Uma Proposta de Solução do Problema 3-SAT em Computação Quântica Usando Qiskit

Ricardo G. M. S. Ruiz[1], **Gabriel A. R. Gomes**[1], **Calebe P. Bianchini**[1]

[1]Faculdade de Computação e Informática
Universidade Presbiteriana Mackenzie
São Paulo – SP – Brasil

{10389321, 10389313}@mackenzista.com.br, calebe.bianchini@mackenzie.br

Abstract. *The expansion of theoretical and practical studies of quantum solutions has become notable in recent years. Based on this growth, this meta-article aims to not only investigate the state-of-the-art strategies of quantum circuits for solving the classical and costly satisfiability problem but also to present a practical solution proposal through the design, construction, and execution of a quantum circuit. The main methodology involves the application of Grover's algorithm to perform an exhaustive search among the possible solutions, implemented using the Qiskit library. The proposed approach aims to demonstrate the satisfactory and unsatisfactory solutions of a given Boolean expression in conjunctive normal form (CNF). Furthermore, it presents the results and discusses their empirical validity through a comparison with pre-programmed abstractions from the aforementioned library, and then proves its robustness using a truth table.*

Resumo. *A expansão do estudo teórico e prático de soluções quânticas tornou-se notável nos últimos anos. Com base nesse crescimento mencionado, este meta-artigo tem como objetivo investigar não somente o estado da arte de estratégias de circuitos quânticos para solucionar o clássico e dispendioso problema da satisfabilidade mas como também apresentar uma proposta de resolução de forma prática, através da diagramação, construção e execução de um circuito quântico. O trabalho utiliza como metodologia principal uma aplicação do algoritmo de Grover, a fim de realizar uma busca exaustiva dentre as possibilidades possíveis, codificado com a biblioteca qiskit. A proposta apresentada visa mostrar as soluções satisfatórias e não satisfatórias de uma dada expressão booleana em forma normal conjuntiva. Além disso, apresenta os resultados e discute sua validade empírica através de uma comparação com abstrações pré-programadas da biblioteca citada anteriormente, e então comprovar sua solidez a partir de uma tabela verdade.*

1. Introdução

O constante avanço da computação e a crescente demanda por soluções mais eficientes têm conduzido a um cenário de busca contínua por algoritmos que otimizem a resolução de problemas complexos. Tanto na ciência e na produção de hardware quanto no mercado de trabalho, algoritmos clássicos têm sido pilares fundamentais para uma ampla gama de aplicações. No entanto, o advento da computação quântica promete revolucionar a forma

como abordamos essas questões, oferecendo um novo paradigma de processamento que pode superar as limitações dos algoritmos clássicos em determinados contextos e superar o limite físico apresentado nos computadores atuais. Enquanto a computação clássica utiliza bits que podem estar em estados 0 ou 1, a computação quântica utiliza qubits que podem existir em estados de sobreposição, representando simultaneamente 0 e 1. Essa capacidade de processamento não determinístico é o que confere à computação quântica sua notável vantagem em certos tipos de cálculos. k-SAT é um desafio importante na ciência da computação que apresenta grandes questões ainda não descobertas, e, portanto, foi escolhido para ser desbravado no artigo em questão.

Este trabalho, fundamentado nos pontos discutidos, tem como objetivo propor um diagrama de circuito para resolver a expressão em FNC apresentada conforme a equação 2. Ademais, pretende construir e testar um circuito quântico utilizando a biblioteca *qiskit*, além de comparar os resultados obtidos para confirmar se a solução encontrada é ótima. A fim de alcançar esses objetivos, a seção 2 expõe os princípios teóricos empregados na elaboração do trabalho, com o fito de estabelecer as bases para uma compreensão clara da proposta em discussão. Em sequência, a seção 3 apresenta as soluções de diversas pesquisas que buscam resolver o complexo problema da satisfação, por sua vez empregadas para elaborar a proposta final apresentada. Logo após, a seção 4 detalha como os trabalhos citados contribuíram para a solução do circuito, as decisões tomadas durante a elaboração deste e, enfim, apresenta o circuito finalizado, que é seguido pela seção 5, onde se realiza a execução do circuito e os resultados são comparados com os de uma tabela verdade. Por último, a seção 6 desenvolve uma conclusão sobre a solução obtida, e reforça a conclusão dos propósitos apresentados, incluindo então ideias para trabalhos futuros.

2. Referencial Teórico

Antes de entrar na implementação proposta, é necessário o entendimento de alguns conceitos da computação clássica, como: Máquina de Turing, problema NP, k-SAT, uma breve introdução a computação quântica, e demais conceitos dela, como Algoritmo de Grover e Porta Hadamard.

A Máquina de Turing é um modelo abstrato de computação que utiliza uma fita infinita dividida em células discretas para manipular símbolos com base em regras definidas. Apesar de sua simplicidade, esse modelo é capaz de executar qualquer algoritmo computacional. A máquina consiste em uma cabeça que percorre a fita, podendo ler e escrever símbolos em cada célula. Cada célula contém um símbolo retirado de um conjunto finito de símbolos. Além disso, a máquina opera em um conjunto finito de estados. No início, a fita contém apenas a sequência de entrada e está vazia em todos os outros lugares. Caso a máquina precise armazenar informações, ela pode escrevê-las na fita. Para ler as informações escritas, a máquina pode mover sua cabeça de volta sobre elas. A computação continua até que a máquina decida produzir uma saída. As saídas "aceitar" e "rejeitar" são determinadas ao entrar em estados especificamente designados para tal. Caso não entre em nenhum estado de aceitação ou rejeição, a máquina continuará indefinidamente, sem parar.

Os problemas NP são aqueles para os quais, dada uma solução, é possível verificar, no pior caso, utilizando uma Máquina de Turing não determinística, em tempo limitado polinomial [Sipser 2007]. Acontece que, para determinados problemas, ao executar esse

mesmo problema só que em uma máquina de turing determinística, obtemos um tempo limitado não polinomialmente. Por conta disso, os problemas NP são considerados os mais custosos de serem resolvidos computacionalmente, e, especialmente em seu pior caso, e, por conta disto, apresentam uma grande importância na teoria da computação.

Compreendendo o conceito de problema NP, é possível chegar no tema sobre o problema de satisfabilidade, e detalhar sua característica. O problema SAT é um dilema combinatório de grande relevância teórica e prática. Assim como outros problemas combinatórios, a busca por uma solução consome tempo, e esse tempo aumenta de maneira exponencial conforme o tamanho das entradas. Esse problema foi um dos primeiros a ser reconhecido como NP-Completo [Cook 1971], possuindo uma complexidade de $O(2^n)$. Seja $x_1, \ldots, x_n$ variáveis booleanas, um literal pode ser uma variável booleana x_1 ou sua negação $\neg x_1$. Uma fórmula k-FNC (Forma Normal Conjuntiva) é uma conjunção de cláusulas, onde cada cláusula é uma disjunção de exatamente k literais. O problema k-SAT é dado um k-FNC, decidir se é satisfazível ou não.

Antes de entrar no assunto da computação quântica, é necessário entender o conceito de processamento de informação quântico, que, é uma área de pesquisa que inclui campos como: computação quântica, criptografia quântica e comunicação quântica [Rieffel and Polak 2011]. Tal área tem como principal característica o uso de conceitos da mecânica quântica ao invés do uso da física tradicional para manipular informação, e consequentemente o seu processamento. Olhando a este ângulo, podemos entender que a computação quântica não altera apenas os componentes de um computador, mas também altera o conceito do que é um computador: por exemplo, enquanto na computação clássica utilizamos bits a fim de representar informações, onde os mesmos podem assumir dois valores: 0 e 1, na computação quântica é utilizado o conceito de *qubit*, onde tal medida de informação não apenas pode representar o valor 0 e 1, mas também pode apresentar uma sobreposição destes estados. Tal característica permite quebrar barreiras de processamento que implicam em diversos usos práticos, como por exemplo: quebra de criptografia RSA [Shor 1997], aceleração de pesquisa de items em um conjunto de dados [Grover 1996], resolver problemas de otimizações combinatórias [Farhi et al. 2014], dentre outros.

O algoritmo de Grover [Grover 1996] é um método quântico para buscar um elemento alvo em um conjunto de dados não ordenado, e é significativamente mais rápido do que qualquer algoritmo clássico para o mesmo problema, apresentando uma complexidade de $O(\sqrt{N})$. Sua aplicação é descrita por meio de um difusor descrito por Grover conforme a equação 1, onde F representa a porta Hadamard, comumente utilizada para colocar o sistema em sobreposição, e R, uma matriz diagonal

$$D = FRF \tag{1}$$

é necessário entender que, para implementar o mesmo, o estado da arte define o uso de operadores de sobreposição quântica , como no caso da porta de Hadamard.[Silva 2018] O termo Hadamard se refere à geração de bits aleatórios. O Hadamard, neste contexto, está relacionado à Transformada de Hadamard, que é uma operação linear fundamental em computação quântica. Ela é usada para criar superposições de estados quânticos, permitindo que um qubit esteja em uma combinação de estados 0 e 1 simultaneamente. Isso é crucial para algoritmos quânticos, como a geração de números aleatórios, pois

permite explorar o paralelismo quântico e realizar cálculos em muitos estados possíveis ao mesmo tempo. A Transformada de Hadamard é representada por uma matriz específica que, quando aplicada a um vetor de estado de um qubit, resulta na superposição desejada de estados.

Com base no entendimento dos conteúdos citados, é possível desenvolver uma discussão sobre as propostas apresentadas no artigo para a solução do problema em questão.

3. Trabalhos Relacionados

Para embasar este trabalho e identificar estudos relacionados relevantes, foi realizada uma pesquisa na base de dados Scopus, uma das mais abrangentes para artigos acadêmicos e publicações científicas. A consulta foi feita utilizando a seguinte query: *"k-sat AND quantum AND computing AND grover"*, considerando o período de 2019 a 2024[1]. Essa busca resultou em 65 trabalhos.

Primeiramente, foi realizado uma filtragem com base em três principais categorias que podem ser categorizadas entre *sim* e *não*, sendo elas:

1. Se o trabalho encontrado pertence a categoria "Título", onde é categorizado se o título apresenta qualquer relação com computação quântica e o problema da satisfabilidade;
2. Se o trabalho encontrado pertence a categoria "Grover", que categoriza se o trabalho utiliza o algoritmo de Grover como metodologia para a resolução do problema;
3. Se trabalho encontrado pertence a categoria "k-SAT", que afirma para validar que o trabalho está resolvendo o mesmo problema em questão.

Ao conduzir o processo de filtragem inicial, foram identificados e selecionados 17 trabalhos relevantes dentro do escopo proposto. Em seguida, foi realizada uma triagem detalhada e rigorosa, que consistiu na leitura minuciosa das introduções e conclusões de cada trabalho. Esse procedimento visou avaliar com maior precisão a pertinência e a relevância dos estudos em relação às questões centrais abordadas no presente artigo. Como resultado dessa análise aprofundada, o conjunto inicial foi reduzido a um total de 6 estudos, os quais representam as contribuições mais significativas para a fundamentação teórica e metodológica desta pesquisa, sendo apresentados de forma organizada na Tabela 1.

Trabalho	**Título**	**Grover**	**Quantum**	**k-SAT**
[Yang et al. 2024]	Sim	Não	Sim	Sim
[Lin et al. 2024]	Não	Não	Sim	Sim
[Varmantchaonala et al. 2023]	Sim	Sim	Sim	Sim
[Wang et al. 2020]	Sim	Sim	Sim	Sim
[Mandl et al. 2024]	Sim	Não	Sim	Não
[Piro et al. 2020]	Sim	Sim	Sim	Sim

Tabela 1. Categorias de filtragem

[1]Data da consulta: 08 de setembro de 2024.

Assim como em [Lin et al. 2024], é explicado que, a maneira mais comum de se resolver um problema k-SAT por meio da computação quântica, seria um circuito composto em dois principais componentes: um oráculo, e um difusor, onde o oráculo seria a nossa entrada, que visa responder "sim" ou "não" para o nosso problema, e o difusor se encaminha de maximizar a probabilidade de se obter o resultado esperado ao medir o estado dos *qubits*.No estudo em questão, é introduzido a forma mais comum de se criar um oráculo para uma expressão lógica em FNC, que para dado uma fórmula $\mathcal{F} : (a) \wedge (\overline{a} \vee b) \wedge (\overline{a} \vee c)$, tendo como três variáveis *booleanas* $a = 1, b = 1, c = 1$, existe um oráculo U composta por três blocos conectados entre si C_1, C_2, C_3 onde C_1 visa processar a primeira cláusula (a), C_2 processa a segunda cláusula $(\overline{a} \vee b)$, e por fim C_3 processa a terceira $(\overline{a} \vee c)$, onde são processadas de forma sequencial devido ao fato de todas cláusulas dependerem da variável a. Cada cláusula por sua vez é composta por portas M_j, onde cada um representa literal l_j, onde caso l_j seja uma negação, é inferido uma porta X, caso o contrário, uma porta identidade I. O qubit obtido então representa o valor da cláusula C_1. Após a junção das cláusulas é obtido assim um bloco correspondente forma geral ω apresentado na figura 1. Após a construção de todos os circuitos, eles são conectados por uma porta CNOT. (bloco $\wedge$), seguido por uma porta Z, e, por fim, é implementado o bloco Ω^{-1}, que representa a operação inversa de Ω a fim de restaurar o estado do vetor de entrada.

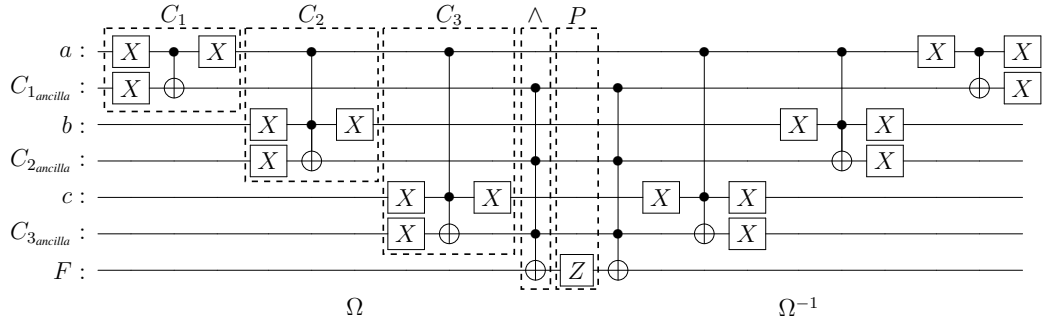


Figura 1. Forma geral do oráculo baseado em [Lin et al. 2024]

Alguns trabalhos visam não apenas propor uma resolução para o clássico problema da satisfabilidade, como também realizar otimizações que reduzem a quantidade de qubits totais de um circuito quântico SAT, como em [Yang et al. 2024] onde além de propor um algoritmo para a criação de uma oráculo, também é feito uma redução na quantidade de qubits *ancilla* existentes em um SAT-oráculo, que podem ser definidos como *qubits* extras incorporados no circuito a fim de armazenar o resultado de cada disjunção de uma expressão FCN. O trabalho menciona que, para construir uma oráculo para uma dada expressão em FNC, é comumente utilizado uma quantidade de qubits *ancilla* de $2m - 1$, sendo m a quantidade de cláusulas da expressão em questão. O artigo sugere uma solução para diminuir o uso e a utilização de recursos computacionais quânticos, ao propor um oráculo com quantidade ajustável de qubits *ancilla*, recurso especialmente útil para circuitos que resolvem expressões com centenas (ou senão milhares) de cláusulas. O estudo em questão começa com a explicação do que são e do motivo pelo qual devemos empregar oráculos em circuitos quânticos, e, juntamente com a explicação do problema SAT, enfatiza-se teoricamente como esses oráculos devem ser utilizados em circuitos que visam resolver o problema da satisfabilidade, reforçando a falha dos algoritmos convencionais quando o assunto é a limitação da quantidade de qubit *ancilla* Após o uso de um

algoritmo que possibilita a limitação da quantidade de *qubits ancilla*, é implementado o oráculo de forma prática, e, então é explicado que a expressão proposta pode ser satisfeita através da aplicação do algoritmo de Grover.

Conforme explicado anteriormente, uma forma geral para um circuito que visa resolver o problema do 3-SAT pode ser apresentado através de um oráculo que representa uma expressão FNC, e um difusor que amplifica e realizar a busca exaustiva do resultado da expressão apresentada. Em [Wang et al. 2020] O artigo cita a vantagem de utilizar Grover por conta de sua complexidade $O\left(\sqrt{N}\right)$ em uma busca não estruturada em uma base de dados. Munido deste conhecimento, o estudo além de montar um oráculo que representa uma expressão em FNC, também apresenta a viabilidade do uso do algoritmo de Grover como difusor. Também é explicado que tal abordagem permite que a complexidade total do circuito seja equivalente a $O(2^{n/2})$. Assim como no trabalho citado anteriormente, o circuito consiste na montagem de um oráculo representante da expressão em FNC, e logo em seguida uma amplificação com um difusor (nesse caso específico, o algoritmo de Grover) a fim de realizar a busca exaustiva pelas soluções satisfatórias da expressão em questão.

Cabe-se ressaltar que os trabalhos relacionados citados até então introduzem uma solução que consiste em três principais etapas: 1. Criação de um oráculo que representa a expressão FNC para o estado inicial, 2. Sobreposição uniforme de estados das cláusulas representadas por cada qubit 3. Em conjunto com o algoritmo de Grover, é realizado uma busca exaustiva no espaço de busca em conjunto. (veja a figura 2).

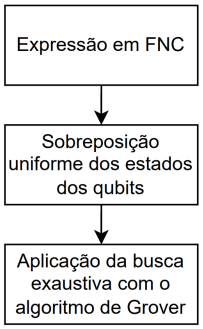


Figura 2. Diagrama de fluxo da montagem dos circuitos apresentados até então

A este ponto, vale citar outros trabalhos que visam acabar com essa abordagem custosa e apresentar melhorias, como em [Varmantchaonala et al. 2023], onde, através de um algoritmo proposto no trabalho, é realizado uma subdivisão do estado inicial, que, conforme citado anteriormente e nos outros trabalhos já é computado exaustivamente com o algoritmo de Grover. Ao invés disso, e de forma não uniforme, é feito uma aplicação de divisão em conquista que visa realizar uma sobreposição de parte do espaço de busca, contando com apenas os valores que são potencialmente uma solução ótima. Tal proposta não uniforme e baseada em divisão em conquista apresenta uma execução do algoritmo de Grover com complexidade $O\left(\sqrt{N/S}\right)$, de sendo N o número de potenciais soluções, e S o fator de redução/subdivisão de espaço utilizado pelo algoritmo. Trazendo uma complexidade relativamente superior ao algoritmo de Grover, que em seu estado da arte, e em conjunto com os métodos uniformes citados anteriormente, apresenta uma complexidade

de $O\left(\sqrt{N}\right)$. No próprio trabalho em questão, também é citado que existem outras maneiras de solucionar o problema do SAT de forma não determinística, mas sim do uso de meta heuristicas, como através da utilização de QAOA (*Quantum Approximate Optimization Algorithm*). Adentrando nessa descoberta, outros trabalhos que utilizam tal *approach* são dignos de serem citados.

Da mesma forma que brevemente citado no trabalho explicado anteriormente, em [Mandl et al. 2024], é apresentado um algoritmo visado ao uso de meta heurísticas para a solução do problema MAX-3SAT(uma generalização do problema SAT), mais em específico o QAOA (*Quantum Approximate Optimization Algorithm*), um algoritmo quântico utilizado para resolver problemas de otimização combinatória, que em sua essência procura encontrar os estados com autovalor mínimo de um dado operador Hamiltoniano H_C. Através de uma *ansatz* modelada para o problema a ser resolvido que pode ser resumida em circuito em seu estado inicial seguido por p repetições que concatenam um operador de separação de fase $U(H_C, \gamma_i)$ e um circuito *mixer* $U(H_B, \beta_i)$. Tal arranjo permite a identificação de um conjunto solução para o problema.

Por fim, o artigo [Piro et al. 2020] avalia a aplicação de computação quântica para a solução de instâncias arbitrárias do problema Exactly-1 K-SAT e mostra as melhorias que são proporcionadas por um solucionador quântico. O problema Exactly-1 K-SAT consiste em encontrar uma atribuição satisfatória na qual exatamente um literal é verdadeiro em cada cláusula. Este trabalho realiza uma generalização de outro solucionador quântico que resolve o problema Exactly-1 3-SAT utilizando o algoritmo de Grover, essa generalização é feita em três etapas:

1. Inicialização, onde é aplicado portas Hadamard para cada variável e para a saída;
2. Codificação do problema, onde cada cláusula é codificada no circuito usando portas que permitem inverter o qubit correspondente à cláusula se ela tiver exatamente um literal verdadeiro;
3. Inversão sobre a média, onde foi modificado os coeficientes correspondentes à solução correta aplicando a operação unitária W:

$$W = \left(-\bigotimes^{n} \mathcal{H} \right) D \left(\bigotimes^{n} \mathcal{H} \right)$$

4. Metodologia

Primeiramente, foi necessário a escolha de qual ferramenta seria utilizada para a montagem e execução do circuito. Para isso, foi utilizado a biblioteca *qiskit*, escolhida por possuir uma grande gama de funções prontas que facilitam a montagem do código, e pelo fato de ter sido o *framework* mais utilizado nos trabalhos encontrados durante a elaboração do texto. Para adentrar na abordagem prática para a resolução do problema, foi feito um breve estudo sobre como utilizar a biblioteca *Qiskit*, ferramenta altamente utilizada para desenvolvimento e execução de circuitos quânticos. Em consonância com o conteúdo exposto no livro [Silva 2018], foi empreendido um estudo aprofundado sobre circuitos quânticos. Tal investigação permitiu não apenas a compreensão das complexidades teóricas, mas também a aplicação prática dos conceitos abordados. Adicionalmente, foi conduzida uma análise rigorosa sobre a implementação do algoritmo de Grover, conforme delineado no trabalho de [Gamberi and Bianchini 2022]. Essa investigação possibilitou a

atualização do procedimento para a versão mais recente do *Qiskit*, considerando que este empregava uma versão anterior da biblioteca em questão.

Na sequência do estudo sobre a base do que foi estudado, procedeu-se à análise de quais algoritmos e estruturas de circuitos deveriam ser utilizadas na formulação do circuito proposto. Dentre as abordagens estudas, cabe-se citar as mesmas propostas em [Mandl et al. 2024], que conforme explicado anteriormente consiste no uso de meta heurísticas e aplicações matemáticas para a criação de um circuito que analisa as possibilidades do problema MAX 3-SAT, o trabalho feito em [Fernandes and Dutra 2019], [Lin et al. 2024] e [Wang et al. 2020], que ao invés do uso de meta-heurísticas, uma abordagem determinística é apresentada, uma vez que ao invés de se basear em um evento probabilístico, as abordagens em questão apresentam uma busca exaustiva pelo resultado a fim de determinar a solução ótima.

Após a definição de todas as abordagens já utilizadas em estudos e no estado da arte, foi necessário a escolha de qual dos métodos seria o mais interessante, simples, de fácil entendimento e prático para a montagem do circuito. Com isto em mente, foi determinado que a solução deveria seguir uma abordagem determinística, a fim de apresentar uma abordagem voltada ao desenvolvimento de *software* apresentada pela mesma em contraste com uma abordagem matemática definida pelo uso de meta-heurísticas.

Em consonância com os trabalhos determinísticos estudados anteriormente, foi necessário entender primeiramente o diagrama do circuito que mais tarde seria portado e executado em um *software*. Dentre os diagramas de circuitos apresentados nos trabalhos relacionados, foi escolhido a abordagem realizada por [Fernandes and Dutra 2019], que, por mais que os demais outros trabalhos já citados apresentem uma possível solução mais eficiente, o mesmo foi escolhido devido ao fato de apresentar a solução original para o problema. O diagrama consiste nas mesmas etapas ilustradas na figura 2, onde o circuito é dividido em dois principais blocos: um oráculo que representa a expressão em FNC escolhida e um difusor composto por uma diagonal D_{x} que se encaminha de maximizar a probabilidade de obter a solução ótima, conforme a figura 3.

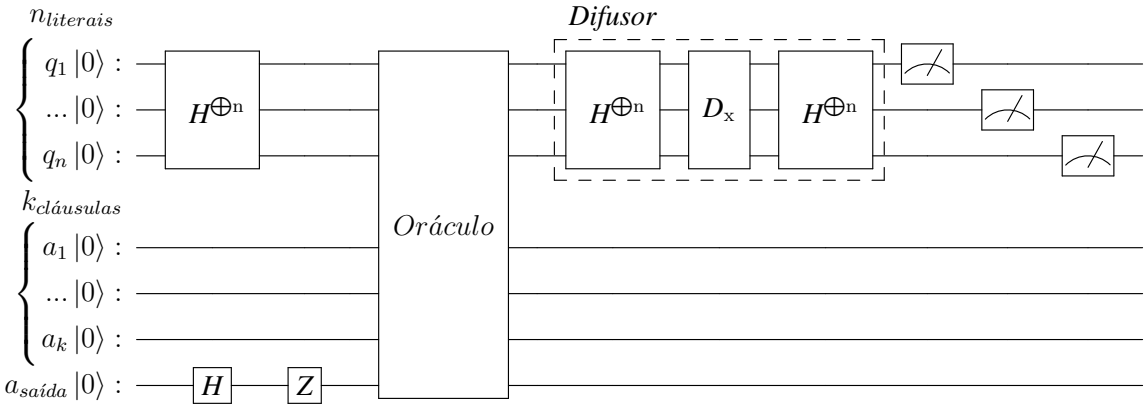


Figura 3. Diagrama para um circuito resolvedor 3-SAT por meio de um oráculo e um difusor

Com o diagrama geral do circuito em mãos, os próximos passos foram: 1. Criar um oráculo e 2. Definir e criar um difusor:

Para o oráculo. foi feito um estudo baseado nos trabalhos de

[Fernandes et al. 2019], [Fernandes and Dutra 2019], [Lin et al. 2024], que descorrem sobre maneiras de se construir a mesma através de uma dada expressão em FNC (veja a figura 1. Desta forma, para a implementação da mesma, foi criado um circuito que representa a expressão FNC conforme a equação 2.

$$(\neg A \vee \neg B \vee \neg C) \wedge (A \vee \neg B \vee C) \wedge (A \vee B \vee \neg C) \tag{2}$$

Em relação a quantidade de *qubits*, é possível definir conforme a equação 3,

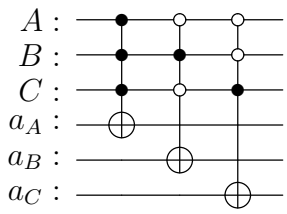
$$qtd_{qubits} = n + k + 1 \tag{3}$$

onde n representada a quantidade de literais, k a quantidade de cláusulas, e qtd_{qubits} a quantidade de *qubits* do circuito. Substituindo os valores conforme a equação 2, obtemos um total de 3 literais e 3 cláusulas, ou seja, $n = 3$ e $k = 3$, e somando com mais um *qubit* para armazenar o resultado conjunção das expressões, resultando, portanto, em um total de 7 *qubits* para o circuito. Em relação a escolha de uma expressão com 3 literais e 3 cláusulas, o motivo se deve ao fato das limitações da computação quântica quando a quantidade de *qubits* de um circuito é aumentada. Como a quantidade de literais e cláusulas afetam nesses fatores, foi optado por seguir com o 3-SAT, uma vez que o mesmo já foi comprovado como NP completo [Cook 1971], e a quantidade de *qubits* obtidas não demonstram valores que possam causar um grande ruído nos resultados obtidos do circuito.

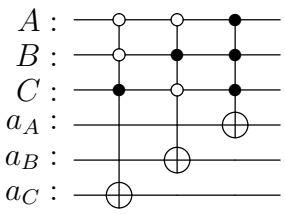
A principal ideia por trás dos *qubits* adicionais (de acordo com a lógica da equação 3, o valor de k) é servir como um *ancilla* para armazenar o resultado da disjunção de cada cláusula, e, portanto, precisamos de uma quantidade de *qubits ancilla* equivalente a quantidade de cláusulas escolhidas. Desta forma, o principal objetivo de cada bloco de cláusula C (figura 1), é representar a negação ou identidade de cada literal da disjunção em questão, e atribuir o resultado da disjunção em seu respectivo *ancilla* por meio de uma porta Toffoli, onde cada literal é um *qubit* de controle (onde seu estado (1/0) é representado caso haja negação ou não, respectivamente), e o *qubit target* é seu respectivo *ancilla* (figura 4a). Em sequência da montagem das cláusulas e a fim de representar a conjunção ($\wedge$), é colocado uma porta Toffoli que tem como controle os *qubits* ancilla (em seus atuais estados), e como *target* o *qubit* extra separado anteriormente para a saída da expressão. Após este ponto, a ideia principal neste momento é reverter o circuito ao seu estado inicial. Tendo isto em mente, é construído outro bloco composto de portas Toffoli, só que agora representando o espelho da expressão montada anteriormente, isto é, a montagem de cada cláusula já feita, mas em ordem inversa. (veja a figura 4b).

Unindo todos os componentes citados anteriormente, é possível obter então o circuito final para o oráculo (veja a figura 5)

Após a implementação do oráculo, foi realizado um estudo para a implementação de um difusor a fim de seguir o estado da arte para implementar o algoritmo de Grover. Da mesma forma que em [Gamberi and Bianchini 2022], foi aplicado o difusor como em seu estado da arte, definido conforme a equação 1 sendo *F* a aplicação da porta Hadamard (no caso para este trabalho, seria a aplicação para os n *qubits*), ou seja, $F = H^{\oplus n}$, e D uma matriz diagonal definida por um vetor sendo o primeiro elemento igual a 1, e os demais assumem o valor de -1:
$D_{i,j} = 0$ se $i \neq j$



(a) Circuito quântico para a equação 2



(b) Simetria da equação 2

Figura 4. Representação da montagem da expressão escolhida por meio de um circuito composto por portas Toffoli e qubits ancilla

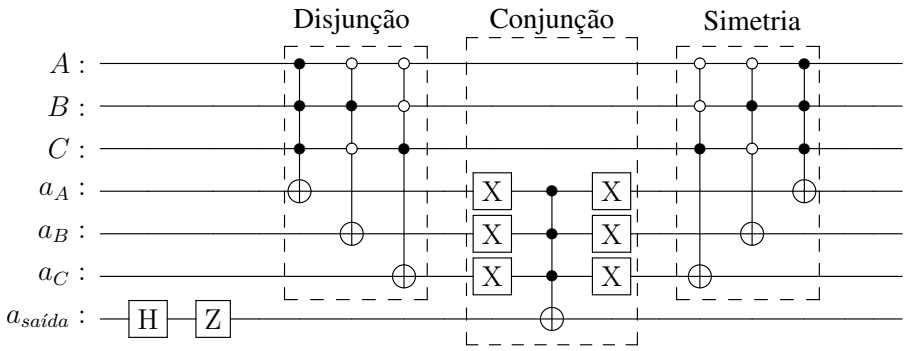


Figura 5. Oráculo criado que representa a expressão FNC conforme a equação 2

$D_{i,i} = 1$, se $i = 0$
$D_{i,i} = -1$, se $i \neq 0$

Tendo em mente o diagrama do circuito proposto conforme a figura 3, o próximo para a resolução do problema foi a criação do circuito. Para essa etapa, foi escolhido a linguagem de programação *Python* em conjunto com a biblioteca *Qiskit*. O primeiro passo então foi a criação do oráculo, que em suma pode ser aplicada da mesma maneira como no bloco 1

```
def CreateOracle():
  qubit_qtd = clauses_qtd = 3
  register_qtd = qubit_qtd + clauses_qtd + 1

  qreg_q = QuantumRegister(register_qtd, 'q')
  creg_c = ClassicalRegister(qubit_qtd, 'c')
  qc = QuantumCircuit(qreg_q, creg_c)

  # Amplifica e inverte a fase do qubit de resultado
  qc.h(6)
  qc.z(6)
  qc.barrier()

  # Disjuncao (~A | ~B | ~C)
  qc.append(CXGate().control(num_ctrl_qubits=2, ctrl_state='11'), [0,1,2,3])
  # Disjuncao (A | ~B | C)
  qc.append(CXGate().control(num_ctrl_qubits=2, ctrl_state='00'), [0,2,1,4])
  # Disjuncao (A | B | ~C)
  qc.append(CXGate().control(num_ctrl_qubits=2, ctrl_state='00'), [0,1,2,5])

  # Conjuncao
  qc.barrier()
```

```
  qc.x(qreg_q[3])
  qc.x(qreg_q[4])
  qc.x(qreg_q[5])
  qc.append(CXGate().control(num_ctrl_qubits=2, ctrl_state='11'), [3,4,5,6])
  qc.x(qreg_q[3])
  qc.x(qreg_q[4])
  qc.x(qreg_q[5])
  qc.barrier()

  # Simetria (A | B | ~C)
  qc.append(CXGate().control(num_ctrl_qubits=2, ctrl_state='00'), [0,1,2,5])
  # Simetria  (A | ~B | C)
  qc.append(CXGate().control(num_ctrl_qubits=2, ctrl_state='00'), [0,2,1,4])
  # Simetria  (~A | ~B | ~C)
  qc.append(CXGate().control(num_ctrl_qubits=2, ctrl_state='11'), [0,1,2,3])

  qc.barrier()

  return qc
```

Listing 1. Criação do oráculo

Com a criação do oráculo, os passos restantes se definem em:
1. Criação da diagonal (veja o trecho de código 2)
2. Criação e anexo do difusor ao oráculo (veja o trecho de código 3)

```
def CreateDiagonalSequence(number, qubits):
    diagonalSize = pow(2,qubits)
    if (diagonalSize < number - 1): return -1
    aux = np.ones(diagonalSize, dtype=int)
    aux[number] = -1
    return aux

def GetDiagonal():
  # Quantidade de literais (A,B,C)
  qubit_qtd = 3
  groverDiagonal = list(CreateDiagonalSequence(0, qubit_qtd))

  # Instancia da diagonal disponibilizada pelo qiskit
  return Diagonal(groverDiagonal)
```

Listing 2. Criação da diagonal

```
def ApplyGrover(oracle):
  num_qubits = 7
  qr = QuantumRegister(num_qubits)
  qc = QuantumCircuit(qr, name)
  qubits = [0,1,2,3,4,5,6]
  qc.compose(oracle, qubits, inplace=True)

  # Aplica Hadamart Para cada qubit queremos medir
  qc.h(0)
  qc.h(1)
  qc.h(2)

  #Aplica a diagonal para os 3 qubits que desejamos medir
  qc.compose(
      GetDiagonal(oracle),qubits=[0,1,2], inplace=True
  )
  # Aplica Hadamart Para cada qubit que queremos medir
  qc.h(0)
  qc.h(1)
  qc.h(2)

  return qc
```

Listing 3. Adição do operador de Grover ao *oráculo*

Com a aplicação de Hadamard nos 3 primeiros *qubits* (literais), e a incluindo a medição dos mesmos, o circuito final então é obtido (veja a figura 6)

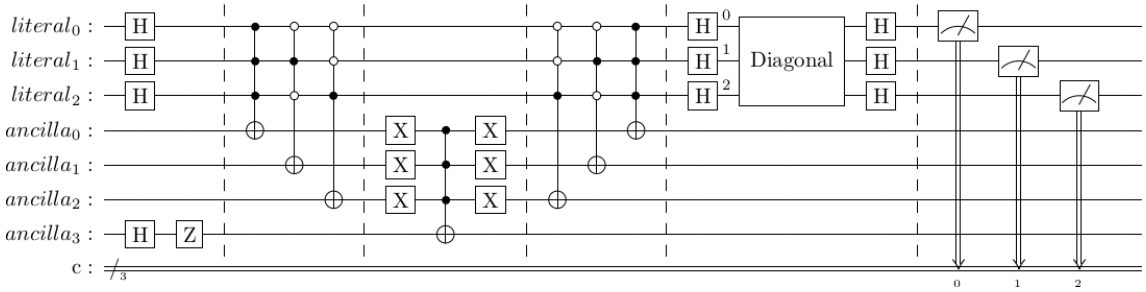


Figura 6. Circuito final

5. Resultados obtidos

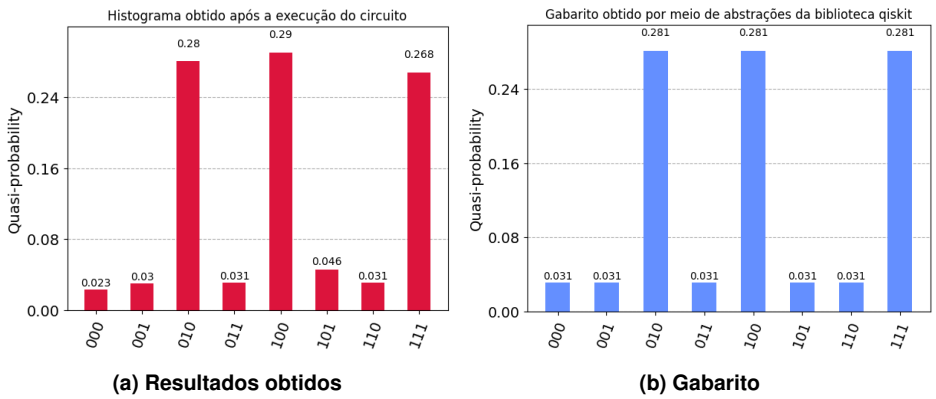
Conforme explicitado anteriormente, foi criado o circuito quântico na figura 6 que visa solucionar a expressão FNC 2.

Executando o circuito obtido, é possível gerar o histograma presente na figura 7a, que representa as atribuições (1/0) dos valores de C, B e A que resultam em não satisfação para a expressão 2. No caso em questão, os valores 000, 001, 011, 101, 110 representam as atribuições de C,B,A, respectivamente para a obtenção da satisfabilidade.

Sendo assim, as atribuições com maior probabilidade indicam os valores de C, B e A que levam a um resultado booleano de "falso", enquanto as com menor probabilidade correspondem às atribuições que resultam no valor booleano de "verdadeiro"para a expressão.

A fim de visualizar comparações com o estado da arte para a resolução do problema, foi utilizado uma implementação de abstrações de circuitos pré programados da biblioteca Qiskit como gabarito, resultando no histograma representado na figura 7b. Os resultados expressados nas figuras 7a e 7b se encaixam perfeitamente nos valores de atribuição para C, B, A a fim de obter a satisfabilidade da equação, como comprovado na tabela verdade exposta em 2.

Apesar desta abordagem apresentar uma solução sem o uso de heurísticas, por



(a) Resultados obtidos **(b) Gabarito**

Figura 7. Resultados de não satisfação da expressão FNC 2

C	B	A	$(\neg A \vee \neg B \vee \neg C) \wedge (A \vee \neg B \vee C) \wedge (A \vee B \vee \neg C)$
F	F	F	T
T	F	F	F
F	T	F	F
T	T	F	T
F	F	T	T
T	F	T	T
F	T	T	T
T	T	T	F

Tabela 2. Tabela verdade da expressão FNC exposta na equação 2

conta da característica determinística intrínseca à computação quântica, o histograma obtido em 7a apresenta ruídos esperados, uma vez que os valores apresentados não demonstram valores definitivos das atribuições de entradas, mas sim a probabilidade de cada entrada reproduzir uma solução não satisfatória. Na tabela 2 os mesmos padrões são observados, o que reforça a consistência e a validade da abordagem quântica. Em contraste, podemos afirmar que uma abordagem heurística por sua vez apresentaria diversas soluções, onde cada uma delas possui uma probabilidade de sucesso, como visto em [Mandl et al. 2024].

6. Conclusões e trabalhos futuros

Em conclusão, os resultados obtidos durante a aplicação deste artigo demonstram o potencial da computação quântica para a resolução de problemas clássicos, como no caso do 3-FNC-SAT. Em conjunto com as comparações realizadas na seção 5, é possível afirmar que os resultados obtidos expressam uma solução ótima, cumprindo, portanto, o objetivo deste artigo de se resolver o problema proposto. A característica não determinística da computação quântica nos permite obter resultados de várias maneiras. Especificamente, o algoritmo de Grover provou ser uma ferramenta versátil para lidar não apenas com buscar em uma base dados, mas também para a solução de problemas NP completos como demonstrado no presente trabalho.

Em relação a trabalhos futuros, cabe citar alguns pontos que não foram destrinchados durante este trabalho, como por exemplo o cálculo da complexidade de tempo e espaço do circuito obtido e compará-lo com a computação clássica, ou então a possibilidade de se generalizar a quantidade de literais, a fim de resolver o problema k-FNC-SAT, uma vez que, conforme explicitado anteriormente, foi escolhido uma expressão com 3 exatos literais tendo em mente o objetivo de manter uma quantidade pequena de *qubits*. Visando aumentar a escalabilidade do circuito, alguns trabalhos como em [Yang et al. 2024] introduzem alternativas para diminuir a quantidade de *qubits ancilla* que poderiam ser aplicadas para aumentar a quantidade de literais, já que a sua redução implica em mais *qubits* livres para adicionarmos ao circuito. Outra abordagem seria realizar um processamento de busca e um oráculo construido de forma paralela e distribuida, assim como em [Lin et al. 2024]. Também vale citar a possibilidade da criação de um circuito de classe P que tem como entrada um problema NP, e que ao ser executado o mesmo seria redu-

zido ao problema do k-SAT, colocando em prática sua NP completude comprovada por [Cook 1971].

O código-fonte desenvolvido é de uso livre e está disponível em um repositório no *Github*[2].

Referências

Cook, S. A. (1971). The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, page 151–158, New York, NY, USA. Association for Computing Machinery.

Farhi, E., Goldstone, J., and Gutmann, S. (2014). A quantum approximate optimization algorithm.

Fernandes, D. and Dutra, I. (2019). Using grover's search quantum algorithm to solve boolean satisfiability problems: Part i. *XRDS*, 26(1):64–66.

Fernandes, D., Silva, C., and Dutra, I. (2019). Using grover's search quantum algorithm to solve boolean satisfiability problems, part 2. *XRDS*, 26(2):68–71.

Gamberi, G. P. and Bianchini, C. D. P. (2022). Estudo de algoritmos quânticos e suas implementações. `https://dspace.mackenzie.br/items/071a8df2-1a8f-45c2-ab22-49749d06479b`.

Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, page 212–219, New York, NY, USA. Association for Computing Machinery.

Lin, S.-W., Wang, T.-F., Chen, Y.-R., Hou, Z., Sanán, D., and Teo, Y. S. (2024). A parallel and distributed quantum sat solver based on entanglement and teleportation. In Finkbeiner, B. and Kovács, L., editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 363–382, Cham. Springer Nature Switzerland.

Mandl, A., Barzen, J., Bechtold, M., Leymann, F., and Wild, K. (2024). Amplitude amplification-inspired qaoa: improving the success probability for solving 3sat. *Quantum Science and Technology*, 9(1):015028.

Piro, F., Askarpour, M., and Di Nitto, E. (2020). Generalizing an exactly-1 sat solver for arbitrary numbers of variables, clauses, and k. volume 2705, page 27 – 37. Cited by: 0.

Rieffel, E. and Polak, W. (2011). *Quantum Computing: A Gentle Introduction*. The MIT Press, 1st edition.

Shor, P. W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509.

Silva, V. (2018). *Practical Quantum Computing for Developers*. Apress L. P., 1st edition.

Sipser, M. (2007). *Introdução à Teoria da Computação: Tradução da 2ª edição norte-americana (trad. Ruy José Guerra Barreto de Queiroz)*. Thomson Learning, São Paulo.

[2] `https://github.com/gabrielms201/Quantum-Implementation-For-SAT-Problem-Solving`

Varmantchaonala, C. M., Fendji, J. L. K. E., Njafa, J. P. T., and Atemkeng, M. (2023). Quantum hybrid algorithm for solving sat problem. *Engineering Applications of Artificial Intelligence*, 121:106058.

Wang, P., Liu, G., and Liu, L. (2020). A generic variable inputs quantum algorithm for 3-sat problem. In *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications(AEECA)*, pages 308–312.

Yang, S., Zi, W., Wu, B., Guo, C., Zhang, J., and Sun, X. (2024). Efficient quantum circuit synthesis for sat-oracle with limited ancillary qubit.