IMPLEMENTAÇÃO DE ALGORITMOS QUÂNTICOS PARA A RESOLUÇÃO DO PROBLEMA NP-COMPLETO DA SATISFATIBILIDADE[1]

Caroline Nagy de Almeida – caroline.de_almeida@siemens.com
Antonio Luis Basile (Orientador) – albasile2@gmail.com

RESUMO

A Computação Quântica surgiu na década de 70 quando físicos questionaram se os principais fundamentos relacionadas a ciência da computação e teoria da informação poderiam ser resolvidas aplicando-se o estudo de sistemas quânticos. Com o intuito de aplicar a teoria da Mecânica Quântica para descrever um conceito abrangente em relação a Máquina de Turing Clássica, a Computação Quântica permite desenvolver, utilizando essas técnicas, algoritmos mais rápidos comparados aos algoritmos desenvolvidos nos computadores clássicos. A partir disto, a Computação Quântica foi amplamente desenvolvida para a solução de problemas de busca e de decisão, usando esse tipo de solução para fornecer uma resposta rápida á problemas NP-Completo. A Satisfatibilidade é o primeiro problema NP-Completo, demonstrado por Stephen Cook, a ser classificado como o mesmo, a partir dele, é possível desenvolver todos os problemas NP-Completos para a computação. Nesta dissertação, é estudada as classes de Satisfatibilidade e seus tempos de execução, utilizando recursos da álgebra linear e classes algorítmicas da computação clássica e da computação quântica, levando a resultados que condizem com o uso da computação quântica para o aprimoramento de tais processos.

Palavras-chave: mecânica quântica, computação quântica, problemas NP-Completo, satisfatibilidade booleana, SAT.

IMPLEMENTATION OF QUANTUM ALGORITHMS FOR THE RESOLUTION OF THE NP-COMPLETE PROBLEM OF SATISFIABILITY

ABSTRACT

Quantum Computation arose in the 1970s when physicists questioned if the main fundamentals related to computer science and information theory could be solved by applying the study of quantum systems. In order to apply the theory of quantum mechanics to describe a comprehensive concept in relation to the Classical Turing Machine, the Quantum Computation allows

[1] Artigo do Trabalho de Conclusão de Curso, Graduação em Engenharia Eletrônica, EE, UPM, São Paulo, 2019.

to develop, using these techniques, faster algorithms compared to algorithms developed in classical computers. From this, Quantum Computation has been extensively developed for the solution of search and decision problems, using this type of solution to provide a rapid response to NP-Complete problems. Satisfiability is the first NP-Complete problem demonstrated by Stephen Cook to be classified as the same, from this problem it is possible to develop all NP-Complete problems for computation. In this dissertation, the classes of Satisfiability and its execution times are studied, using resources from linear algebra and algorithmic classes of classical computation and quantum computation, leading to results that correspond to the use of quantum computation for the improvement of such processes.
Keywords: quantum mechanics, quantum computation, NP-Complete problems, boolean satisfiability, SAT.

1 INTRODUÇÃO

A Computação Quântica é utilizada como forma de obter um processamento algorítmico mais rápido em relação aos computadores clássicos usando as técnicas de álgebra propostas na Mecânica Quântica. A partir disto, a Computação Quântica foi amplamente desenvolvida para a solução de problemas de busca e de decisão, usando esse tipo de solução para fornecer uma resposta rápida á problemas NP-Completo. A Satisfatibilidade é um problema NP-Completo usado como base para outros problemas NP-Completos, consiste em determinar se existe um determinada valoração para as variáveis de uma determinada fórmula booleana tal que esta valoração satisfaça esta fórmula em questão, é um problema de decisão instanciado por expressão booleana escrita somente com operadores AND, OR e NOT, com isso o uso de um algoritmo satisfatível de *k* variáveis por cláusula booleana na expressão (chamado de *k*-SAT) possuem tempo de execução amplamente superior em Máquina de Turing comum comparados ao uso dos computadores quânticos. Utilizando soluções mais rápidas para a Satisfatibilidade é possível extrair dos mesmos soluções mais rápidas para outros problemas NP-Completos e utilizar esse recurso como aprimoramento do campo de inteligência artificial e testes de circuitos digitais que são formulados como instâncias SAT, motivando a pesquisa de algoritmos satisfatíveis eficientes.

2 METODOLOGIA

2.1 SATISFATIBILIDADE QUÂNTICA

A satisfatibilidade quântica consiste em algoritmos que tentam resolver o problema NP-completo da satisfatibilidade convencional, tentando reduzi-lo á tempo polinomial, sejam simbolicamente ou não, utilizando artifícios da computação quântica e da mecânica quântica. Nos computadores convencionais não é possível encontrar um algoritmo eficiente que resolva um problema SAT aleatório em tempo polinomial enquanto nos computadores quânticos, usando o

paralelismo quântico para resolver e avaliar todos os resultados presentes nas cláusulas do problema é dito mais eficiente e possível para resoluções de problemas NP-completos como o SAT. Nesta dissertação, será usado modelos de circuitos quânticos como forma de resolução e redução temporal do SAT, essa escolha é devido à similaridade dos circuitos quânticos aos circuitos convencionais já conhecidos e a lógica booleana do SAT. Essa comparação pode ser usada posteriormente para manipulações de circuitos eletrônicos convencionais.

2.2 SATISFATIBILIDADE QUÂNTICA USANDO PORTAS *FREDKIN*

2.2.1 Portas e Circuitos *Fredkin*

As portas lógicas de *Fredkin* são conhecidas por possuírem propriedades universais reversíveis, uma porta lógica de *Fredkin* tem três *qubits* de entrada e três *qubits* de saída, no qual pode ser exemplificado pôr *a, b, c* e *a', b', c'* respectivamente. O *qubit c* é um *qubit* de controle e o seu valor não é alterado com nenhuma ação proveniente da porta de *Fredkin*, logo *c' = c*. O *qubit c* se chama *qubit* de controle porque é ele que controla o que acontece com os outros dois *qubits a* e *b*. Se *c* for iniciado com 0 então *a* e *b* são deixados sozinhos, *a' = a, b' = b*. Se *c* for iniciado com 1 então *a* e *b* sofrem ações conforme acontecem nas portas *SWAP* (definida na seção 3 desta dissertação) em que *a' = b* e *b' = a*. As portas *Fredkin* são chamadas de reversíveis porque com as saídas *a', b', c'*, é possível determinar as entradas *a, b, c*, para que a propriedade de reversibilidade seja obedecida e os valores originais das entradas *a, b, c* sejam recuperadas, é preciso aplicar outra porta de *Fredkin* nas saídas *a', b', c'*. Colocando-se portas de *Fredkin* juntas, obtemos os circuitos *Fredkin*, a figura a seguir corresponde a um circuito de *Fredkin* com três portas e duas camadas:

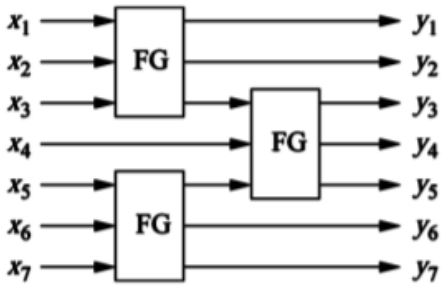

Figura 1: Circuito *Fredkin* composto por três portas *Fredkin*. FONTE: (LEPORATI; FELLONI, 2007)

Pode-se definir a função booleana do circuito como a composição das funções computadas por cada camada de portas *Fredkin*. Os recursos usados por um circuito de *Fredkin* para calcular uma função booleana, são considerados o tamanho e a profundidade do circuito, respectivamente definidos pelo número de portas *Fredkin* e o número de camadas. A porta *Fredkin* é funcionalmente completa para qualquer função booleana $f:\{0,1\}\to\{0,1\}$ onde exista um circuito *Fredkin* que calcula uma saída prefixada. Como mostrado na seção 3 desta dissertação, a porta *Fredkin* pode ser representada por uma matriz de ordem 8 ($8 = 2^3$ onde o número 3 é o número de entradas e saídas):

$$U_{FG} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (1)$$

Pode-se verificar que para todo $x_1, x_2, x_3 \in \{0,1\}$, $U_{FG}|x_1, x_2, x_3\rangle = |y_1, y_2, y_3\rangle$ onde $(y_1, y_2, y_3) = FG(x_1, x_2, x_3)$. É possível associar uma matriz unitária de ordem 2^n para qualquer *n*-entradas reversíveis de um circuito *Fredkin* FC_n (LEPORATI; FELLONI, 2007). Para que os estados de um circuito de *Fredkin* permaneça inalterado durante a performance de computação da camada, é necessário o uso de *qubits* a mais para realizar determinadas operações, formalmente nesse caso chamado de operador identidade. O operador identidade que atua sobre um único fio é:

$$ID_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (2)$$

Essa matriz unitária associada a *n*-entradas/*n*-saídas é facilmente identificada como uma matriz ID_n de ordem 2^n que também pode ser expressa como um produto tensor de *n*-dobras de ID_1:

$$ID_n = \otimes^n ID_1 \quad (3)$$

É possível então observar que a matriz unitária associada a uma camada é obtida através do cálculo do produto tensorial das matrizes que correspondem as portas de *Fredkin* e ao operador identidade. Como já definido nesta seção, FC_n é um circuito de *Fredkin* de *n* variáveis com portas lógicas denotadas por operadores unitários U_{FG}, usando a definição acima de operadores identidades temos que um operador identidade ID_m, sendo m o número de *qubits*, as camadas do circuito de *Fredkin* são construídas calculando o produto tensorial de ID_m com $U_{FG,}$ podemos usar como exemplo a figura 14 onde a matriz unitária está associada a primeira e a segunda camadas do circuito de *Fredkin*:

$$U_{FG} \otimes ID_1 \otimes U_{FG} \text{ e } ID_2 \otimes U_{FG} \otimes ID_2 \quad (4)$$

Seja *i* o número de camadas de FC_n, é possível determinar a matriz unitária U_{FC_n} correspondente ao circuito inteiro de *Fredkin* FC_n de instância ϕ com *n* variáveis, essa matriz unitária equivalente seria o produto das matrizes $U_{i_1}, U_{i_2}, ..., U_{i_j}$ associada a *i* camadas de FC_n:

$$U_{FC_n} = U_{i_j} \cdot ... \cdot U_{i_2} \cdot U_{i_1} \quad (5)$$

A fórmula acima pode ser representada na fórmula associativa algébrica $\otimes^n \mathbb{C}^2$. Essa fórmula consiste no produto das fórmulas que representam $U_{i_1}, U_{i_2}, ..., U_{i_j}$ associadas as camadas *i* de FC_n (LEPORATI; FELLONI, 2007). As definições de operador identidade e camadas para circuitos

Fredkin serão utilizadas a posteriori para a resolução da satisfatibilidade quântica utilizando portas *Fredkin.*

Além de serem reversíveis e conservativas as portas *Fredkin* podem ser configuradas para simular portas AND, NOT, CROSSOVER e FANOUT além de ser possível agrupá-las para simular qualquer circuito clássico. As configurações necessárias para esse tipo de simulação são mostradas na figura a seguir:

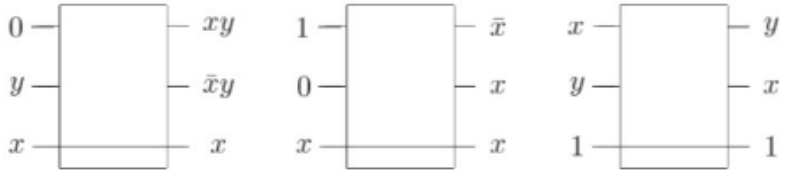

Figura 2: Circuito *Fredkin* configurado com uma performance de porta AND (esquerda), NOT (meio) e CROSSOVER (direita). A porta lógica do meio também serve como configuração de por FANOUT, desde que os dois *x* da saída sejam iguais. FONTE: (NIELSEN; CHUANG, 2000).

2.2.2 O Uso de Portas *Fredkin* no problema da Satisfatibilidade

Seja ϕ_n uma instância do 3-SAT como *n* variáveis, seja também um circuito *Fredkin* denotado FC_m com $m \geq n$ que computa ϕ_n nas primeiras saídas. Seja U_{FC_m} um operador unitário que corresponde a matriz de FC_m (LEPORATI; FELLONI, 2007). Seja:

$$H_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \tag{6}$$

uma matriz unitária que corresponde a uma porta *Hadamard* que possui um estado base $|0\rangle$ de um único *qubit*, temos:

$$H_1|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{7}$$

A porta *Hadamard* produz o estado de superposição para todas as *m* variáveis possíveis, usando a técnica de computação quântica de operadores de identidade definida nos circuitos *Fredkin*, temos m-resultados do produto tensorial de H_1, denominado $\otimes^m H_1$, para o uso de mais *qubits* segue a nova equação:

$$H_m = \otimes^m H_1 \frac{1}{\sqrt{2^m}} \otimes^m \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \tag{8}$$

cujo efeito em uma base de estado $|0...0\rangle$ da base computacional de $\otimes^n \mathbb{C}^2$ é:

$$H_m|0...0\rangle = \otimes^m H_1|0\rangle = \otimes^m \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2^m}} \sum_{x_1,...,x_m \in \{0,1\}} |x_1,...,x_m\rangle \tag{9}$$

Com isto, é criado uma superposição uniforme para todos os estados da base computacional $\otimes^n \mathbb{C}^2$. Conclui-se que ao aplicar um operador linear representado por U_{FC_m} em superposição é obtido como resultado uma combinação linear como todos os resultados "clássicos" possíveis na saída do circuito. Neste caso, a saída possui dois resultados possíveis:

a) $|0\rangle$, se ϕ_n não é satisfatível;

b) $a_0|0\rangle + a_1|1\rangle$, sendo $a_1 \neq 0$, se ϕ_n for satisfatível. O multiplicador $|a_1|$ será diretamente proporcional ao número de valores que satisfazem ϕ_n e portanto, pode ser exponencialmente pequeno em relação ao a_0, sendo que $|a_0|^2 + |a_1|^2 = 1$.

A probabilidade de observar um estado $|i\rangle$, sendo $i \in \{0,1\}$ é $|a_i|^2$. Isso significa que se ϕ_n é satisfatível, no pior caso seria preciso um número exponencial de cálculos e sucessivas medidas para obter um vetor de estado $|1\rangle$ na primeira saída de *FCm*. Com a finalidade de evitar o problema de magnitude exponencial, é necessário construir um operador linear seleção *O* representado pela matriz não-unitária:

$$2^n \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} = 2^n a^\dagger a = 2^n |1\rangle\langle 1| \tag{10}$$

O operador seleção pode ser construído através de:

$$O^{(m)} = O \otimes (\otimes^{m-1} ID_1), \tag{11}$$

no qual aplica-se o operador *O* no valor da primeira saída do circuito, e o operador identidade nas outras.

Usando o operador seleção mais a fórmula dor circuito de *Fredkin* com a porta *Hadamard* demonstrada acima, obtém-se o seguinte resultado:

$$O^{(m)} \cdot U_{FC_m} \cdot H_m |0...0\rangle \tag{12}$$

Observando o resultado do vetor, as duas saídas possíveis são:

a) O vetor nulo 0, se ϕ_n não é satisfatível, então:

$$O|0\rangle = 2^n |1\rangle\langle 1|0\rangle = 0 \tag{13}$$

b) O vetor não nulo, se ϕ_n é satisfatível:

$$O(a_0|0\rangle + a_1|1\rangle) = a_0 2^n |1\rangle\langle 1|0\rangle + a_1 2^n |1\rangle\langle 1|1\rangle = 0 + a_1 2^n |1\rangle = a_1 2^n |1\rangle \tag{14}$$

O coeficiente 2^n foi escolhido de forma que o vetor resultante não fosse de um valor pequeno, essa escolha ajudará a distinguir este resultado de um vetor nulo. Com os resultados enfim concluídos

é possível verificar a solução do problema NP-Completo 3-SAT com seu tempo reduzido para polinomial, é importante ressaltar que o circuito quântico FC_m depende da instância ϕ_n do 3-SAT para resolvê-lo. Para a resolução em tempo polinomial as seguintes condições devem ser obedecidas:

1) Ser possível construir e aplicar um operador $2^n |1\rangle\langle 1|$ para a saída da versão quântica do circuito de *Fredkin* FC_m;
2) A existência de um observador externo capaz de distinguir o resultado entre um vetor nulo e um vetor não-nulo (LEPORATI; FELLONI, 2007).

3 FUNDAMENTAÇÃO TEÓRICA

3.1 MECÂNICA QUÂNTICA

A Mecânica Quântica surgiu como forma de desenvolver matematicamente as teorias da física, comprovou-se a mesma através do experimento da dupla fenda de Young. O experimento tinha como objetivo medir o tamanho de uma onda de luz, o experimento consiste em emitir Fótons derivados de alguma fonte de luz como um laser, com isso é possível observar que essa fonte se comportará de maneira diferente dependendo da quantidade de fendas que forem utilizadas, quando apenas uma fenda for utilizada, a maioria das ondas de luz as atravessava mas a difração na borda das fendas produzia dois conjuntos próximos de ondas circulares que entravam em interferência, provando que a luz se comportava como onda porém, posteriormente, quando a descoberta do fóton foi realizado por Einstein, o experimento foi refeito inserindo apenas uma mínima quantidade de luz nas fendas, e o resultado foi que a luz estava se comportando com partícula, mostrando assim a tese de Einstein e provando a dualidade onda-partícula da luz.

A teoria da Mecânica Quântica faz o uso de combinações lineares de vetores de estados com coeficientes complexos tendo sua probabilidade calculada a partir dos quadrados dos módulos desses números complexos. O modelo matemático onde reside a Mecânica Quântica em termos formais é o Espaço de Hilbert (DE SOUZA, 2015). O Espaço de Hilbert consiste em representar os vetores da mecânica quântica em um espaço vetorial complexo, assim, cada vetor no espaço *H* representa um estado que poderia ser ocupado pelo sistema. Portanto, dados dois estados quaisquer, a soma algébrica deles também é um estado.

Esses vetores possuem estados normalizados e são representados por:

$$|\psi>$$

Figura 3: Representação do vetor de estado quântico, chamados de “Kets” de acordo com a notação de Dirac. FONTE: (NIELSEN; CHUANG, 2000).

Na matemática, são chamados funcionais todas as funções lineares que associam vetores de um espaço vetorial qualquer a um escalar, os funcionais dos vetores de um espaço também formam um espaço, que é chamado espaço "dual". (KUHN, 1978) Os funcionais são representados por:

$$<\psi|$$

Figura 4: Representação do vetor funcional, chamados de "Dual" de acordo com a notação de Dirac. FONTE: (NIELSEN; CHUANG, 2000).

3.2 TEORIA COMPUTACIONAL E COMPUTAÇÃO QUÂNTICA

A Computação Quântica surgiu para suprir a necessidade de resolver algoritmos exponencialmente mais rápido que os computadores já existentes. Baseando-se nas teorias da Mecânica Quântica, a Computação Quântica aplica as teorias da Superposição, Emaranhamento, Interferência e Paralelismo para moldar a álgebra de seu hardware. Aperfeiçoada pelos físicos teóricos Richard Feynman e David Deutsch na década de 1980, a computação quântica começou a ser tratada com importância e introduzida através do conceito da Máquina de Turing Quântica, apresentada por Benioff em 1980. Feynman então publicou um artigo em 1982 onde o mesmo dissertava que as Máquinas de Turing convencionais não seriam capazes de simular sistemas quânticos por ocasionar problemas de complexidade exponencial, sendo assim necessário a construção de computadores extremamente eficientes baseados nos princípios da Mecânica Quântica para a realização de tais simulações.

3.2.1 *qubits*

Em computadores clássicos, bits são considerados a única informação de processamento utilizada pelos mesmos, onde todos os dados podem ser resumidos a bits, sendo que um bit pode receber apenas um valor entre duas opções, o número zero e o número um. Nos computadores quânticos, a existência de bits são representados pelos equivalentes chamados bits quânticos, ou *qubit*, onde o mesmo possui dois estados representados por vetores denominados $|0\rangle$ e $|1\rangle$. Os análogos | > representam um vetor quântico e são chamados de notação de Dirac. A principal diferença entre um computador quântico e um computador quântico é o os *qubits* que são "configurados" de acordo com a lei da Mecânica Quântica chamada de superposição. A superposição indica que os estados quânticos $|0\rangle$ e $|1\rangle$ são assumidos ao mesmo tempo.

Os vetores $|0\rangle$ e $|1\rangle$ podem ser representados por matrizes:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ e } |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{15}$$

Se um computador quântico possuir *n qubits* em superposição ele poderá fazer uma operação de 2^n valores simultaneamente. Um *qubit* em superposição possui a notação $|\psi\rangle$ e o vetor é representado por:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{16}$$

Como citado no nesta seção, α e β são números complexos, são chamados de amplitude e satisfazem a seguinte condição:

$$|\alpha|^2 + |\beta|^2 = 1 \tag{17}$$

Temos que:

$$\begin{aligned} \alpha &= a + bi \qquad a, b \in \\ \beta &= c + di \qquad c, d \in R \end{aligned} \tag{18}$$

Onde:

A probabilidade de $|\psi\rangle = |0\rangle$ é igual a $|\alpha|^2 = a^2 + b^2$

A probabilidade de $|\psi\rangle = |1\rangle$ é igual a $|\alpha|^2 = c^2 + d^2$

Como visto nas equações acima, o *qubit* requer quatro números (a,b,c e d) para ser descrito, com os valores de sua amplitude, é possível reescrever a equação $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ como:

$$|\psi\rangle = e^{i\gamma}(\cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle), \tag{19}$$

onde $\gamma, \phi, \theta \in \mathbb{R}$. A fase global $e^{i\gamma}$ pode ser ignorada por não possuir efeito observável (NIELSEN; CHUANG, 2000). Assim, pode-se reescrever a equação como:

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle \tag{20}$$

Desta forma, um vetor de dois estados (*qubit*) pode ser representado por um ponto em uma esfera no $\mathbb{R}^3$, chamada de esfera de Bloch (Figura 3) (MARQUEZINO, 2006).

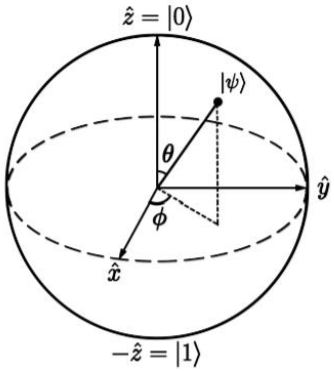

Figura 5: Representação geométrica do vetor de dois estados (*qubit*) através da esfera de Bloch.

FONTE: (NIELSEN; CHUANG, 2000).

Um conjunto de *n qubits* em superposição de 2^n estados possui valores de (00...), (10...), (01...), (11...), que correspondem a *qubits* no estado $|0\rangle$ e no estado $|1\rangle$, esses estados codificam todos os números possíveis de serem representados por *n bits*. Isso permite aplicar uma operação física que

corresponderia a um cálculo computacional, simultaneamente a todas as entradas possíveis, chamado de paralelismo (NICOLAU, 2010).

3.2.2 Paralelismo Quântico

O Paralelismo Quântico é a propriedade quântica que permite que um computador quântico encontre diversos valores para os seus sistemas simultaneamente, como exemplo podemos citar um sistema $f(x)$ sendo os seus resultados x_0, x_1, x_2, ..., x_f sendo encontrado simultaneamente (NIELSEN; CHUANG, 2000).

Para demonstrar o teorema do paralelismo, usa-se um vetor de estado quântico hipotético $|\psi_1\rangle$ formado por dois *qubits* $|a\rangle$ e $|b\rangle$,

$$|\psi_1\rangle = |a\rangle|b\rangle \tag{21}$$

onde esse vetor desenvolve a função de mapear um único bit *a* para *f(a)*, sendo a função *f(a)* demonstrada através de operador unitário conforme $f(a):\{0,1\}\rightarrow\{1,0\}$ e $a,b\in\{0,1\}$, tal que

$$|a,b\rangle \rightarrow |a,b\oplus f(a)\rangle, \tag{22}$$

obtendo a transformação para um operador unitário *U*, tal que

$$|U\rangle|a\rangle|b\rangle = |a\rangle|b\oplus f(a)\rangle, \tag{23}$$

$\oplus$ indica a soma módulo 2. Supondo que o valor inicial do *qubit* $|a\rangle$ na equação $|\psi_1\rangle = |a\rangle|b\rangle$ seja a superposição $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ e que $|b\rangle$ seja o estado base computacional $|0\rangle$, obtém-se o seguinte estado:

$$\begin{aligned} U|\psi_1\rangle &= U\frac{|0\rangle+|1\rangle}{\sqrt{2}}|0\rangle \\ U|\psi_1\rangle &= \frac{|0\rangle|f(0)\rangle+|1\rangle|f(1)\rangle}{\sqrt{2}} \end{aligned} \tag{24}$$

Portanto pode-se dizer que o vetor de estado quântico $|\psi_1\rangle$ permite avaliar uma função em pontos distintos simultaneamente ao contrário de um computador clássico no qual seriam necessárias múltiplo processadores ou execuções repetidas de um algoritmo (MARQUEZINO, 2006), (ALVES, 2003).

O paralelismo pode ser generalizado á funções com números arbitrários de *bits* utilizando a *transformada de Hadamard*. A transformada de *Hadamard* consiste em uma operação que utiliza a porta de *Hadamard*, denotada como:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \begin{matrix} \frac{1}{\sqrt{2}}(|0\rangle+|1\rangle) \\ \frac{1}{\sqrt{2}}(|0\rangle-|1\rangle) \end{matrix} \tag{24}$$

Essa operação demonstra *n* portas de *Hadamard* agindo em paralelo com *n qubits*. Um exemplo a ser demonstrado é usando *n* = 2 com *qubits* inicial igual a $|0\rangle$, então obtém-se como saída:

$$\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)=\frac{|00\rangle+|01\rangle+|10\rangle+|11\rangle}{2} \tag{25}$$

Duas portas de *Hadamard* agindo em paralelo são denominadas como $H^{\otimes 2}$, $\otimes$ é chamado produto tensorial, ou "tensor". Para transformadas de *Hadamard* com *n qubits* de estado inicial $|0\rangle$, o resultado da mesma é:

$$\frac{1}{\sqrt{2^n}}\sum_x |x\rangle \tag{26}$$

A somatória corresponde a todos os valores possíveis de *x* sendo denominado $H^{\otimes x}$. O *Hadamard* produz uma superposição uniforme em todos os estados de uma base computacional além de ser eficiente produzindo uma superposição de 2^n estados usando somente *n qubits*.

3.2.3 Circuitos Quânticos

Na computação clássica, a manipulação de resultados pode ser gerada através do uso de lógica matemática, onde os bits são manuseados usando a álgebra ao seu favor, essa manipulação matemática pode ser reconhecida através da lógica booleana. As operações mais simples feitas por computadores clássicos são aquelas que atuam em apenas um bit através da lógica booleana inversora, onde um bit é invertido para obter-se um resultado final como representados na tabela a seguir:

Entrada	**Porta NOT**	**Identidade**
0	1	0
1	0	1

Tabela 1: Representação da operação inversora sobre um bit

Na computação quântica, há infinitas portas lógicas não triviais que podem atuar em um único *qubit* (MARQUEZINO, 2006). Qualquer matriz 2x2 pode representar uma porta lógica quântica, diante disso a representação de *qubits* em matrizes é:

$$n\ qubits \equiv 2^n\ x\ 2^n\ (dimensão\ da\ matriz) \tag{27}$$

Logo, temos que a aplicação de 1 *qubit* sobre outro *qubit* é representada por:

$$\begin{aligned} qubit_p &= |\psi_p\rangle \\ qubit_q &= |\psi_q\rangle \\ &\therefore P \otimes Q \end{aligned} \tag{28}$$

Circuitos quânticos utilizam de forma computacional as leis da física quânticas mostradas nesta seção, além de ser parecido com circuitos clássicos em relação ao uso de lógico booleana na maioria

deles tornando a compreensão de computação quântica mais facilitada devido a essa semelhança, as portas lógicas dos circuitos quânticos são representadas por “caixas” e “fios”. Os “fios” representam o fluxo de dados que circulam entre uma porta e outra. São os circuitos quânticos que determinam quais e em quais ordem os operadores lógicos são aplicados a um ou mais *qubits* (DE OLIVEIRA, 2015). Os operados mais comuns e que serão utilizados a posteriori nos algoritmos da satisfatibilidade quântica desta dissertação são os operadores:

- Pauli-X

O operador Pauli-X define que o *qubit* resultante seja o inverso do *qubit* de entrada, se comportando com a porta lógica clássica denominada NOT.

É representado pela simbologia de portas quânticas como:

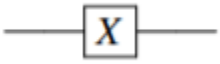

Figura 6: Porta lógica quântica “Pauli-X”. Fonte: (NIELSEN; CHUANG, 2000).

Tem como representação matricial e representação vetorial:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad \begin{aligned} X|0\rangle &= |1\rangle \\ X|1\rangle &= |0\rangle \end{aligned} \tag{29}$$

- *Hadamard*

Como definido do tópico 3.2.2 deste artigo, as portas lógicas *Hadamard* tem como objetivo criar uma superposição de estados

É representado pela simbologia de portas quânticas como:

Figura 7: Porta lógica quântica “*Hadamard*”. FONTE: (NIELSEN; CHUANG, 2000).

Tem como representação matricial e representação vetorial:

$$H = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \qquad \begin{aligned} H|0\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ H|1\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{aligned} \tag{30}$$

- *SWAP*

As portas lógicas *SWAP* também atuam sobre dois ou mais *qubits* sem precisar usar o produto tensorial, elas atuam em dois *qubits* trocando os seus valores (NIELSEN; CHUANG, 2000).

É representado pela simbologia de circuito quântico como:

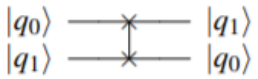

Figura 8: Circuito quântico "*SWAP*". FONTE: (NIELSEN; CHUANG, 2000).

Tem como representação matricial:

$$SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{31}$$

- *Fredkin*

A porta lógica de *Fredkin* possui 3 *qubits* e atua como uma porta controlada, dessa forma caso o *qubit* de controle ● tenha valor 1 os outros *qubits* × tem seus valores trocados entre si (DE OLIVEIRA, 2015). As portas lógicas *Fredkin* serão discutidas como base de uso para uma das resoluções do problema da satisfatibilidade quântica.

É representado pela simbologia de circuito quântico como:

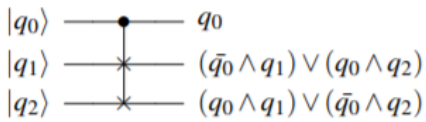

Figura 9: Circuito quântico "*Fredkin*". FONTE: (NIELSEN; CHUANG, 2000).

Tem como representação matricial:

$$FG = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{32}$$

3.3 COMPLEXIDADE COMPUTACIONAL E SATISFATIBILIDADE

3.3.1 Teoria da Complexidade Computacional

A Complexidade Computacional consiste em uma junção da ciência da computação e da matemática para o estudo da classificação de problemas computacionais de acordo com o seu nível dificuldade, classificando-os e relacionando-os; o estudo da complexidade computacional utiliza os recursos do espaço e tempo necessários para resolver esses problemas.

De acordo com Garey e Johnson (1979), um algoritmo é denominado polinomial quando sua complexidade é uma função polinomial no tamanho da entrada. Um algoritmo é eficiente quando sua complexidade é um polinômio no tamanho da entrada.

Algoritmos polinomiais podem ser descritos como uma função de complexidade *O(p(n))* onde *p(n)* é um polinômio, são considerados, de acordo com o seu tempo de execução, algoritmos de tempo de pior caso.

Os problemas, segundo Garey e Johnson (1979) são classificados, de acordo com a sua complexidade, da seguinte maneira:

- Tratável: problemas computacionais que possuem soluções algorítmicas em tempo polinomial;
- Intratável: problemas computacionais decidível que não possuem soluções algorítmicas em tempo polinomial, mas podem ser resolvidos em tempo superpolinomial;
- Indecidível: problemas difíceis, de decisão, cuja solução é impossível de obter através da construção de um algoritmo que retorne uma resposta decidível, sendo sim ou não.

A partir dessas classificações podemos classificar os problemas em duas classes de complexidade chamadas de tratáveis e intratáveis. Um problema é considerado tratável se, e somente se, existir um algoritmo para resolvê-lo usando recursos polinomiais, esse tipo de problema é denominado P.

A classe P consiste em problemas que podem ser resolvidos através da fórmula:

$$O(n^k) \qquad k = constate; \tag{33}$$

$$n = tamanho\ de\ entrada\ do\ problema; \tag{34}$$

Um problema é considerado intratável se, e somente se, o melhor algoritmo existente para resolvê-lo requer recursos exponenciais, esse tipo de problema é denominado NP. Problemas NP podem ser verificáveis em tempo polinomial, ou seja, dada uma determinada solução (certificado) é possível verificá-la em tempo polinomial. (FUX, 2004).

As definições acima podem ser usadas para mostrar que P $\subseteq$ NP, porém não é possível afirmar se P é ou não um subconjunto próprio de NP.

O subconjunto derivado da classe NP, conhecido como NP-Completo, possui a subsequente classificação: Um conjunto é NP-Completo se, e somente se:

- É um problema NP;
- Qualquer outro problema NP é redutível em tempo polinomial equivalente a ele.

Conclui-se que a classe P tratável, por possuir problemas “fáceis”, de tempo de execução polinomial simples pode ter facilmente resolvidos os seus problemas em um computador clássico já

o problema NP-Completo pode ter soluções verificadas em um computador clássico porém, alguns problemas como o problema do Caixeiro Viajante, pode se tornar intratável em um computador clássico se submetido a um número muito grande de entradas a serem fatoradas. (DE SOUZA, 2015)

3.3.3 Satisfatibilidade

A satisfatibilidade foi o primeiro problema provado ser da classe NP-Completo, o problema consiste em determinar se uma fórmula booleana é satisfeita ou não de forma que se existe uma determinada valoração para as variáveis de uma determinada fórmula booleana tal que esta valoração satisfaça a fórmula em questão.

Uma expressão booleana contém as seguintes condições para ser construída e considerada uma fórmula booleana: variáveis que assumam valores de decisão, no caso 0 para falso e 1 para verdadeiro, um operador unitário para negação (NOT), operadores binários AND e OR e parênteses.

Fórmulas booleanas retornam apenas um valor, seja ele FALSO ou VERDADEIRO (0 e 1). A Satisfatibilidade pode ser definida então como uma instância que pode ser apresentada através de uma expressão booleana denominada fórmula normal conjuntiva (CNF) que possui três variáveis por cláusula. A expressão é:

$$(x_{11} \vee x_{12} \vee x_{13}) \wedge (x_{21} \vee x_{22} \vee x_{23}) \wedge (x_{31} \vee x_{32} \vee x_{33}) \ \ldots \tag{35}$$

Onde x é uma variável ou uma negação de uma variável literal e cada variável pode aparecer mais de uma vez em cada fórmula. Uma desta cláusulas é dita satisfeita se pelo menos um dos literais assume o valor 1, não satisfeita se todos os seus literais assumem o valor 0, unitária se todos os literais, exceto um, assumem o valor 0, e não resolvível caso contrário (chamados literais livres, que não assumem nenhum valor).

3.3.4 Definição Algébrica da Satisfatibilidade

Sejam $X = \{x_1, x_2, ..., x_n\}$ um conjunto de variáveis booleanas, *a* uma atribuição de valores a *X* e *t* uma função $t : X \to \{0,1\}$, se *t(x) = 0*, diz-se que *x* é falso sobre *t*. Se x_i é uma varável em *X*, então x_i e $\overline{x_i}$ são literais sobre *X*. O literal x_i é verdadeiro se, somente se, a variável x_i é verdadeira; o literal $\overline{x_i}$ é verdadeiro se, e somente se, a variável x_i é falsa (PARREIRA, 1995). O número de variáveis possíveis para *X* é 2^n. De acordo com De Oliveira (2015), pode-se dizer que *f(X')* é usado para representar o conjunto de todos os subconjunto de *X'*, sendo $X' = \{x_1, \overline{x_2}, ..., x_n\}$ e $C \in f(X')$ constitui o que é chamado de cláusula. Uma vez que este problema figura como uma fórmula booleana na forma normal conjuntiva, para que $C = \{C_1, C_2, ..., C_n\}$ seja satisfatível, faz-se necessário que todos os valores de *t(C)* sejam verdadeiros para todo $C_i (i = 1, 2, 3, ..., m)$, ou seja, $t(C) \equiv \wedge_{i=1}^{m} t(C_i) = 1$ onde

$t(C) \equiv \vee_{x \in C^{t(x)}}$ e $\wedge$ e $\vee$ são os operadores lógicos *AND* e *OR*. Resumindo, o problema da SAT se resumo a perguntar se é possível atribuir um conjunto de valores verdade a *C* que o faço satisfatível.

Ainda de acordo com De Oliveira (2015), a definição formal de SAT é:

Dado um conjunto $X' = \{x_1, \overline{x_2}, ..., x_n\}$ de literais e suas negações, e um conjunto $C = \{C_1, C_2, ..., C_n\}$ de cláusulas, determinar se *C* é satisfatível ou não.

O problema conhecido com 3-SAT também pode ser definido formalmente como:

Dado um conjunto $X' = \{x_1, \overline{x_2}, ..., x_n\}$ de literais e suas negações, e um conjunto $C = \{C_1, C_2, ..., C_n\}$ de cláusulas que contém 3 elementos de *X'*, determinar se *C* é satisfatível ou não.

4 RESULTADOS

4.1 SIMULAÇÕES USANDO PORTAS *FREDKIN*

Dado o circuito *Fredkin* que é representado pelo circuito quântico:

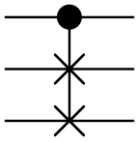

Figura 10: Circuito *Fredkin*. FONTE: (NIELSEN; CHUANG, 2000).

Onde o símbolo $\bullet$ representa o *qubit* de controle, em que o seu valor na entrada e na saída permanecem o mesmo, e o símbolo $\times$ representa os dois *qubits* de saída.

Para que este circuito se comporte como lógica booleana clássica, permitindo-se usá-lo como portas AND e OR e assim aplicá-lo ao problema 3-SAT são usados portas Pauli-X, ou mais conhecidas como portas NOT para circuitos clássicos. As portas Pauli-X podem ser ilustradas por:

Figura 11: Representações de portas quânticas Pauli-X. FONTES: (GIDNEY, 2004), (NIELSEN; CHUANG, 2000).

Para que o circuito de *Fredkin* se comporte como uma porta lógica AND é necessário colocar uma porta Pauli-X antes do primeiro *qubit* de saída, fazendo-o comportar-se como uma porta NAND, e o segundo *qubit* de saída se comporta como uma porta AND.

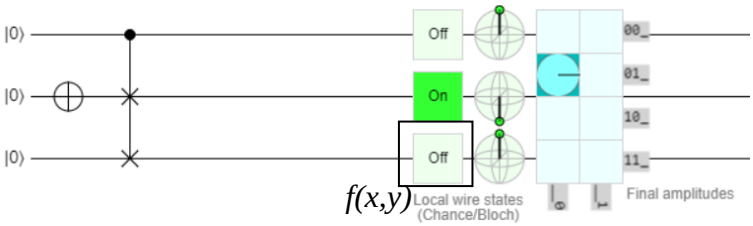

Figura 12: Representação da porta AND a partir do circuito de *Fredkin*. FONTE: O Autor (2019), (GIDNEY, 2004)

Para que o circuito de *Fredkin* se comporte como uma porta lógica OR é necessário colocar uma porta Pauli-X antes do primeiro *qubit* de saída, fazendo-o comportar-se como uma porta NOR, e o segundo *qubit* de saída se comporta como uma porta OR.

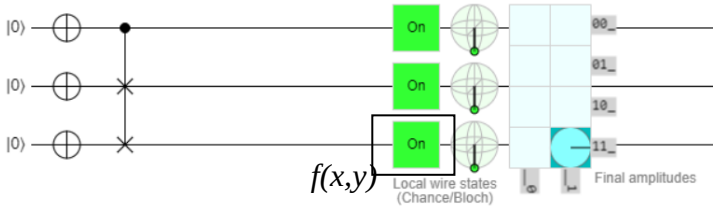

Figura 13: Representação da porta OR a partir do circuito de *Fredkin*. FONTE: O Autor (2019), (GIDNEY, 2004)

4.1.1 Simulando o 3-SAT através de portas *Fredkin*

Para simular o problema da satisfatibilidade quântica usando portas *Fredkin* usou-se as portas AND e OR equivalentes para a construção do circuito, sendo o primeiro exemplo a equação de um problema 3-SAT arbitrário escolhido pela autora:

$$(x \vee y \vee z) \wedge (\bar{x} \vee \bar{y} \vee z) \wedge (\bar{x} \vee y \vee \bar{z}) \tag{36}$$

A partir disso, o circuito foi montado e as cláusulas positivas são representadas com o resultado “On” e as cláusulas negadas são representadas por “Off” na ilustração do circuito a seguir:

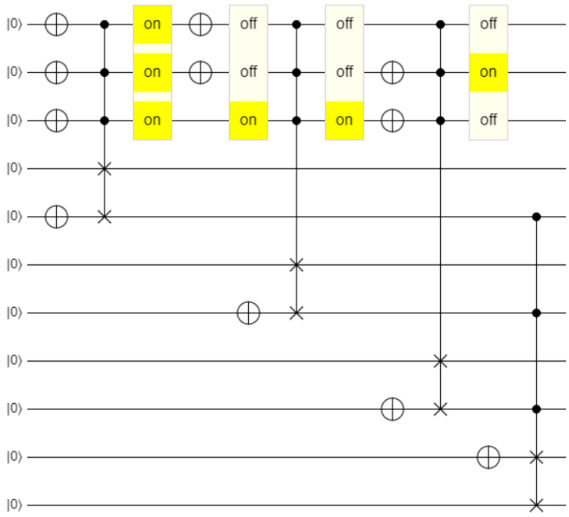

Figura 14: Representação das saídas de um 3-SAT a partir de simulação no *Quirk*. FONTE: O Autor (2019), (GIDNEY, 2004)

Após a montagem do circuito lógico usando *Fredkin*, colocou-se então as portas *Hadamard* de acordo com a demonstração algébrica da satisfatibilidade quântica usando *Fredkin*, assim a propriedade de paralelismo quântico pode ser observada.

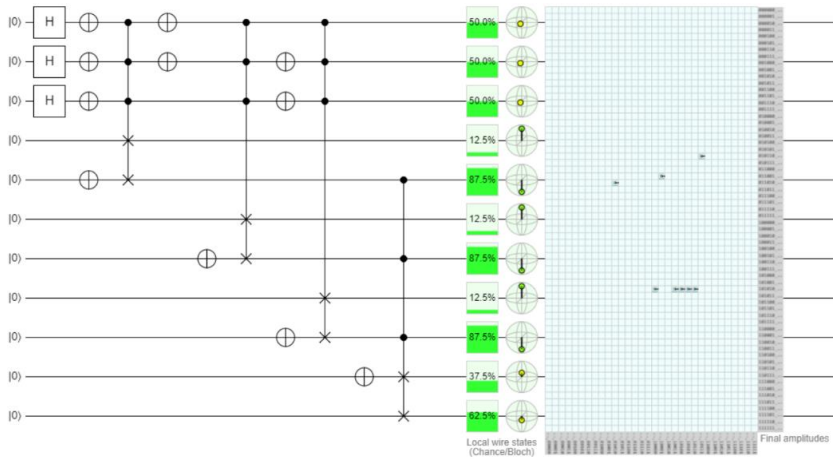

Figura 15: Representação de um 3-SAT a partir de um circuito *Fredkin* por simulação no *Quirk*.

FONTE: O Autor (2019), (GIDNEY, 2004)

Com isso encontrou-se as amplitudes dos *qubits* finais representados por oitos resultados equivalentes a:

$$U_{FC_m} = |01011010111\rangle|01100110001\rangle|01101001010\rangle|10101010000\rangle \\ |10101010011\rangle|10101010100\rangle|10101010101\rangle|10101010110\rangle \quad (37)$$

Os resultados podem ser encontrados no mapa de amplitudes apresentado na figura 22, onde os ponto no mapa representam os resultados finais do circuito quântico.

Usando a demonstração algébrica

$$H_m|0...0\rangle = \otimes^m H_1|0\rangle = \otimes^m \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2^m}} \sum_{x_1,...,x_m \in \{0,1\}} |x_1,...,x_m\rangle, \quad (38)$$

com $m = 3$ temos que:

$$U_{FC_m} \cdot H_m|0...0\rangle = \frac{1}{2\sqrt{2}}|01011010111\rangle|01100110001\rangle|01101001010\rangle|10101010000\rangle \\ |10101010011\rangle|10101010100\rangle|10101010101\rangle|10101010110\rangle \quad (39)$$

Para satisfazer demonstração algébrica da satisfatibilidade é necessário o uso do operador de seleção para que o circuito não seja executado de forma exponencial para obter um vetor de estado $|1\rangle$ na primeira saída de FC_m (conforme demonstração na seção 4):

$$|\psi\rangle = O^{(m)} \cdot U_{FC_m} \cdot H_m|0...0\rangle \quad (40)$$

Para a primeira possibilidade de resultado, quando o vetor for nulo e a fórmula não satisfatível, é usada a fórmula do operador de seleção:

$$O|0\rangle = 2^n|1\rangle\langle 1|0\rangle = 0 \quad (41)$$

Substituindo o operador O na fórmula inicial temos:

$$|\psi\rangle = O^{(3)} \frac{1}{2\sqrt{2}} |01011010111\rangle|01100110001\rangle|01101001010\rangle|10101010000\rangle$$
$$|10101010011\rangle|10101010100\rangle|10101010101\rangle|10101010110\rangle \quad (42)$$

$$|\psi\rangle = 0 \cdot \frac{1}{2\sqrt{2}} |01011010111\rangle|01100110001\rangle|01101001010\rangle|10101010000\rangle$$
$$|10101010011\rangle|10101010100\rangle|10101010101\rangle|10101010110\rangle$$

Logo, o resultado é zero, sendo não satisfatível e obtendo o vetor nulo como era esperado.

Para a segunda possibilidade de resultado, quando o vetor não for nulo e a fórmula satisfatível, é usada a fórmula do operador de seleção:

$$O(a_0|0\rangle + a_1|1\rangle) = a_0 2^n |1\rangle\langle 1|0\rangle + a_1 2^n |1\rangle\langle 1|1\rangle = 0 + a_1 2^n |1\rangle = a_1 2^n |1\rangle \quad (43)$$

Como demonstrado na seção 4 desta dissertação, os multiplicadores α_0 e α_1 são a probabilidade de o vetor indicar o valor $|0\rangle$ ou $|1\rangle$ na última saída do circuito (LEPORATI; FELLONI, 2007), logo:

$$|\psi\rangle = a_1 2^3 |1\rangle \cdot \frac{1}{2\sqrt{2}} |01011010111\rangle|01100110001\rangle|01101001010\rangle|10101010000\rangle$$
$$|10101010011\rangle|10101010100\rangle|10101010101\rangle|10101010110\rangle \quad (44)$$

$$|\psi\rangle = a_1 2\sqrt{2} |1\rangle|01011010111\rangle|01100110001\rangle|01101001010\rangle|10101010000\rangle$$
$$|10101010011\rangle|10101010100\rangle|10101010101\rangle|10101010110\rangle$$

Podendo então verificar que $|\psi\rangle$ não será um vetor nulo em nenhum momento, tornando a fórmula então satisfatível.

Além do uso algébrico proposto por Leporati e Felloni (2007), é possível calcular qual a probabilidade da saída de *f(x,y,z)* ser satisfatível, isto é, ser um vetor não nulo. Essa probabilidade é calculada no *Quirk*, o mesmo mostra na última saída do circuito (que corresponde a *f(x,y,z)*), a porcentagem que corresponde a probabilidade de receber uma saída igual a 1.

Esse cálculo é feito através dos coeficientes de probabilidade do qubit apresentados na seção 3 desta dissertação. No Quirk, esses coeficientes são calculados usando a geometria da esfera Bloch (também apresentada na seção 3) e através da mesma, encontra-se a porcentagem da saída ser satisfatível como é mostrado a seguir na saída do mesmo circuito apresentado na figura 15:

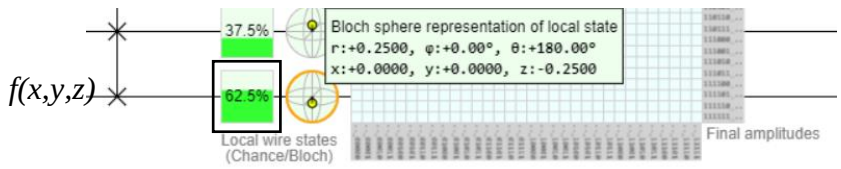

Figura 16: Representação do resultado de *f(x,y,z)* de um 3-SAT a partir de um circuito *Fredkin* por simulação no *Quirk*. FONTE: O Autor (2019), (GIDNEY, 2004)

Segundo a simulação realizada, a probabilidade de se obter um vetor não nulo na saída do circuito é 62,5%.

5 DISCUSSÃO

Através do uso da ferramenta de simulações de circuitos quânticos *Quirk* foi possível montar um circuito quântico equivalente a um circuito clássico para um problema arbitrário NP-Completo 3-SAT. Usando portas quânticas *Fredkin* foi possível transformá-las em portas AND e OR manipulando-a através do uso de portas quânticas Pauli-X, esta que tem a função equivalente a portas clássicas NOT, sendo esta propriedade de transformação das portas *Fredkin* extremamente útil para comparações entre computadores clássicos e computadores quânticos. Usando este problema arbitrário 3-SAT é possível observar que, neste caso, para cada problema é necessário que seja feito um novo circuito no simulador *Quirk*, sendo assim uma possibilidade de melhoria para trabalhos futuros.

O problema arbitrário 3-SAT escolhido para simulação mostra que, após a sua montagem, ele ainda se comporta como um circuito clássico e que para a propriedade de computador quântico seja inserida no mesmo são usadas portas *Hadamard* como objetivo de inserir o paralelismo quântico no circuito e assim obter diversos resultados calculados ao mesmo tempo. Com o uso dessa porta, encontrou-se diversas "amplitudes" no simulador *Quirk*, essas amplitudes são os pontos equivalentes ao produto tensorial entre as entradas *x, y* e *z* do problema arbitrário 3-SAT.

As amplitudes encontradas são essenciais para colocar em prática a prova algébrica proposta por Leporati e Felloni (2007), elas são substituídas na cláusula da transformada de *Hadamard*, mesmo com essa substituição os valores das amplitudes não são influenciadores para decidir se a fórmula é satisfatível ou não, mas sim o operador seleção.

O operador seleção age sobre a saída *f(x,y,z)* do circuito 3-SAT feito com portas *Fredkin*, o operador é um multiplicador algébrico, este não está presente no circuito, somente no cálculo para determinar se a fórmula é satisfatível ou não. Se a saída *f(x,y,z)* corresponde a um vetor nulo, o operador seleção também é nulo, multiplicando-o na fórmula proposta por Leporati e Felloni (2007) a equação resulta em um valor nulo, sendo assim não satisfatível. Se a saída não for um vetor nulo, o operador seleção recebe um coeficiente a_1, esse coeficiente é a probabilidade da última saída *f(x,y,z)* da fórmula ser um vetor positivo. É possível observar que, neste caso, o vetor não é nulo, não existe nenhum multiplicador nulo na equação demonstrada na seção 5.1.1 que faça com que essa fórmula seja nula, logo o vetor resultante é positivo e a fórmula é satisfatível.

É importante deixar claro que para que essas condições sejam verdadeiras, o observador precisa distinguir um vetor nulo de um vetor não nulo e que seja possível a inserção do operador seleção na última saída do circuito em que está o resultado *f(x,y,z)*, é importante também explicar que

nessa simulação, o operador identidade não foi usado pois o mesmo não influencia no valor do resultado do circuito (LEPORATI; FELLONI, 2007).

6 CONSIDERAÇÕES FINAIS

Neste trabalho foi apresentado o conceito de computação quântica para resolução de problemas NP-Completo como o problema da Satisfatibilidade. Além disto, foi decorrido todo o fundamento teórico necessário para tal processo.

Após a simulação de um problema arbitrário 3-SAT através do *Quirk* e de sua substituição na prova algébrica da satisfatibilidade quântica através da circuitos *Fredkin* conclui-se que apesar de atualmente ser possível simular circuitos quânticos em computadores clássicos como o simulador *Quirk*, ainda há partes do problema exposto, como o operador seleção, que não são possíveis simular no *Quirk* para este tipo de problema, sendo necessário finaliza-lo através de prova matemática para assim conseguir mostrar que é possível a redução a tempo polinomial. O difícil acesso a um computador quântico de fato também foi um dos problemas encontrados, apesar de toda a álgebra deste computador já desenhada e provada matematicamente, os computadores quânticos existentes ainda são instáveis e suscetíveis a erros, ainda há muito trabalho a ser desenvolvido para este ramo.

A redução de problemas NP-Completo de exponencial para polinomial é um avanço extremamente importante para a inteligência artificial e redes neurais, sendo possível inteligências com uma memória extremamente maior, processamento mais rápido e infinitos resultados. O uso de circuitos quânticos para tais aprimoramentos simulando circuitos clássicos é extremamente relevante também para a eletrônica clássica, onde a mesma sofreria reduções em tamanhos de circuito, aumento de processamento de dados e saídas infinitas.

Conclui-se que a satisfatibilidade quântica é um dos problemas NP-Completo a ser pioneiro na redução de um problema a fim de chegar no resultado NP = P, com ela será possível reduzir qualquer problema NP-Completo a polinomial, o estudo dela é fundamental para esta redução e, com o aprimoramento dos computadores quânticos, esses algoritmos discutidos teoricamente poderão ser implementados e ter seu tempo real testado.

REFERÊNCIAS

ALVES, Flávio Luís. **Computação Quântica: Fundamentos Físicos e Perspectivas.** 2003. 64 f. Monografia (Graduação) - Curso de Ciência da Computação, Universidade Federal de Lavras, Lavras, MG, 2003. Disponível em: <http://repositorio.ufla.br/bitstream/1/9369/1/MONOGRAFIA_Computa%C3%A7%C3%A3o_qu%C3%A2ntica_fundamentos_f%C3%ADsicos_e_%20perspectivas.pdf>. Acesso em: 02 mar. 2019.

BAKER, Joanne. **50 Ideias de Física Quântica que Você Precisa Conhecer.** São Paulo: Editora Planeta do Brasil, 2015. 214 p.

BUENO, Letícia. **Teoria da Complexidade Computacional.** São Paulo: Universidade Federal do Abc, 2011. 7 slides, P&B. Disponível em: <http://professor.ufabc.edu.br/~leticia.bueno/classes/aa/materiais/complexidade2.pdf>. Acesso em: 25 out. 2017.

DE SOUZA, Flavio Jesus. **Emprego da Computação Quântica em Problemas de Difícil Decisão.** 2015. 79 f. Tese (Mestrado em Computação Aplicada) – Universidade Estadual do Ceará e do Instituto Federal de Educação, Ciência e Tecnologia do Ceará, Ceará.

DE OLIVEIRA, Nicolas Melo. **Abordagens Quânticas Para P *Versus* NP e Simulações Simbólicas.** 2015. 79 f. Monografia (Graduação) - Curso de Ciência da Computação, Universidade Federal Rural de Pernambuco, Recife, PE, 2015. Disponível em: <http://www.bcc.ufrpe.br/sites/ww3.bcc.ufrpe.br/files/TCC%20-%20Nicolas%20Melo.pdf>. Acesso em: 13 fev. 2019.

GAREY, M. R. e JOHNSON, D. S. **Computers and Intractability. A guide to the theory of NP-Completeness**, W.H. Freeman and Company: New York, 1979.

GRATTAGE, Jonathan. **QML: A Functional Quantum Programming Language**, [200-?]. Disponível em: <http://sneezy.cs.nott.ac.uk/QML/>
GIDNEY, Craig. **The Quirky Quantum Simulator**. Software aberto online. Versão 2.2 [S. l.]. jan. 2004. Disponível em: < https://algassert.com/quirk>. Acesso em: 10 abr. 2019

FUX, Jacques. **Análise de Algoritmos SAT para Resolução de Problemas Multivalorados.** 2004. 45 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, 2004.

KUHN, T.S.. **Black-body theory and the quantum discontinuity**. 1. ed. Oxford: Clarendon Press,1978.

MARQUEZINO, Franklin de Lima. **Computação Quântica e Inteligência de Enxames Aplicados na Identificação de Acidentes de uma Usina Nuclear PWR.** 2006. 135 f. Dissertação (Mestrado) - Curso de Modelagem Computacional, Laboratório Nacional de Computação Científica, Petrópolis, RJ, 2006.

NICOLAU, Andressa dos Santos. **A Transformada de Fourier Quântica Aproximada e sua Simulação.** 2010. 108 f. Dissertação (Mestrado) – Pós-Graduação em Engenharia Nuclear, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, 2010.

NIELSEN, Michael A.; CHUANG, Isaac L.. **Quantum Computation an Quantum Information.** 10. ed. New York: Cambridge University Press, 2010. 698 p.

PARREIRA, Anderson Delcio. **Métodos Algébricos-Enumerativos para o Problema de Máxima Satisfatibilidade Ponderada.** 1995. 123 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade de Campinas (UNICAMP), Campinas, SP, 1995. Disponível em: <http://repositorio.unicamp.br/jspui/bitstream/REPOSIP/276104/1/Parreira_AndersonDelcio_M.pdf >. Acesso em: 20 fev. 2019.