica pode ser usada para distinguir entre $q = 0$ e $q > 0$. Agora, dado o seguinte mapa logístico:

$$x_{n+1} = a x_n (1 - x_n), \qquad x_n \in [0, 1]$$

onde a é o parâmetro que leva a equação acima a produzir um comportamento caótico como na Figura 4.1. Para o valor inicial $x_0 = 0$, então $x_n = 0$ para toda unidade de tempo n.

Figura 4.1: Mapa logístico - mudança de x_n no tempo n



Fonte: (OHYA; VOLOVICH, 1999)

Para a proposta do autor, o computador quântico em questão seria formado por 2 blocos. O primeiro teria como saída o estado $|\psi\rangle = \sqrt{1-q^2}|0\rangle + q|1\rangle$, o qual é transformado em uma matriz densidade da forma:

$$\rho = q^2 P_1 + (1-q^2)P_0,$$

onde P_0 e P_1 são projetores para os estados $|0\rangle$ e $|1\rangle$. O segundo bloco é um computador quântico realizando cálculos do mapa logístico clássico e a matriz densidade ρ é interpretada como o dado inicial:

$$\rho_{n+1} = a\rho_n(1-\rho_n)$$

Depois de 1 passo, o sistema será:

$$\rho_1 = aq^2(1-q^2)I,$$

onde I é a matriz identidade em $\mathbb{C}^2$. Se $q > 0$, então a dinâmica caótica amplificará a magnitude de q de tal forma que seja possível detectá-lo. A transição de ρ_n para ρ_{n+1} é não-linear.

4.1.2 Comentários sobre *Three "quantum" algorithms to solve 3-SAT*

O segundo tratamento quântico dado aos problemas NP-completos usado neste trabalho é o encontrado em (LEPORATI; FELLONI, 2007). Na proposta apresentada pelos autores, são

apontados 3 algoritmos quânticos para o problema **3-SAT** - o qual é um problema NP-completo e, também, um caso especial do problema da satisfatibilidade, onde cada cláusula possui exatamente 3 literais.

Da mesma forma que na seção anterior, aqui serão exibidas as particularidades da solução idealizada por (LEPORATI; FELLONI, 2007), bem como será feito uso de trechos do artigo para explicitar seus detalhes.

Os conceitos dos 3 algoritmos são, dada uma instância ϕ do 3-SAT:

1. Um circuito quântico Fredkin que computa uma superposição de todas as atribuições clássicas possíveis para as variáveis e retorna uma saída, na qual será aplicado um *operador não-unitário* usado como um *operador de seleção* capaz de decidir sobre a satisfatibilidade da fórmula booleana em questão.

2. O segundo algoritmo computa a mesma superposição de atribuições para as variáveis, porém, em uma máquina utilizando registradores quânticos. O operador aqui empregado é visto como uma *instrução* na máquina citada.

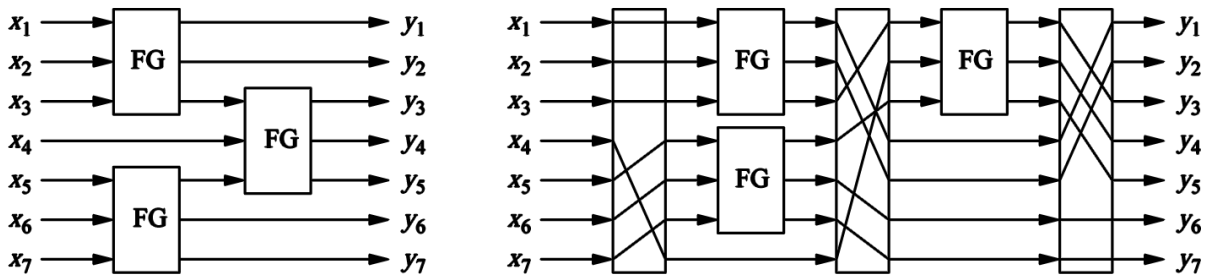
3. Por último, tem-se a superposição calculada como a energia de uma membrana em um sistema quântico P, no qual o operador usado é concebido como uma *regra* desse sistema.

Os conceitos acima tomam como suposição a existência de um observador externo capaz de discriminar um vetor nulo de um vetor não-nulo. Tal suposição tornaria desnecessária a aplicação da amplificação usada na proposta explicitada na Seção 4.1.1, porém, justamente por ser uma suposição, ainda é necessário o estudo de abordagens que não levam em conta este observador externo. E, para este trabalho, será investigado o algoritmo que faz uso de circuitos quânticos Fredkin.

É argumentado em (LEPORATI; FELLONI, 2007) que, a partir das portas lógicas Fredkin, é possível criar camadas de portas lógicas desse tipo responsáveis por computar as funções booleanas necessárias para avaliar uma instância ϕ do 3-SAT.

A Figura 4.2 ilustra um circuito composto por portas Fredkin.

Figura 4.2: Exemplo de circuito Fredkin e sua versão normalizada



Fonte: (LEPORATI; FELLONI, 2007)

Assim como na seção anterior, aqui também são necessários qubits adicionais para a realização das operações. O circuito Fredkin para avaliação de n variáveis é denotado por FC_n e cada porta lógica desse circuito é dita ser U_{FG}. Em conjunto com operadores identidade ID_m (sobre m qubits), as camadas são construídas computando o produto tensorial de ID_m com U_{FG}. Para a primeira camada L_1 da Figura 4.2, tem-se o seguinte operador:

$$U_{L_1} = U_{FG} \otimes ID_1 \otimes U_{FG}$$

Logo, seja l o número de camadas do circuito FC_n necessárias para computar uma instância ϕ com n variáveis, U_{FC_n} é a matriz unitária que equivale ao circuito FC_n como produto das matrizes $U_{L_1}, U_{L_2}, ..., U_{L_l}$ associadas às l camadas de FC_n:

$$U_{FC_n} = U_l \cdot ... \cdot U_2 \cdot U_l$$

Portanto, dada uma instância ϕ_n do 3-SAT, com n variáveis, existe um circuito Fredkin FC_m (com $m > n$) que computa ϕ_n.

O primeiro passo para a avaliação da fórmula consiste na aplicação da porta lógica Hadamard a qual cria uma superposição de todas as atribuições clássicas possíveis para as n variáveis como segue:

$$H_n = \otimes^n H_1 = \frac{1}{\sqrt{2^m}} \otimes^n \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

o que produz:

$$H_n = |0 \cdots 0\rangle = \otimes^n H_1 |0\rangle = \otimes^n \frac{1}{\sqrt{2^n}} (|0\rangle + |1\rangle) = \frac{1}{\sqrt{2^n}} \sum_{x_1,...,x_n \in \{0,1\}} |x_1, ..., x_n\rangle$$

Em uma das linhas da saída do circuito FC_m será computado $f(\mathbf{x})$ para $\mathbf{x} = (x_1, x_2, ..., x_n)$. Nessa saída, um dos dois seguintes resultados possíveis será obtido:

1. $|0\rangle$ se ϕ_n não é satisfatível
2. Uma combinação linear $\alpha_0|0\rangle + \alpha_1|1\rangle$ (com $\alpha_1 \neq 0$) se ϕ_n é satisfatível

Assim, o problema agora consiste em realizar uma medição no qubit que computa $f(\mathbf{x})$ e observar o estado $|i\rangle$, com $i \in \{0,1\}$, o qual tem probabilidade associada de $|\alpha|^2$. Porém, esta probabilidade pode ser extremamente pequena, necessitando (no pior caso) executar um número exponencial de computações e sucessivas medições.

Portanto, com a finalidade de contornar essa questão, é considerado que, caso seja possível construir o operador linear O representado pela seguinte matriz não-unitária:

$$2^n \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} = 2^n |1\rangle\langle 1|,$$

então também é possível construir o *operador seleção* $O^{(m)}$:

$$O^{(m)} = O \otimes ID_{m-1} = O \otimes (\otimes^{m-1} ID_1)$$

Ou seja, esse operador seleção faz com que o operador O tenha ação sobre uma das saídas do circuito enquanto o operador identidade ID age em todos as outras linhas de saída.

Dessa forma, ao final da computação tem-se o seguinte:

$$O^{(m)} \cdot U_{FC_m} \cdot H_n |0 \cdots 0\rangle$$

E, ao observar o vetor resultante, uma das seguintes duas saídas possíveis será observada:

1. O vetor nulo 0, se ϕ_n não é satisfatível:

$$O|0\rangle = 2^n |1\rangle\langle 1|0\rangle = 0$$

2. Um vetor não-nulo se ϕ_n é satisfatível:

$$O(\alpha_0|0\rangle + \alpha_1|1\rangle) = \alpha_0 2^n |1\rangle\langle 1|0\rangle + \alpha_1 2^n |1\rangle\langle 1|1\rangle = 0 + \alpha_1 2^n |1\rangle = \alpha_1 2^n |1\rangle$$

É fundamental notar que o fato 2^n foi escolhido de forma que o vetor resultante não seja tão pequeno, possibilitando, assim, que o mesmo possa ser distinguível de um vetor nulo.

Por fim, e concluindo a concepção desta abordagem, pode-se concluir que se as seguintes condições forem satisfeitas, então é factível a existência de um dispositivo computacional quântico capaz de resolver 3-SAT em tempo polinomial:

1. Se for possível construir e aplicar o operador $2^n|1\rangle\langle 1|$ na saída do circuito Fredkin FC_m que possui o resultado de $f(\mathbf{x})$
2. A existência de um observador externo capaz de distinguir um vetor nulo de um vetor não-nulo

4.2 Discussão

Como visto por meio das descrições de ambas as soluções propostas, algumas suposições são indispensáveis para que a classe de problemas NP-completos possa ser efetivamente resolvida em tempo polinomial. Tais pressupostos corroboram a ideia de que a concepção da computação

quântica possui, ainda, alguns entraves. Isto reforça a necessidade do estudo deste modelo computacional, tal como a utilidade que as simulações provêm ao mesmo.

Sobre a construção dos circuitos propostos nas duas abordagens, o mesmo possui uma topologia não geral. Ou seja, para cada instância do problema, um diferente circuito será criado/ativado naquele momento. Considera-se que esta tarefa é de responsabilidade do projetista do circuito no hardware.

Tendo isso em vista, no Capítulo 5, serão detalhados os principais aspectos relativos ao desenvolvimento dos códigos que simulam os circuitos aqui apresentados. Os mesmos são responsáveis por avaliar instâncias do problema da satisfatibilidade de maneira condizente com o que foi descrito neste capítulo.

5

Descrição dos Simuladores

Como dito no Capítulo 4, para este trabalho, foram escolhidas as soluções apontadas em (OHYA; VOLOVICH, 1999) e em (LEPORATI; FELLONI, 2007). Visto que nenhum dos dois artigos apresentam um método para a construção do circuito nem um exemplo de computação para uma fórmula booleana qualquer, os códigos desenvolvidos para tal nesta pesquisa foram implementados a partir do entendimento do presente autor e se configuram como uma das contribuições deste trabalho. Outro ponto a ser destacado diz respeito à ausência no SymPy/Quantum de alguns operadores apresentados em ambas as abordagens e, à vista disso, foi necessário criá-los de forma matricial a fim de tornar factível a simulação das abordagens.

Os pseudocódigos presentes na Seção 5.2 estão expostos de maneira simplificada visando, assim, uma melhor compreensão do procedimento em si.

5.1 Comandos Python/SymPy

Aqui, serão listados os principais comandos em Python/SymPy, com o suporte do pacote *Quantum*, que foram utilizados para a implementação dos simuladores propostos.

Primeiramente, tem-se que importar as classes que contém os métodos necessários. Os comandos seguintes dizem respeito à criação de símbolos, qubits, aplicação de portas lógicas e uso de matrizes para criação de estruturas não presentes no Quantum.

Imports

```
>>> from sympy import *
>>> from sympy.physics.quantum.qapply import qapply
>>> from sympy.physics.quantum.tensorproduct import TensorProduct
>>> from sympy.physics.quantum.qubit import Qubit, matrix_to_qubit,
qubit_to_matrix, matrix_to_density, measure_partial
>>> from sympy.physics.quantum.gate import CGate, XGate, HadamardGate,
SwapGate
>>> from sympy.physics.quantum.circuitplot import CircuitPlot
>>> from matplotlib import *
```

Fórmula e Símbolos

```
>>> x,y,z = symbols('x,y,z')
#criação dos símbolos/variáveis booleanas
>>> formula = [[x],[y,z],[x,~z],[~x,~y,z]]
#fórmula como lista de listas
>>> Not(x).free_symbols.pop()
#retorna o símbolo livre da variável booleana
```

Qubit e Produto Tensorial

```
>>> qubit = Qubit('0')
#cria o qubit |0>
>>> Qubit('1')*Dagger(Qubit('1'))
# retorna |1><1|
>>> qbit.dimension()
#retorna o número de qubits no estado
>>> TensorProduct(Qubit('0'), Qubit('0'))
#realiza o produto tensorial entre dois qubits e retorna |0>x|0>
```

Portas lógicas e Aplicação

```
>>> qapply(HadamardGate(0)*Qubit('0'))
#aplica a porta lógica Hadamard ao qubit de índice 0, retornando
sqrt(2)*|0>/2 + sqrt(2)*|1>/2
>>> qapply(XGate(0)*Qubit('0'))
#aplica a porta lógica X (not) ao qubit de índice 0, retornando |1>
>>> qapply(CGate(0, XGate(1))*Qubit('01'))
#porta lógica CNOT - qubits de controle é o de índice 0 e qubit alvo é o
de índice 1 (onde é aplicada a porta lógica XGate)
>>> qapply(CGate(0), SwapGate(1,2)*Qubit('010'))
# porta lógica Fredkin - qubit de índice 0 controla a troca entre os
valores dos qubits de índice 1 e 2
```

Uso de matrizes

```
>>> qubit_to_matrix(Qubit('0'))
# retorna Matrix([[1], [0]])
>>> matrix_to_qubit(Matrix([[1],[0]]))
# retorna |0>
>>> matrix_tensor_product(A, B)
# comando específico para realização do produto tensorial entre objetos na
forma de matriz
>>> matrix_to_density(m)
# cria um objeto Density a partir de uma matriz
```

Gráficos e diagramas

```
>>> CircuitPlot(circuito, n, labels)
# exibe o diagrama do circuito, com n qubits e rotulados pelas labels
>>> plot(x, y)
# x e y são listas com pontos a serem plotados no gráfico
```

5.2 Pseudocódigos

Tendo em vista a sintaxe dos comandos do Python/SymPy expostos na seção anterior, esta seção trata da implementação dos procedimentos que: analisam uma fórmula na FNC, cria o qubit inicial, cria o circuito que avalia a instância apresentada e, por fim, aplica os operadores responsáveis por concluir a satisfatibilidade da mesma.

Aqui, serão apresentados os pseudocódigos que representam os procedimentos citados. Apesar de o Python apresentar uma sintaxe limpa e de fácil compreensão, a escolha por pseudocódigos se deu em virtude de o mesmo prover uma linguagem genérica e universal, possibilitando que possam ser implementados em qualquer linguagem de programação que contenha os conceitos quânticos necessários. Além disso, os pseudocódigos aqui estão apresentados de forma sucinta, excluindo algumas operações de checagem de variáveis e procedimentos auxiliares, focando no objetivo do procedimento em si.

O presente autor não considera que há prejuízo em omitir os códigos completos em Python, uma vez que os comandos essenciais e as peculiaridades quânticas presentes no mesmo estão expostas na seção anterior e, também, no Quadro 2.1.

Algumas observações sobre os pseudocódigos e os diagramas:

- Como dito, os pseudocódigos demonstram os passos que foram seguidos para o desenvolvimento dos códigos. Algumas manipulações necessárias para contornar eventuais empecilhos provocados pela ausência de alguma estrutura no SymPy foram omitidas dos mesmos, visando apresentar de forma clara e limpa os procedimentos
- "ancillas" são qubits auxiliares necessários para a realização da computação
- Os diagramas foram obtidos com a classe `circuitplot` do SymPy, mas foram transcritos para a linguagem LaTeX com o pacote *Q-circuit*[1] para uma representação mais limpa e compacta

5.2.1 Computação quântica e dinâmica caótica

A abordagem exposta em (OHYA; VOLOVICH, 1999) foi simulada simbolicamente através da implementação dos pseudocódigos a seguir:

[1] http://physics.unm.edu/CQuIC/Qcircuit/

Pseudocódigo 1 *Analisa fórmula - Abordagem circuito+caos*

```
1: Procedimento ANALISA-FÓRMULA(fórmula)
2:     listaLiterais ← {}
3:     ancilla ← 0
4:     Para cláusula em fórmula Faça
5:         Se tamanho(cláusula) > 1 Então
6:             ancilla ← ancilla + 1
7:         Fim
8:         Para literal em cláusula Faça
9:             listaLiterais ← listaLiterais ∪ {símbolo livre do literal}
10:        Fim
11:    Fim
12:    espaço ← ancilla + tamanho(listaLiterais) + 1   ▷ Armazena a dimensão do espaço de
   Hilbert necessária para computar a fórmula no circuito quântico
13:    Retorne listaLiterais, espaço
14: Fim
```

Fonte: O autor

O pseudocódigo 1 armazena em uma lista os literais que compõem a fórmula booleana sem levar em consideração se estão negados ou não, tomando-se apenas o símbolo livre e sem repetições. Assim, com a quantidade de literais e a quantidade de qubits auxiliares decorrentes do tamanho da fórmula, é calculado o espaço necessário para o estado quântico inicial.

Pseudocódigo 2 *Cria qubit - Abordagem circuito+caos*

```
1: Procedimento CRIA-QUBIT(n)
2:     qubit ← |0⟩
3:     Para 1...n − 1 Faça
4:         qubit ← ProdutoTensorial(qubit, |0⟩)
5:     Fim
6:     Para i de 1...tamanho(listaLiterais) Faça
7:         qubit ← HadamardGate(i) ▷ Cria a superposição a partir da quantidade de literais na
   fórmula
8:     Fim
9:     Retorne qubit
10: Fim
```

Fonte: O autor

Então, uma vez calculado o espaço necessário para computar a fórmula, tem-se no pseudocódigo 2 a criação do estado inicial bem como a aplicação da porta lógica Hadamard nos qubits referentes aos literais para a criação da superposição das possíveis valorações que podem ser assumidas pelos literais.

O pseudocódigo 3 apresenta o algoritmo seguido para a construção do circuito que computa a fórmula booleana. Foi feito uso de um dicionário para simular a memória e armazenar dados provisórios e auxiliares. Assim, a partir da análise das cláusulas e literais da fórmula, são guardados os índices dos qubits que estarão envolvidos na operação das portas lógicas para, então, aplicar os circuitos relativos às funções AND e OR a estes qubits e computar a instância do problema através do circuito.

Pseudocódigo 3 *Cria circuito - Abordagem circuito+caos*

```
1:  Procedimento CRIA-CIRCUITO(fórmula)
2:      ANALISA-FÓRMULA(fórmula)
3:      CRIA-QUBIT(espaço)
4:      memória = {'AND':[], 'OR:[]'}   ▷ Armazena posições de variáveis para uso posterior
5:      ancillaIndex ← dimensão(Qubit) - tamanho(listaLiterais) - 1
6:      Para i de 1...tamanho(listaLiterais) Faça
7:          memória[literal] = espaço - i
8:      Fim
9:      Para cláusula em fórmula Faça
10:         Se tamanho(cláusula)=1 Então
11:             'AND' ← ∪ {memória[cláusula]}
12:         Senão
13:             Para literal em cláusula Faça
14:                 'OR' ← ∪ {memória[literal]}
15:             Fim
16:             circuito ← ∪ {CGate('OR', XGate(ancillaIndex)) * XGate(ancillaIndex)}
17:             'AND' ← ∪ {ancillaIndex}
18:             ancillaIndex ← ancillaIndex - 1
19:         Fim
20:     Fim
21:     circuito ← ∪ {CGate('AND', XGate(ancillaIndex))}
22: Fim
```

Fonte: O autor

No pseudocódigo 4, $(1-p(1))$ e $p(1)$ são as probabilidades de se encontrar $|0\rangle$ ou $|1\rangle$ no último qubit do estado resultante do circuito. Estes valores são obtidos através de uma *medição parcial* (no SymPy). Como dito no texto, este último passo para concluir a satisfatibilidade não é trivial, então, a fim de computar o mapa logístico a partir da matriz densidade ρ, os valores contidos nela são usados como os valores iniciais para a amplificação de valores pequenos de $p(1)$.

Pseudocódigo 4 *Aplica circuito e operador - Abordagem circuito+caos*

```
1: Procedimento APLICA-CIRCUITO-OPERADOR(circuito)
2:     qubit ← aplica(circuito * qubit)          ▷ Aplica o circuito gerado ao qubit inicial
3:     a ← 3.71                                  ▷ Valor sugerido no texto
4:     estado ← medição_parcial(qubit, 0) ▷ Medição parcial com base em um índice do qubit
   (índice 0)
5:     P_0 ← |0⟩⟨0|                              ▷ Projetor para o estado |0⟩
6:     P_1 ← |1⟩⟨1|                              ▷ Projetor para o estado |1⟩
7:     ρ ← (1 − p(1))P_0 + p(1)P_1               ▷ Matriz densidade
8:     Para i de 1...100 Faça
9:         ρ_i ← a × p(1) × (1 − p(1))
10:        p(1) ← ρ_i
11:    Fim
12: Fim
```

Fonte: O autor

5.2.2 Circuito quântico com portas lógicas Fredkin

Pseudocódigo 5 *Analisa fórmula - Abordagem Fredkin*

```
1:  Procedimento ANALISA-FÓRMULA(fórmula)
2:      listaLiterais ← {}
3:      ancilla ← 0
4:      Para cláusula em fórmula Faça
5:          Se tamanho(cláusula) > 1 Então
6:              ancilla ← ancilla + 2
7:          Fim
8:          Para literal em cláusula Faça
9:              listaLiterais ← listaLiterais ∪ {símbolo livre do literal}
10:         Fim
11:     Fim
12:     ancilla ← ancilla + 2
13:     espaço ← ancilla + tamanho(listaLiterais) ▷ Armazena a dimensão do espaço de Hilbert
    necessária para computar a fórmula no circuito quântico
14:     Retorne listaLiterais, espaço
15: Fim
```

Fonte: O autor

Para o pseudocódigo 5 tem-se praticamente o mesmo do pseudocódigo 1. A única diferença consiste na forma como os qubits auxiliares são contados (os *ancillas*); aqui, é necessário um maior número desses qubits para realizar a computação.

Pseudocódigo 6 *Cria qubit - Abordagem Fredkin*

```
1:  Procedimento CRIA-QUBIT(n, tamanho(listaLiterais))
2:      qubit ← |0⟩
3:      Para 1...n − 1 Faça
4:          qubit ← ProdutoTensorial(qubit, |0⟩)
5:      Fim
6:      Para i de 1...tamanho(listaLiterais) Faça
7:          qubit = HadamardGate(i) ▷ Cria a superposição a partir da quantidade de literais na
    fórmula
8:      Fim
9:      Retorne qubit
10: Fim
```

Fonte: O autor

Aqui, no pseudocódigo 6, é exposto o mesmo procedimento feito no pseudocódigo 2.

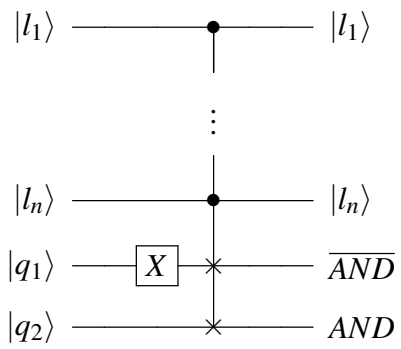
Pseudocódigo 7 *Porta lógica AND-Fredkin - Abordagem Fredkin*

1: **Procedimento** AND-FREDKIN($[nQubits], q_1, q_2$)
2: **Retorne** $ControlledGate([nQubits], Swap(q_1, q_2)) * Not(q_1)$
3: **Fim**

Fonte: O autor

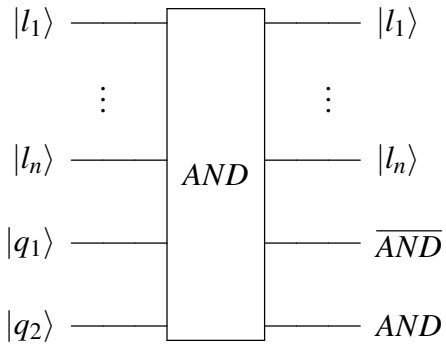
A aplicação do pseudocódigo 7 resulta no seguinte circuito - onde de l_1 até l_n são os literais nos quais é feito o AND e q_1 e q_2 são ancillas; todos são iniciados no estado $|0\rangle$:

Figura 5.1: Operação AND feita com portas lógicas Fredkin



Fonte: O autor

Figura 5.2: Operação AND feita com portas lógicas Fredkin - versão simplificada



Fonte: O autor

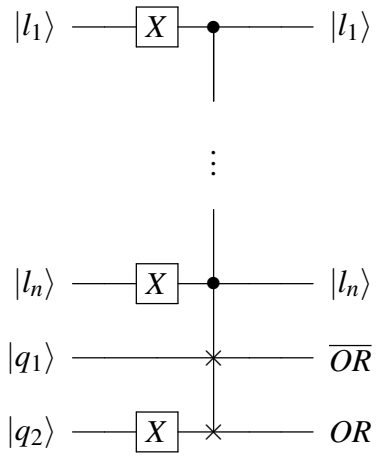
Pseudocódigo 8 *Porta lógica OR-Fredkin - Abordagem Fredkin*

1: **Procedimento** OR-FREDKIN($[nQubits], q_1, q_2$)
2: **Retorne** $ControlledGate([nQubits], Swap(q_1, q_2)) * Not(q_2) * Not([nQubits])$
3: **Fim**

Fonte: O autor

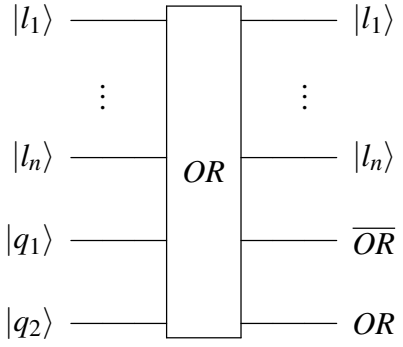
A aplicação do pseudocódigo 8 resulta no seguinte circuito - onde de l_1 até l_n são os literais nos quais é feito o OR e q_1 e q_2 são ancillas; todos são iniciados no estado $|0\rangle$:

Figura 5.3: Operação OR feita com portas lógicas Fredkin



Fonte: O autor

Figura 5.4: Operação OR feita com portas lógicas Fredkin - versão simplificada



Fonte: O autor

O pseudocódigo 9 possui o mesmo objetivo do pseudocódigo 3, porém, desta vez, é feito uso das portas lógicas Fredkin para realizar as operações AND e OR.

Pseudocódigo 9 *Cria circuito - Abordagem Fredkin*

```
1:  Procedimento CRIA-CIRCUITO(fórmula)
2:      ANALISA-FÓRMULA(fórmula)
3:      CRIA-QUBIT(espaço)
4:      memória = {'AND':[], 'OR':[]}   ▷ Armazena posições de variáveis para uso posterior
5:      ancillaIndex ← dimensão(Qubit) - tamanho(listaLiterais) - 1
6:      Para i de 1...tamanho(listaLiterais) Faça
7:          memória[literal] = espaço - i
8:      Fim
9:      Para cláusula em fórmula Faça
10:         Se tamanho(cláusula) = 1 Então
11:             'AND' ← ∪ {memória[cláusula]}
12:         Senão
13:             Para literal em cláusula Faça
14:                 'OR' ← ∪ {memória[literal]}
15:                 ancillaIndex ← ancillaIndex - 2
16:             Fim
17:             circuito ← ∪ {OR-FREDKIN('OR', ancillaIndex_1, ancillaIndex_2)}
18:             'AND' ← ∪ {ancillaIndex_2}
19:         Fim
20:     Fim
21:     circuito ← ∪ {AND-FREDKIN('AND', ancillaIndex_1, ancillaIndex_2)}
22: Fim
```

Fonte: O autor

Por fim, tem-se o procedimento descrito no pseudocódigo 10, onde é realizada a aplicação do operador descrito em (LEPORATI; FELLONI, 2007) capaz de decidir pela satisfatibilidade ou não da fórmula booleana configurada como uma instância do problema.

Pseudocódigo 10 *Aplica circuito e operador - Abordagem Fredkin*

```
1:  Procedimento APLICA-CIRCUITO-OPERADOR(circuito)
2:      qubit ← aplica(circuito * qubit)
3:      O ← 2^n|1⟩⟨1|                    ▷ Qubit |1⟩ operado com seu conjulgado transposto
4:      I ← ([[1,0],[0,1]])                                          ▷ Matriz identidade
5:      Para i de 1...espaço-2 Faça
6:          I = ProdutoTensorial(I, [[1,0],[0,1]])
7:      Fim
8:      O^(m) ← ProdutoTensorial(I, O)
9:      aplica(O^(m) * qubit)
10: Fim
```

Fonte: O autor

5.3 Observações

Seguem algumas ponderações acerca do que foi apresentado nesta seção:

- Algo que deve ser ressaltado é que, a cada aplicação de uma porta lógica no circuito quântico, os qubits que não estão sendo afetados por esta porta tem seus valores operados pela porta lógica identidade. Estas aplicações da identidade são omitidas dos diagramas do circuito visando não sobrecarregar a representação do mesmo, uma vez que tais portas, como mostrado na Seção 2.1.2, não atuam de modo a alterar os valores de saída dos qubits.

- Devido à falta dos operadores encontrados em (OHYA; VOLOVICH, 1999) e (LEPORATI; FELLONI, 2007) no SymPy e consequente implementação dos mesmos na sua forma matricial, parte do código acabou por fugir da proposta simbólica. Uma possível solução para este contratempo trata da expansão do pocote quântico e é citada na Seção 7.1, de **Trabalhos Futuros**.

- Importante salientar que a complexidade dos simuladores não foi avaliada devido ao seu caráter de simulação simbólica, a qual faz com que características que seriam implementadas em hardware estão sendo simuladas através de software, o que implica em uma ordem de complexidade diferente da encontrada em uma possível implementação real.

- No caso da implementação do trabalho contido em (LEPORATI; FELLONI, 2007), era possível construir o circuito apenas com portas lógicas fredkin, porém, aumentaria muito a dimensão do circuito uma vez que seriam necessários mais bits auxiliares. Portanto, foi decidido por usar as portas Fredkin apenas para computar as operações lógicas AND e OR, ao passo que outras ações foram realizados com portas lógicas usuais.

6

Simulações

6.1 Metodologia

Para a validação dos simuladores simbólicos desenvolvidos foram usadas diversas instâncias do problema SAT e 3-SAT. Porém, para ilustrar os circuitos que computam tais fórmulas, na próxima seção são dados os diagramas dos circuitos responsáveis por avaliar as seguintes fórmulas booleanas (a primeira é uma instância geral do SAT e a segunda, uma instância do 3-SAT):

$$\textit{Instância } 1 - (x) \wedge (y \vee z) \wedge (x \vee \bar{z}) \wedge (\bar{x} \vee \bar{y} \vee z)$$

$$e$$

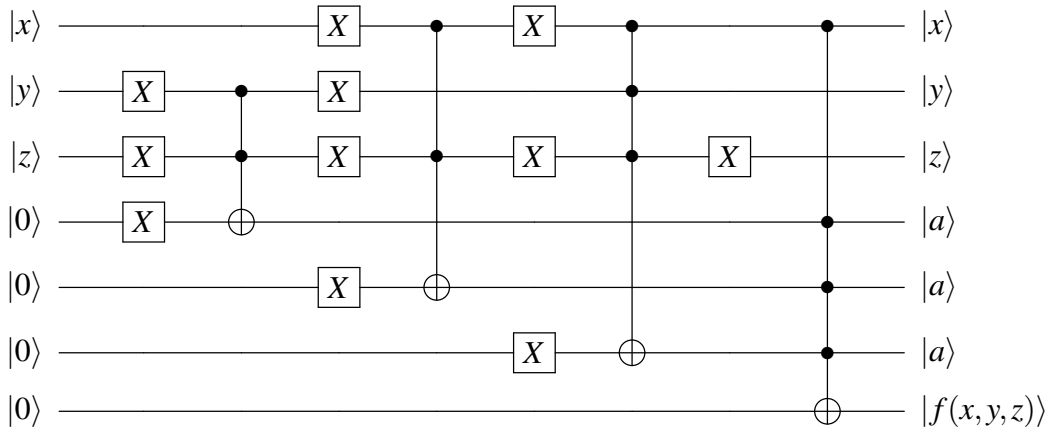
$$\textit{Instância } 2 - (x \vee y \vee z) \wedge (x \vee \bar{y} \vee z) \wedge (\bar{x} \vee y \vee \bar{z})$$

Dessa forma, para cada um dos simuladores, cada uma dessas 2 instâncias do problema foi inserida como a entrada. Pode-se observar na saída dos circuitos o qubit $|a\rangle$; o mesmo representa uma saída que não tem importância para o desenrolar da abordagem, sendo, portanto, $|a\rangle \in \{0,1\}$. Quando da representação simplificada da aplicação da porta lógica AND, a simbologia $AND_{C_{1,\ldots,n}}$ representa a operação aplicada às n cláusulas da fórmula.

6.2 Diagramas dos circuitos

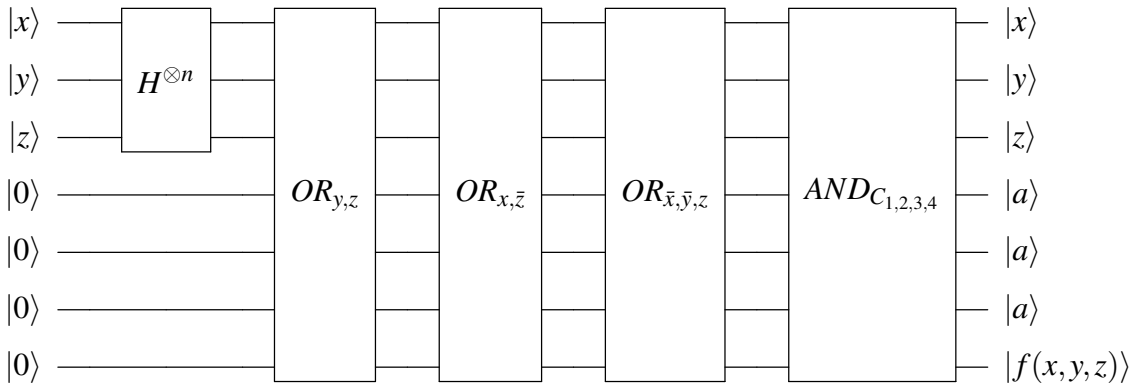
A Figura 6.1 representa o circuito que analisa a fórmula 1; a Figura 6.2 representa sua forma simplificada.

Figura 6.1: Circuito com portas lógicas usuais na CQ que avalia a **Instância 1**



Fonte: O autor

Figura 6.2: Circuito com portas lógicas usuais na CQ que avalia a **Instância 1** - versão simplificada



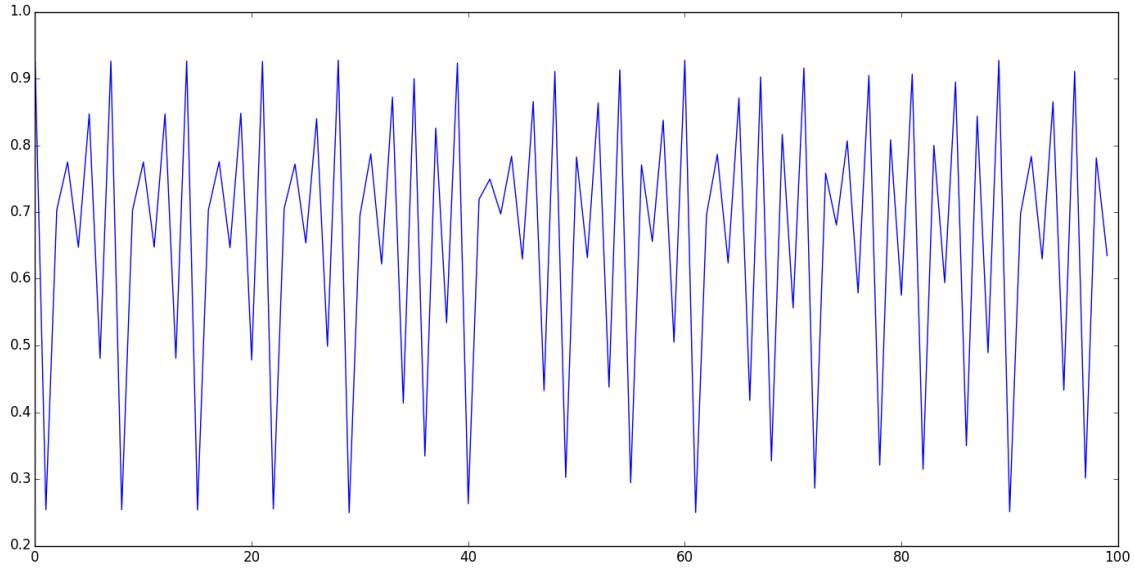
Fonte: O autor

A saída do circuito será:

$$|\psi\rangle = \frac{1}{2\sqrt{2}}(|0000110\rangle + |0011010\rangle + |0101110\rangle + |0111010\rangle + |1000110\rangle + |1011111\rangle + |1101100\rangle + |1111111\rangle)$$

Este estado quântico será a entrada para o pseudocódigo 4, o qual resultará em uma probabilidade de $1/4$ de encontrar o qubit $|1\rangle$ na última posição do estado acima. Como esse resultado pode ser muito pequeno para ser medido, a aplicação no mapa logístico gera o seguinte comportamento caótico:

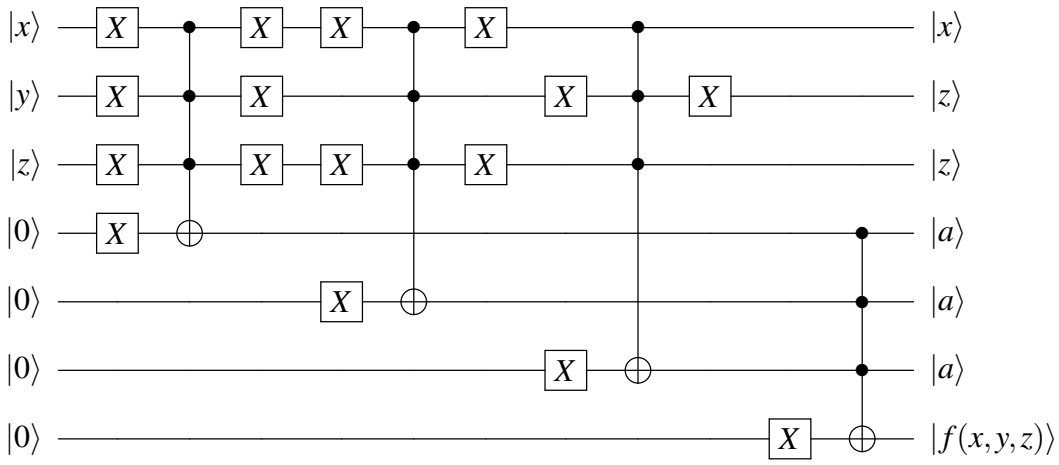
Figura 6.3: Comportamento caótico para a **Instância 1**



Fonte: O autor

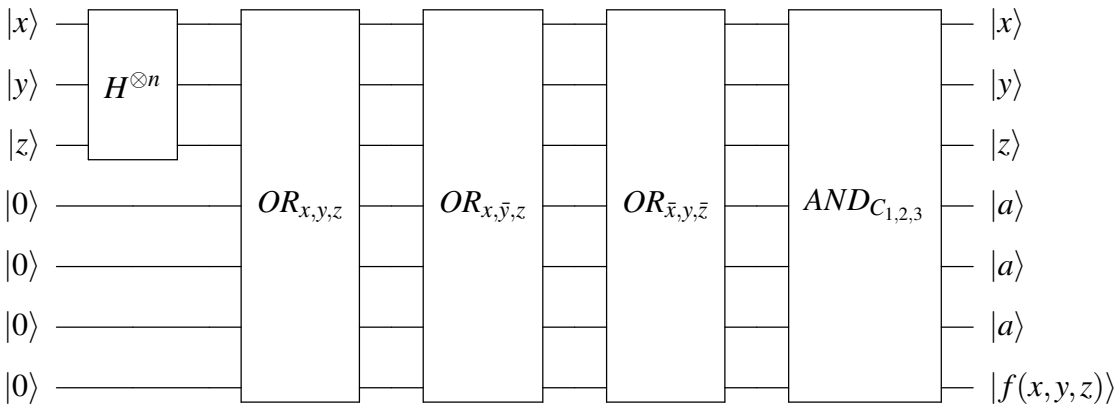
A Figura 6.4 representa o circuito que analisa a fórmula 2; a Figura 6.5 representa sua forma simplificada.

Figura 6.4: Circuito com portas lógicas usuais na CQ que avalia a **Instância 2**



Fonte: O autor

Figura 6.5: Circuito com portas lógicas usuais na CQ que avalia a **Instância 2** - versão simplificada



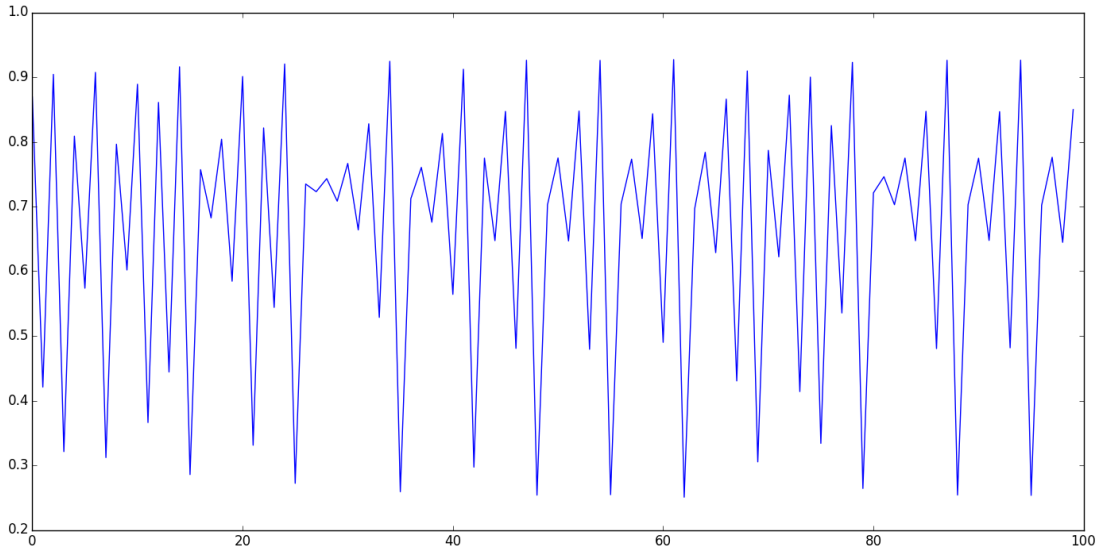
Fonte: O autor

A saída do circuito será:

$$|\psi\rangle = \frac{1}{2\sqrt{2}}(|0000110\rangle + |0011111\rangle + |0101010\rangle + |0111111\rangle + |1001111\rangle + |1011100\rangle + |1101111\rangle + |1111111\rangle)$$

Novamente, este estado quântico será a entrada para o pseudocódigo 4, o qual resultará em uma probabilidade de $5/8$ de encontrar o qubit $|1\rangle$ na última posição do estado acima. Como esse resultado pode ser muito pequeno para ser medido, a aplicação no mapa logístico gera o seguinte comportamento caótico:

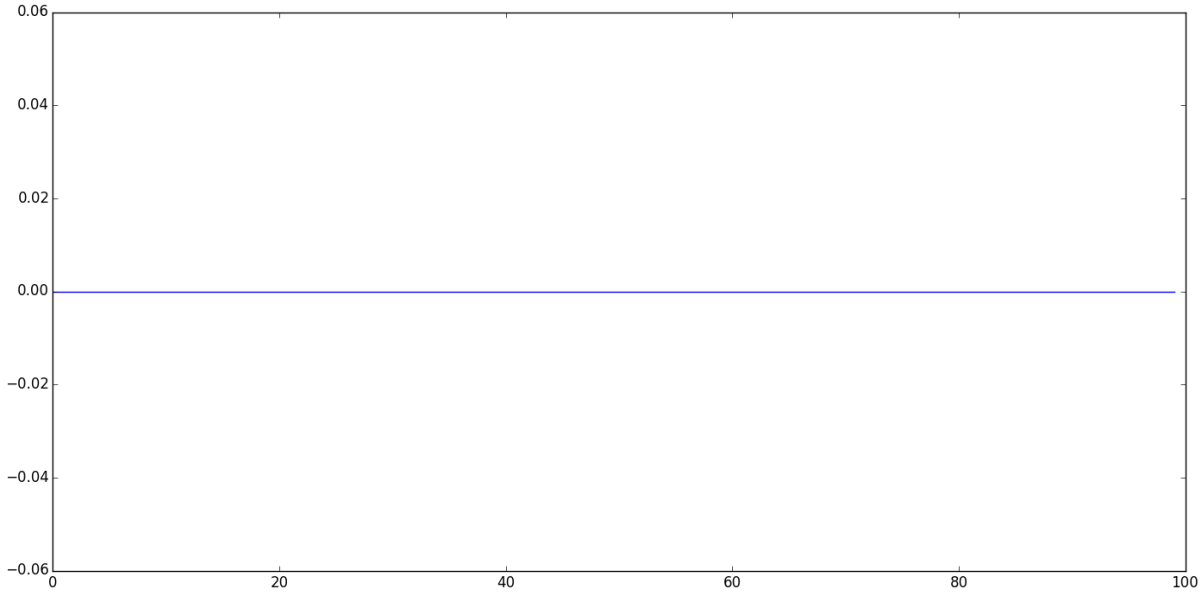
Figura 6.6: Comportamento caótico para a **Instância 2**



Fonte: O autor

Observa-se, então, que a aplicação da dinâmica caótica pode prover uma maior possibilidade de se obter a resposta correta para a satisfatibilidade da fórmula. Para o caso de uma instância do problema que não possua uma atribuição de valores para as variáveis que a torne satisfatível, o gráfico do comportamento caótico da mesma será o seguinte:

Figura 6.7: Comportamento do mapa logístico para uma fórmula não satisfatível

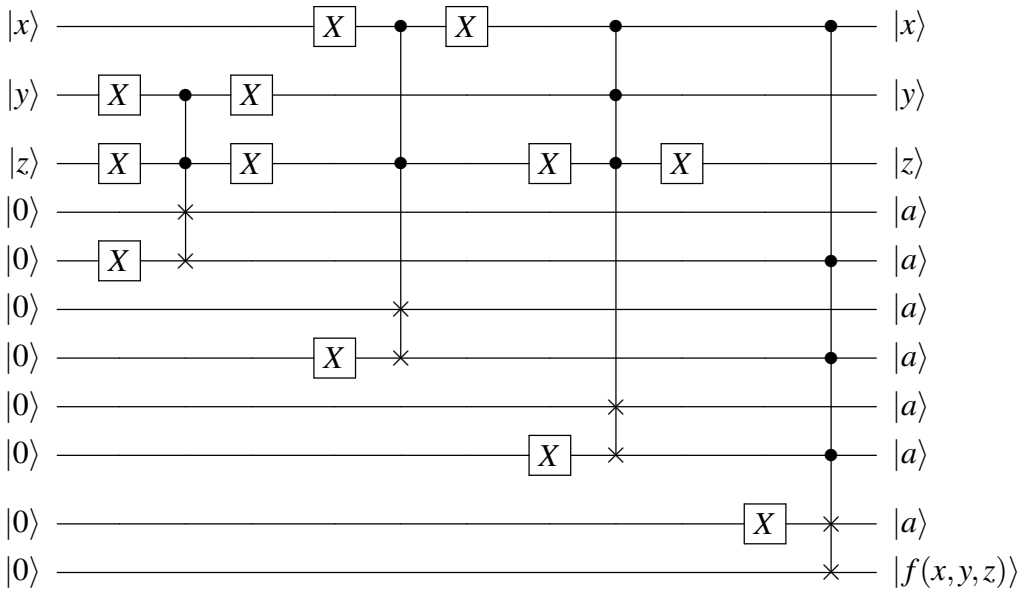


Fonte: O autor

Portanto, mesmo que haja uma pequena probabilidade de observar o qubit $|1\rangle$ na última posição do estado quântico resultante do circuito, ainda assim será possível obter o comportamento caótico através da aplicação do mapa logístico. Por outro lado, caso nenhuma atribuição de valores para as variáveis seja satisfatória, a saída será sempre 0.

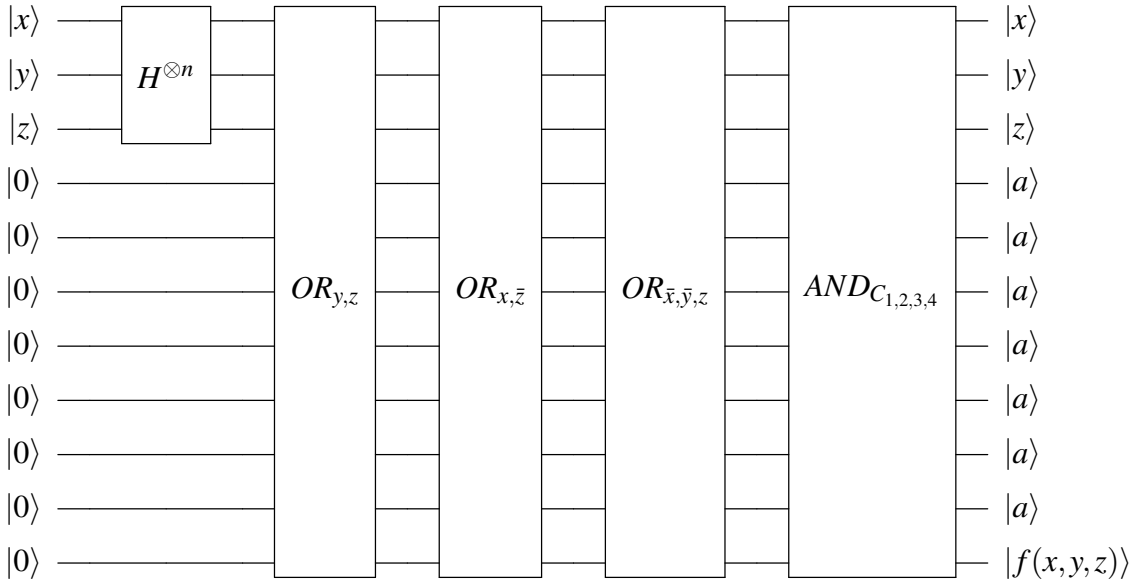
A Figura 6.8 representa o circuito que analisa a fórmula 1 na abordagem Fredkin; a Figura 6.9 representa sua forma simplificada.

Figura 6.8: Circuito com portas lógicas Fredkin que avalia a **Instância 1**



Fonte: O autor

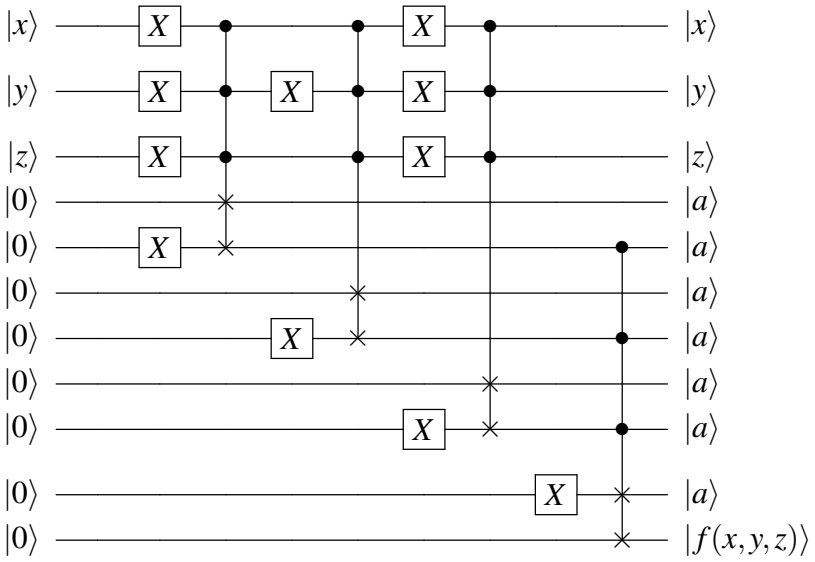
Figura 6.9: Circuito com portas lógicas Fredkin que avalia a **Instância 1** - versão simplificada



Fonte: O autor

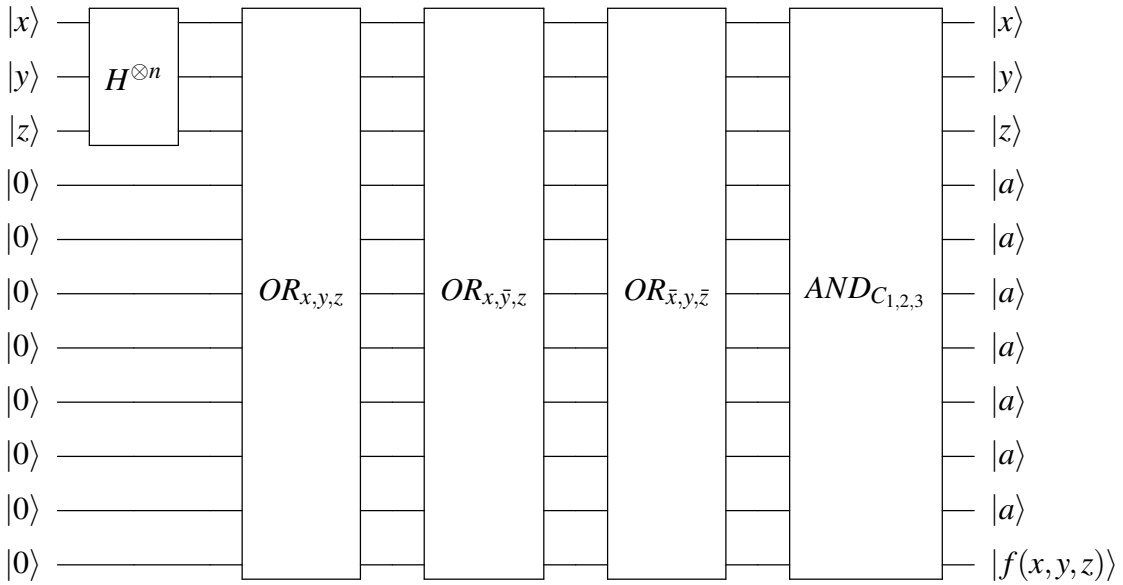
A Figura 6.10 representa o circuito que analisa a fórmula 2 na abordagem Fredkin; a Figura 6.11 representa sua forma simplificada.

Figura 6.10: Circuito com portas lógicas Fredkin que avalia a **Instância 2**



Fonte: O autor

Figura 6.11: Circuito com portas lógicas Fredkin que avalia a **Instância 2** - versão simplificada



Fonte: O autor

Com relação aos circuitos Fredkin, tem-se o seguinte estado resultante do circuito que

avalia a fórmula 1:

$$|\psi\rangle = \frac{1}{2\sqrt{2}}(|00010010110\rangle + |00101100110\rangle + |01001010110\rangle + |01101100110\rangle + \\ |10010010110\rangle + |10101010101\rangle + |11001011010\rangle + |11101010101\rangle)$$

A partir disto, é realizada a aplicação do operador $O^{(m)}$, o qual terá o seguinte resultado:

$$|\psi\rangle' = O^{(m)}|\psi\rangle \\ \alpha_0 2^n|1\rangle\langle 1|0\rangle + \alpha_1 2^n|1\rangle\langle 1|1\rangle \\ \alpha_1 2^n|1\rangle,$$

onde α_0 e α_1 são as probabilidade de se obter $|0\rangle$ e $|1\rangle$, respectivamente, na última posição do estado acima e n é o número de variáveis da fórmula. Assim, é possível distinguir este vetor de um vetor nulo.

Já para a fórmula 2, o estado quântico obtido na saída do circuito é:

$$|\psi\rangle = \frac{1}{2\sqrt{2}}(|00010010110\rangle + |00101010101\rangle + |01001100110\rangle + |01101010101\rangle + \\ |10001010101\rangle + |10101011010\rangle + |11001010101\rangle + |11101010101\rangle)$$

Aqui, novamente, tem-se a aplicação do operador $O^{(m)}$ como descrito acima. Para o caso de uma fórmula não satisfatível, o fator α_1 seria 0, dessa forma:

$$|\psi\rangle' = O^{(m)}|\psi\rangle \\ \alpha_0 2^n|1\rangle\langle 1|0\rangle + \alpha_1 2^n|1\rangle\langle 1|1\rangle \\ 2^n|1\rangle\langle 1|0\rangle$$

E, portanto, o vetor resultante seria nulo, como esperado.

7

Conclusão

Este documento atualiza e incrementa o estado da arte sobre computação quântica e problemas NP-completos apresentado por (LIMA; ISIDRO; LULA JÚNIOR, 2007), enfatizando as abordagens que utilizam o modelo de circuitos da computação quântica Porém, outras metodologias também foram mencionadas e listadas, como as soluções que fazem uso da não-linearidade e do modelo adiabático. Para tanto, foi feito uso das funcionalidades presentes no pacote quântico do SymPy, o *Quantum*, visando a implementação simbólica de tais abordagens.

A partir desta pesquisa foi possível observar que a investigação dos problemas NP-completos permite, ainda, novas e interessantes abordagens, sejam elas clássicas ou quânticas. E, mesmo que não tenha sido possível até o momento provar que $P = NP$ (ou que $P \neq NP$), a procura por soluções eficientes (ou, pelo menos, mais eficientes que as disponíveis) é, de fato, necessária, haja vista a importância matemático-computacional de resolver tais problemas de um modo eficaz.

Tanto o método proposto por (OHYA; VOLOVICH, 1999) quanto o proposto por (LEPORATI; FELLONI, 2007) foram considerados de relativamente fácil entendimento, viabilizando a codificação de ambos para simulação simbólica em computadores clássicos. A única ressalva a ser feita para estes trabalhos refere-se à ausência neles de um algoritmo capaz de construir de modo eficiente os circuitos compostos por portas lógicas quânticas responsáveis por analisar instâncias do problema. Tal entrave foi solucionado pelo presente autor através da aplicação dos pseudocódigos presentes na Seção 5.2.

Como mostrado na Seção 3.2.2, algumas poucas concepções de simuladores quânticos foram apresentadas. Tal fato indica dois pontos a serem analisados: primeiro, esse tipo de software é, de fato, necessário para o estudo e ensino da computação quântica, uma vez que objetiva mostrá-la sem as dificuldades inerentes apresentadas pela própria natureza da mecânica quântica; segundo, a quantidade reduzida de trabalhos que tratam dessa modelagem simbólica para a computação quântica mostra que, uma vez aceito que esse tipo de software, de fato, auxilia estudantes e pesquisadores, o surgimento e divulgação de ideias nesse âmbito (seja no SymPy ou em outro CAS) se caracteriza como uma importante contribuição para a comunidade acadêmico-científica.

Importante ressaltar que, apesar de se configurar como uma excelente ferramenta tanto para estudos como para cálculos científicos/simbólicos, o SymPy não fornece estruturas facilmente maleáveis que possibilitem a manipulação das mesmas, dificultando, assim, a construção de operadores não disponíveis no seu pacote quântico. Embora este fato tenha trazido algum empecilho durante a fase de desenvolvimento, o presente autor não julga ser este um fator de alta relevância quando da escolha de um CAS para computação quântica simbólica, sendo considerado que o SymPy/Quantum propicia o estudo das peculiaridades quânticas de maneira simplificada.

Como contribuições deste Trabalho de Conclusão de Curso, destaco o levantamento e revisão das propostas que se dispõem a investigar como a computação quântica pode agir para dispor soluções eficientes para os problemas NP-completos e a aplicação do SymPy na implementação simbólica de tais soluções.

7.1 Trabalhos Futuros

Esta seção conclui a escrita do presente trabalho com algumas possibilidades de continuidade do que foi proposto até aqui.

No tocante à ausência de alguns operadores e estruturas no *Quantum*, uma alternativa futura é explorar internamente a linguagem para, então, poder expandi-la de forma a fornecer uma representação simbólica que facilite a construção destes operadores sem que os mesmos interfiram no desempenho das simulações. Com relação ao que já foi desenvolvido, o presente autor pretende implementar uma interface gráfica para os simuladores propostos, objetivando facilitar a compreensão dos mesmos através dos diagramas dos circuitos gerados para analisar uma instância do SAT passada como parâmetro. Além disto, almeja-se incluir características básicas da computação quântica integradas à esta interface, possibilitando que o usuário tenha capacidade de explorar de modo visual as peculiaridades presentes na CQ.

Outro ponto que pode ser trabalhado diz respeito à otimização de circuitos quânticos. Uma vez que as duas principais abordagens exploradas neste trabalho propõem um modelo de solução para problemas NP-completos a partir do modelo de circuitos da CQ, estudar como otimizar os mesmos pode prover uma melhora no desempenho durante a execução das simulações, bem como simplificar a representação dos circuitos, tornando-a mais limpa. Em (WONG, 2012) e (AMY, 2013) são fornecidos algoritmos que possibilitam a otimização de circuitos quânticos. Particularmente, em (WONG, 2012), tem-se o uso do SymPy para a implementação de tais algoritmos, o que faz com que haja uma maior propensão em adotar estes como base para a otimização. Já em (AMY, 2013) tem-se a proposta de técnicas que automatizem o processo de otimização de circuitos quânticos, investigando como minimizar a profundidade dos mesmos e reduzir o espaço de busca neles.

Por fim, as implementações desenvolvidas neste trabalho podem ser aplicadas aos modelos de redes neurais quânticas propostas em (OLIVEIRA et al., 2008) e (OLIVEIRA, 2009).

Referências

AARONSON, S. NP-complete Problems and Physical Reality. **SIGACT News**, New York, NY, USA, v.36, n.1, p.30–52, Mar. 2005.

ABRAMS, D. S.; LLOYD, S. Nonlinear Quantum Mechanics Implies Polynomial-Time Solution for NP-complete and #P Problems. **Phys. Rev. Lett.**, [S.l.], v.81, p.3992–3995, Nov 1998.

AHARONOV, D. et al. Adiabatic Quantum Computation is Equivalent to Standard Quantum Computation. **SIAM J. Comput.**, Philadelphia, PA, USA, v.37, n.1, p.166–194, Apr. 2007.

AHARONOV, D.; NAVEH, T. Quantum NP - A Survey. , [S.l.], 2002. arXiv:quant-ph/0210077.

AMY, M. **Algorithms for the Optimization of Quantum Circuits**. 2013. Dissertação (Mestrado em Ciência da Computação) — University of Waterloo, Waterloo, Ontario, Canada.

ANDRECUT, M.; ALI, M. Adiabatic Quantum Gates. **International Journal of Theoretical Physics**, [S.l.], v.43, n.4, p.933–938, 2004.

ARAÚJO, A. de; FINGER, M. Classical and quantum satisfiability. **LSFA**, [S.l.], v.81, p.79–84, 2011. arXiv:1203.6161 [cs.CC].

ARI, N.; MAMATNAZAROVA, N. Symbolic python. **Electronics, Computer and Computation (ICECCO), 2014 11th International Conference on**, [S.l.], p.1–8, Sept 2014.

BARBOSA, A. d. A. **Um Simulador Simbólico de Circuitos Quânticos**. 2007. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal de Campina Grande, Campina Grande, Paraíba, Brasil.

BOLOTIN, A. **Computational solution to quantum foundational problems**. arXiv:1403.7686 [quant-ph], Phys. Sci. Int. J. 2014; 4(8): 1145-1157.

CLEVE, R. An introduction to quantum complexity theory. **arXiv preprint quant-ph/9906111**, [S.l.], v.28, 1999.

CUGINI, A. Quantum Mechanics, Quantum Computation, and the Density Operator in SymPy. , [S.l.], 2011. Disponível em <http://digitalcommons.calpoly.edu/physsp/38/>. Acesso em: 11 nov. 2015.

CURRY, M. Symbolic Quantum Circuit Simplification in SymPy. , [S.l.], 2011. Disponível em <http://digitalcommons.calpoly.edu/physsp/39/>. Acesso em: 11 nov. 2015.

DAM, W. van; MOSCA, M.; VAZIRANI, U. How powerful is adiabatic quantum computation? **Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on**, [S.l.], p.279–287, Oct 2001. arXiv:quant-ph/0206003.

DEUTSCH, D. Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer. **Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences**, [S.l.], v.400, n.1818, p.97–117, 1985.

DEUTSCH, D.; JOZSA, R. Rapid Solution of Problems by Quantum Computation. **Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences**, [S.l.], v.439, n.1907, p.553–558, 1992.

DICARLO, L. et al. Demonstration of two-qubit algorithms with a superconducting quantum processor. **Nature**, [S.l.], v.460, n.7252, p.240–244, July 2009.

FARHI, E. et al. **Quantum Computation by Adiabatic Evolution**. [S.l.: s.n.], 2000. arXiv:quant-ph/0001106.

FARHI, E. et al. A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem. **Science**, [S.l.], v.292, p.472–476, apr 2001.

FELDMANN, M. **Solving satisfiability by statistical estimation**. arXiv:1205.6658 [cs.CC].

FORTNOW, L. The Status of the P Versus NP Problem. **Commun. ACM**, New York, NY, USA, v.52, n.9, p.78–86, Sept. 2009.

FREEDMAN, M. H. P/NP, and the quantum field computer. **Proceedings of the National Academy of Sciences**, [S.l.], v.95, n.1, p.98–101, 1998.

FUX, J. **Análise de Algoritmos SAT para Resolução de Problemas Multivalorados**. 2004. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal de Minas Gerais, Belo Horizonte, Minas Gerais, Brasil.

GAITAN, F.; CLARK, L. Graph isomorphism and adiabatic quantum computing. **Phys. Rev. A**, [S.l.], v.89, p.022342, Feb 2014.

GAREY, M. R.; JOHNSON, D. S. **Computers and Intractability**: a guide to the theory of np-completeness. New York, NY, USA: W. H. Freeman & Co., 1979.

GIOVANNETTI, V.; LLOYD, S.; MACCONE, L. Quantum Random Access Memory. **Phys. Rev. Lett.**, [S.l.], v.100, p.160501, Apr 2008.

GROVER, L. K. A Fast Quantum Mechanical Algorithm for Database Search. In: TWENTY-EIGHTH ANNUAL ACM SYMPOSIUM ON THEORY OF COMPUTING, New York, NY, USA. **Proceedings...** ACM, 1996. p.212–219. (STOC '96).

JAEGER, G. **Quantum Information: an overview**. 1.ed. [S.l.]: Springer, 2007.

JOYNER, D. et al. Open Source Computer Algebra Systems: sympy. **ACM Commun. Comput. Algebra**, New York, NY, USA, v.45, n.3/4, p.225–234, Jan. 2012.

KENDALL, G.; PARKES, A.; SPOERER, K. A Survey of NP-Complete Puzzles. **International Computer Games Association Journal**, [S.l.], v.31, n.1, p.13–34, 2008.

LANTING, T. et al. Entanglement in a Quantum Annealing Processor. **Phys. Rev. X**, [S.l.], v.4, p.021041, May 2014.

LEPORATI, A.; FELLONI, S. Three "quantum" algorithms to solve 3-SAT. **Theoretical Computer Science**, [S.l.], v.372, n.2–3, p.218–241, 2007. Membrane Computing.

LIMA, A. F.; ISIDRO, C. R. G.; LULA JÚNIOR, B. Considerações sobre as possilibidade de solução quântica para problemas NP-completos. **II Workshop-Escola de Computação e Informação Quântica, WECIQ 2007**, Campina Grande, PB, Março 2007.

LUCAS, A. Ising formulations of many NP problems. **Frontiers in Physics**, [S.l.], v.2, n.5, 2014.

MCMAHON, D. **Quantum computing explained**. Hoboken, N.J. Wiley-Interscience: IEEE Computer Society, 2007.

MERMIN, N. D. From Cbits to Qbits: teaching computer scientists quantum mechanics. **American Journal of Physics**, [S.l.], v.71, p.23–30, 2003. arXiv:quant-ph/0207118.

MERMIN, N. D. **Quantum computer science** : an introduction. Cambridge: Cambridge University Press, 2007. Index inclus.

MÉZARD, M.; PARISI, G.; ZECCHINA, R. Analytic and Algorithmic Solution of Random Satisfiability Problems. **Science**, [S.l.], v.297, n.5582, p.812–815, Aug. 2002.

MONROE, C.; KIM, J. Scaling the Ion Trap Quantum Processor. **Science**, [S.l.], v.339, n.6124, p.1164–1169, 2013.

MOORE, G. Cramming More Components Onto Integrated Circuits. **Proceedings of the IEEE**, [S.l.], v.86, n.1, p.82–85, Jan 1998.

NIELSEN, M. A.; CHUANG, I. L. **Quantum Computation and Quantum Information**: 10th anniversary edition. 10th.ed. New York, NY, USA: Cambridge University Press, 2011.

OHYA, M. Quantum algorithm for {SAT} problem andquantum mutual entropy. **Reports on Mathematical Physics**, [S.l.], v.55, n.1, p.109 – 125, 2005.

OHYA, M. New quantum algorithm solving the NP complete problem. **P-Adic Numbers, Ultrametric Analysis, and Applications**, [S.l.], v.4, n.2, p.161–165, 2012.

OHYA, M.; MASUDA, N. **NP problem in quantum algorithm**. arXiv:quant-ph/9809075, OpenSyst.Info.Dyn.7:33-39,2000.

OHYA, M.; VOLOVICH, I. **Mathematical Foundations of Quantum Information and Computation and Its Applications to Nano- and Bio-systems**. [S.l.]: Springer Netherlands, 2011. (Theoretical and Mathematical Physics).

OHYA, M.; VOLOVICH, I. V. Quantum Computing, NP-complete Problems and Chaotic Dynamics. **CoRR**, [S.l.], v.quant-ph/9912100, 1999.

OHYA, M.; VOLOVICH, I. V. New quantum algorithm for studying NP-complete problems. **Reports on Mathematical Physics**, [S.l.], v.52, n.1, p.25 – 33, 2003.

OLIVEIRA, M. d. O. On the Satisfiability of Quantum Circuits of Small Treewidth. **Computer Science - Theory and Applications - 10th International Computer Science Symposium in Russia**, [S.l.], p.157–172, 2015.

OLIVEIRA, N. M. de; OLIVEIRA, W. R. de. Simulando Solução Polinomial Quântica para SAT. **V Workshop-Escola de Computação e Informação Quântica, WECIQ 2014**, Campina Grande, PB, Março 2015.

OLIVEIRA, W. R. de. Quantum RAM Based Neural Netoworks. **ESANN**, [S.l.], v.9, p.331–336, 2009.

OLIVEIRA, W. R. de et al. Quantum Logical Neural Networks. **Neural Networks, Brazilian Symposium on**, Los Alamitos, CA, USA, v.0, p.147–152, 2008.

PINSKI, S. **Adiabatic Quantum Computing**. 2011. Dissertação (Mestrado em Ciência da Computação) — Loughborough University, Loughborough, Leicestershire, East Midlands, England, United Kingdom. arXiv:1108.0560 [physics.pop-ph].

RADTKE, T.; FRITZSCHE, S. Simulation of n-qubit quantum systems. I. Quantum registers and quantum gates. **Computer Physics Communications**, [S.l.], v.173, n.1–2, p.91 – 113, 2005.

SARKAR, A.; BHATTACHARYYA, T. K.; PATWARDHAN, A. Quantum logic processor: implementation with electronic mach-zehnder interferometer. **Applied Physics Letters**, [S.l.], v.88, n.21, p.–, 2006.

SHOR, P. W. Algorithms for Quantum Computation: discrete logarithms and factoring. In: ANNUAL SYMPOSIUM ON FOUNDATIONS OF COMPUTER SCIENCE, 35., Washington, DC, USA. **Proceedings...** IEEE Computer Society, 1994. p.124–134. (SFCS '94).

SILVERMAN, M. P. **Quantum superposition counterintuitive consequences of coherence, entanglement, and interference**. [S.l.]: Springer, 2008.

SIMON, D. R. On the Power of Quantum Computation. **SIAM J. Comput.**, Philadelphia, PA, USA, v.26, n.5, p.1474–1483, Oct. 1997.

SONG, D. The P versus NP Problem in Quantum Physics. **NeuroQuantology**, [S.l.], v.12, n.4, 2014.

WANG, J. et al. A quantum method to test the satisfiability of Boolean functions. **Solid-State and Integrated Circuit Technology (ICSICT), 2012 IEEE 11th International Conference on**, [S.l.], p.1–5, Oct 2012.

WONG, R. G. An Algorithm For Quantum Circuit Optimization. , [S.l.], 2012. Disponível em <http://digitalcommons.calpoly.edu/cscsp/16/>. Acesso em: 11 nov. 2015.

YANOFSKY, N. S.; MANNUCCI, M. A. **Quantum Computing for Computer Scientists**. 1.ed. New York, NY, USA: Cambridge University Press, 2008.