



ARTIGO DE PESQUISA/RESEARCH PAPER

Simplificação de Multiplexadores Quânticos para Compilação de Software Quântico

Quantum Multiplexer Simplification for Quantum Software Compilation

José Alex de Carvalho  [Universidade Federal de Pernambuco | jac@cin.ufpe.br]

Adenilton José da Silva  [Universidade Federal de Pernambuco | ajsilva@cin.ufpe.br]

 Centro de Informática, Universidade Federal de Pernambuco, Av. Jorn. Aníbal Fernandes, s/n, Cidade Universitária, Recife, PE, 50740-560, Brasil.

Resumo. A inicialização de estados quânticos é uma das principais rotinas do processo de compilação de software quântico. Entretanto, o custo dessa etapa pode acabar comprometendo a eficiência total da computação. Este trabalho propõe um método de otimização para a inicialização de estados que podem ser fatorados em estados menores. Essa finalidade é obtida através da remoção de operadores e controles desnecessários para multiplexadores quânticos. A abordagem proposta é competitiva com as outras técnicas da literatura em número total de CNOTs. Ademais, o tempo de processamento clássico do método aqui proposto é significativamente menor do que o dos outros trabalhos que otimizam esse tipo de inicialização.

Abstract. Quantum state initialization is one of the main routines in the quantum software compilation process. However, the cost of this step can compromise the overall computing efficiency. This work proposes an optimization method for the initialization of states that can be factored into smaller states. This purpose is achieved by removing unnecessary operators and controls for quantum multiplexers. The proposed approach is competitive with other techniques in the literature in terms of the total number of CNOTs. Furthermore, the classical processing time of the method proposed here is significantly lower than that of other works that optimize this type of initialization.

Palavras-chave: Computação Quântica, Compilação Quântica, Estados Separáveis, Otimização de Multiplexadores.

Keywords: Quantum Computing, Quantum Compiling, Separable States, Multiplexer Optimization.

Recebido/Received: 22 May 2025 • Aceito/Accepted: 19 June 2025 • Publicado/Published: 11 July 2025

1 Introdução

A computação quântica é uma área interdisciplinar que busca utilizar as propriedades intrínsecas da mecânica quântica para o processamento de informações. Existe uma gama de problemas que podem ser resolvidos com um ganho considerável em eficiência computacional através do uso de algoritmos quânticos Grover [1996]; Shor [1997]. Entretanto, os dispositivos quânticos da atualidade ainda estão em uma etapa inicial de desenvolvimento. Dessa forma, limitações de hardware, como a grande taxa de ruído associado às operações Preskill [2018], são um desafio para o aproveitamento das vantagens da teoria quântica. Contudo, um progresso considerável vem sendo feito recentemente na área Wu *et al.* [2021] e, em especial, no desenvolvimento de estratégias como a mitigação de erros Kim *et al.* [2023].

Nesse contexto, o processo de compilação de software quântico deve ser capaz de traduzir a rotina quântica de alto nível em um conjunto de operações, a serem aplicadas em uma dada ordem, de acordo com o conjunto de portões quânticos disponíveis naquela arquitetura. Uma sub-rotina importante dentro do procedimento de compilação de software quântico é a etapa da preparação de estados quânticos (*QSP - Quantum State Preparation*). As rotinas de preparação de estados são responsáveis pela inicialização de variáveis durante a computação, por meio da busca de um operador unitário SP_x que inicializa uma variável x como uma variável quântica, levando o circuito quântico do estado inicial ao estado $|x\rangle$.

Entretanto, além das limitações citadas acima, existem questões relacionadas à própria mecânica quântica que levam a dificuldades que não existem na computação clássica. Por exemplo, o teorema da não clonagem indica que não é possível realizar a clonagem de um estado quântico Wootters *et al.* [1982]. Ademais, o fenômeno quântico de descoerência Hughes *et al.* [1996] torna obrigatória a necessidade de recarregar a informação em um intervalo de tempo regular. Portanto, o custo computacional da inicialização de estados, tipicamente exponencial, gera um possível impacto negativo no ganho teórico proposto para a computação quântica na resolução de problemas.

Este trabalho busca utilizar uma abordagem de representação intermediária para a inicialização de estados por meio de uma árvore sintática abstrata Araujo *et al.* [2021], permitindo identificar simplificações na preparação de uma categoria de estados quânticos que podem ser decompostos em um produto de outros estados. O algoritmo proposto busca essas simplificações de forma eficiente, minimizando o tempo necessário para detectar a separabilidade de estados. De um modo mais geral, o presente estudo propõe uma metodologia para a simplificação de multiplexadores quânticos, impactando a preparação de estados baseada em multiplexadores Bergholm *et al.* [2005], mas também podendo ser útil em diversas situações em que a presença desses multiplexadores é comum.

2 Simplificação de Multiplexadores Quânticos para QSP

Essa seção descreve a abordagem de simplificação proposta nesse trabalho e como ela impacta o procedimento de compilação de software quântico. A seção 2.1 descreve brevemente os conceitos fundamentais da computação quântica. O conceito de multiplexador quântico será introduzido na seção 2.2, assim como sua importância como componente primordial no procedimento de inicialização de estados, segundo a abordagem de Bergholm *et al.* [2005], que será detalhada na seção 2.3. A seção 2.4 descreve o funcionamento da metodologia de otimização a partir da árvore abstrata que representa a inicialização de estados, assim como propõe um algoritmo de busca por simplificações a partir dessa representação.

2.1 Estados Quânticos

As unidades básicas de informação na computação quântica são os bits quânticos, também conhecidos como qubits ou q-bits. Eles são descritos em um espaço vetorial complexo bidimensional. Um estado quântico corresponde à descrição matemática da informação que é armazenada em um sistema de n qubits. Cada estado de um qubit pode ser descrito como uma combinação linear dos estados da base computacional $|0\rangle$ e $|1\rangle$, onde

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

Dessa forma, o estado de um qubit tem a forma geral $|\psi_1\rangle = a|0\rangle + b|1\rangle$. Sendo a e b números complexos tais que $|a|^2 + |b|^2 = 1$. Para descrever um estado com n qubits, utilizamos um vetor de 2^n dimensões. Matematicamente, dizemos que um sistema não emaranhado de n qubits é descrito pelo produto tensorial dos estados desses n qubits, ou seja,

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_{n-1}\rangle \otimes |\psi_n\rangle = \bigotimes_{i=1}^n |\psi_i\rangle. \quad (1)$$

Definimos um estado quântico de mais de um qubit como emaranhado se não for possível descrever esse estado como um produto tensorial dos seus componentes. Por exemplo, tentar separar o estado $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ em um produto de dois qubits vai levar a um sistema impossível.

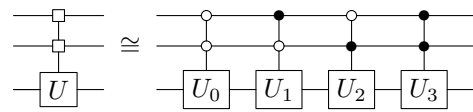
As operações realizadas nos estados quânticos são descritas através de um operador $U : \mathcal{H} \mapsto \mathcal{H}$, onde \mathcal{H} é o espaço de Hilbert associado aos vetores de estado. Esse operador U pertence a uma categoria de operadores chamados de operadores unitários, que são os operadores cuja matriz inversa corresponde ao seu operador adjunto, ou seja, $U^\dagger U = U U^\dagger = I$.

Um portão lógico de dois qubits particularmente importante é o portão CNOT, que muda o estado do qubit de alvo de $|0\rangle$ para $|1\rangle$, e vice-versa, desde que o valor no qubit de controle seja igual a $|1\rangle$. É possível decompor quaisquer operadores de n qubits em portões CNOT e em portões de um qubit; ademais, a taxa de erros associada a esse portão lógico é relativamente alta para os dispositivos quânticos atuais. Portanto, o número total de portões CNOT em um circuito é uma métrica de interesse para aplicações que executem em dispositivos quânticos na atualidade.

2.2 Multiplexadores Quânticos

Um multiplexador quântico U_t^k , ou portão uniformemente controlado, é uma porta lógica quântica que aplica um operador dentro um conjunto de operadores unitários $\{U_0, U_1, \dots, U_{N-1}\}$ no bit de alvo t de acordo com os valores dos k qubits de controle, onde $N = 2^k$. Dessa forma, é possível definir matematicamente um multiplexador como o operador $\sum_{i=0}^{2^k-1} |i\rangle\langle i| \otimes U_i$, onde $\{|i\rangle\}_{i=0}^{2^k-1}$ são vetores da base computacional do espaço dos k qubits de controle.

Na representação de circuitos, nós podemos representar o bit que atua como controle através de um símbolo de quadrado. Um círculo preenchido pela cor preta representa que um operador é aplicado quando o bit de controle é igual a $|1\rangle$. Por sua vez, um círculo em branco representa que o operador é ativado em $|0\rangle$. Por exemplo, temos a representação de um multiplexador com dois qubits de controle abaixo.



Um modo eficiente de implementar um multiplexador está descrito em Bergholm *et al.* [2005]. Uma aplicação importante para esses componentes dos circuitos quânticos é a síntese de operadores unitários Shende *et al.* [2006].

2.3 Preparação de Estados Quânticos

A preparação de estados quânticos (*QSP - Quantum State Preparation*) consiste na busca por um operador unitário SP_x que inicializa um vetor complexo $\vec{x} = [x_0, \dots, x_{2^n-1}]$ em um circuito quântico de n qubits através do estado final do circuito $|x\rangle = \sum_{i=0}^{2^n-1} x_i |i\rangle$. Portanto, $SP_x |e\rangle = |x\rangle$. Onde $|e\rangle$ representa o estado inicial do circuito. Dessa forma, essa é a etapa do procedimento quântico de compilação onde ocorre a inicialização de variáveis que serão utilizadas durante a execução de algoritmos quânticos.

Existem diversas formas de inicializar estados quânticos, utilizando diferentes abordagens ou uma mescla delas, como decomposição de operadores Plesch and Brukner [2011], árvores abstratas Mottonen *et al.* [2004]; Bergholm *et al.* [2005]; Araujo *et al.* [2021], *flip-flops* Park *et al.* [2019] ou algoritmos variacionais Nakaji *et al.* [2022]. Aqui, vamos descrever uma inicialização de estados através do uso de árvores.

Um problema equivalente ao de achar o operador SP_x é buscar por um operador SP_x^\dagger tal que $SP_x^\dagger |x\rangle = |0\rangle_n$. Por enquanto, vamos considerar $|e\rangle = |0\rangle_n$. Uma possível forma de encontrar o circuito que implementa esse operador é utilizar uma cascata de multiplexadores U_t^k , tais que cada multiplexador retira o emaranhamento de um qubit do sistema, ou seja, $U_t^k |a\rangle_i = |a'\rangle_{i-1} \otimes |0\rangle$. Onde $|a\rangle_i$ e $|a'\rangle_{i-1}$ representam estados intermediários.

Uma forma abstrata de representar esse procedimento é através de uma árvore. Os nós a uma distância h da raiz representam os operadores que são aplicados no qubit q_h . Uma seta tracejada ou sólida chegando a um nó indica que o operador é aplicado com controle em $|0\rangle$ ou em $|1\rangle$ com relação ao qubit q_{h-1} , respectivamente. Assim, o conjunto de setas da raiz até o nó em questão indica o conjunto de controles aos quais o operador está condicionado. A Figura 1

ilustra a árvore correspondente à inicialização de um estado de 3 qubits.

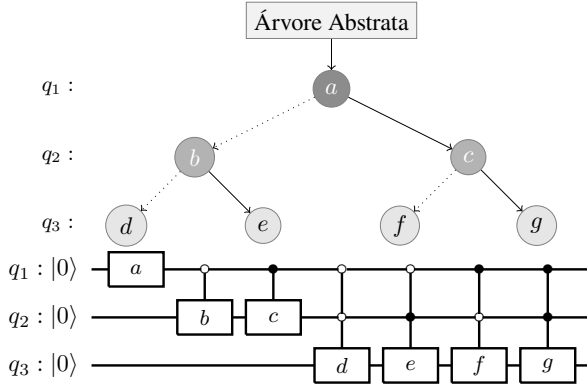


Figura 1. Árvore abstrata e circuito quântico, respectivamente, que implementam a inicialização de um estado arbitrário de 3 qubits.

Por fim, inverter o circuito obtido permite encontrar o circuito que de fato implementa o estado $|x\rangle$, ou seja, equivale ao operador SP_x . A Figura 1 representa esse circuito para um estado arbitrário de 3 qubits. Caso o estado inicial do circuito $|e\rangle$ não seja $|0\rangle_n$, é possível aplicar uma cascata de multiplexadores que leve o estado do circuito até $|0\rangle_n$ e depois aplicar SP_x .

2.4 Simplificação de Multiplexadores para QSP

2.4.1 Método de Simplificação

Quando o estado final de um circuito está totalmente desemaranhado, ou seja, o estado pode ser decomposto em um produto de estados de um qubit, é de se esperar que todos os operadores controlados que são aplicados no qubit alvo de um multiplexador sejam iguais, uma vez que o estado desse qubit não está condicionado ao restante dos qubits. Dessa forma, é evidente que é possível simplificar esse multiplexador simplesmente adicionando o operador no qubit alvo, removendo todos os controles e operadores repetidos. Uma aplicação imediata dessa simplificação é na inicialização de estados totalmente separáveis a partir da abordagem descrita na seção anterior.

Entretanto, podemos ter situações onde o estado final é separável em estados menores, mas que não são, necessariamente, de um qubit. Nessas situações, ocorre um padrão característico de repetição de operadores nos multiplexadores, sempre com tamanho de potências de dois. A razão para isso é que, se um dado controle, digamos que em um qubit q_i , que não tem importância na hora de determinar qual operador será aplicado no alvo, as ramificações da árvore abstrata que surgem tanto para $q_i = |0\rangle$ quanto para $q_i = |1\rangle$ serão iguais. Isso significa dizer que subárvores com raiz em q_{i+1} vindas de um mesmo nó em q_i terão exatamente o mesmo padrão em suas folhas. Seja, por exemplo, a árvore da figura 2, onde o multiplexador final é tal que $[U_7, U_8] = [U_9, U_{10}]$ e $[U_{11}, U_{12}] = [U_{13}, U_{14}]$. Para essa árvore, as subárvores com raiz em U_3 e U_4 são iguais, assim como as subárvores com raiz em U_5 e U_6 . Dessa forma, o multiplexador no último nível da árvore é dividido em dois segmentos onde ocorrem repetições, como é indicado na figura. Definimos d como sendo a distância entre operadores repetidos dentro dessas

regiões. Nesse caso, $d = 2$. Dado que uma árvore binária completa possui 2^h folhas, onde h é a altura da árvore, a distância entre o multiplexador onde essas repetições ocorrem e a raiz das subárvores é de $\log_2 d$, e o controle desnecessário se situa a uma distância $\log_2 d + 1$ a partir do nível desse multiplexador.

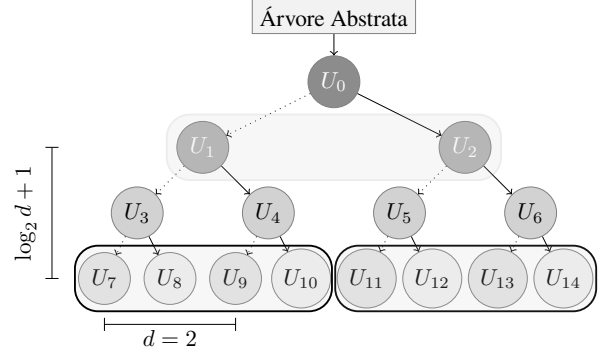


Figura 2. Árvore abstrata que representa como um padrão de repetição no último nível da árvore pode indicar a existência de um controle desnecessário nos multiplexadores da preparação de estados.

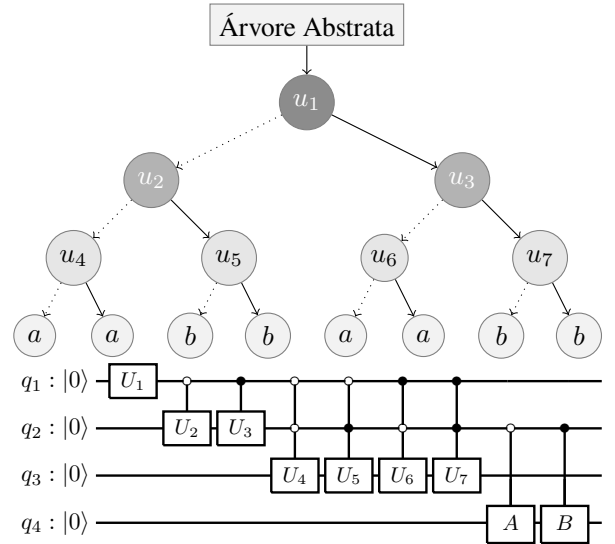


Figura 3. Árvore abstrata que representa a inicialização de um estado utilizando o multiplexador representado por V e circuito que a implementa.

A Figura 3 representa a árvore de uma inicialização de estados com um multiplexador com repetições dos operadores A e B no último nível da árvore. Perceba que a decisão de aplicar um operador ou o outro depende inteiramente de se seguir uma trajetória a partir de uma seta tracejada ou sólida no qubit q_2 . Note que, nesse caso, temos uma repetição com distância $d_1 = 1$ e uma outra repetição para $d_2 = 4$. Portanto, caso exista mais de um padrão de repetição, os diferentes controles desnecessários serão obtidos a partir da mesma regra. Assim, os controles em q_1 e q_3 são desnecessários para esse multiplexador. A figura também mostra o circuito final para a preparação de estados após a remoção dos controles desnecessários.

2.4.2 Algoritmo de Busca

A simplificação dos multiplexadores quânticos responsáveis pela inicialização de estados pode ser realizada a partir de uma busca na árvore relacionada à inicialização. A otimização

ocorre quando um conjunto de operadores iguais é detectado de acordo com as regras definidas na seção anterior. O procedimento descrito em Algoritmo 1 representa essa rotina algorítmica.

Algoritmo 1: Função de Simplificação

```

1 def simplifica (mux, qubit_alvo):
2   operadores_removidos = set ()
3   controles_removidos = list ()
4   se comprimento (mux) > 1 então
5     controles_removidos,
      operadores_removidos =
        busca_simplificacao (mux,
          qubit_alvo)
6   se operadores_removidos não está vazio então
7     mux_simplificado = mux sem os índices em
        operadores_removidos
8     retorne controles_removidos,
        mux_simplificado
9   mux_simplificado = mux
10  retorne controles_removidos,
    mux_simplificado;
```

Algoritmo 2: Função de Busca por Repetições

```

1 def busca_simplificacao (mux, qubit_alvo):
2   operadores_removidos = set ()
3   controles_removidos = list ()
4   para j = 0 até log2 (comprimento (mux))
5     faça
6       d = 2j
7       conjunto_removido = set ()
8       se mux[0] == mux[d] então
9         conjunto_removido =
          verifica_repeticao (d, mux)
10      se conjunto_removido não está vazio então
11        controle_removido = qubit_alvo + log2
          (d) + 1
12        Adicione controle_removido em
          controles_removidos
13        Adicione conjunto_removido em
          operadores_removidos
14  retorne controles_removidos,
    operadores_removidos
```

O algoritmo 1 implementa essa busca por simplificações. O multiplexador é representado como uma lista intitulada mux com os operadores a serem aplicados. A chamada à função de simplificação na linha 5 é responsável por retornar quais são os controles desnecessários e os índices dos operadores a serem removidos. Se de fato existem controles a serem removidos do multiplexador, um novo multiplexador é criado com apenas os controles necessários.

Na função no algoritmo 2, temos de fato a busca pelas repetições no multiplexador com o intuito de determinar possíveis controles a serem removidos. O laço da linha 4 percorre apenas as posições do multiplexador cuja posição

seja uma potência de dois. Se nessa posição for encontrado um operador igual ao que está na posição inicial, chamamos a função `verifica_repeticao`, que consiste em verificar se o multiplexador está dividido em $\text{comprimento}(\text{mux})/2d$ segmentos e retornar os operadores a serem removidos, onde d é a distância entre os operadores repetidos no interior desses segmentos. Se a verificação foi bem-sucedida e existirem operadores a serem removidos, o índice do controle a ser removido é calculado de acordo com a expressão vista anteriormente, ou seja, é igual ao controle no qubit que está a $\log_2 d + 1$ posições acima do qubit de alvo.

O custo total do algoritmo de busca por repetições em um dado multiplexador é linearítmico com relação ao número total de operações multiplexadas por esse multiplexador. Seja N o comprimento do multiplexador. Perceba que o algoritmo 2 percorre apenas as posições cujo número é uma potência de dois, o que resulta no máximo em $\log_2 N$ iterações. A função de verificação no pior caso irá comparar todos os operadores, dois a dois. Dessa forma, temos um custo máximo em $O(\frac{N}{2} \log_2 N)$. Finalmente, as tarefas do algoritmo 1 consistem em chamar a função de busca por repetições, com custo linearítmico, e remover os operadores desnecessários de mux, o que tem um custo no máximo linear. Portanto, o custo final do algoritmo de busca está em $O(N \log_2 N)$.

3 Experimentos e Discussão

Com o intuito de avaliar a eficiência da técnica sugerida pelo presente trabalho, dois experimentos distintos foram conduzidos. O primeiro dos experimentos consiste em simular a inicialização de um estado real com dois componentes desemaranhados, com diferentes números de qubits, utilizando algumas das várias abordagens de preparação de estados existentes na literatura. Para esse objetivo, foi utilizada uma mescla da abordagem de simplificação descrita neste trabalho com a técnica de inicialização de estados descrita em Bergholm *et al.* [2005]. Denotamos essa junção de técnicas através da sigla UCGE (Um acrônimo para a junção de *Uniformly Controlled One-Qubit Gates* com a palavra Embaranhamento). As métricas utilizadas para comparação foram o número final de portas CNOTs do circuito e o tempo de execução necessário de processamento clássico para a inicialização. Os resultados para esses testes estão indicados na figura 4.

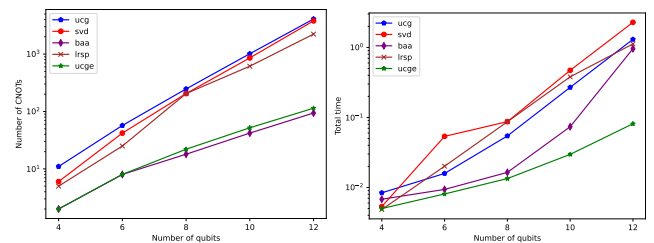


Figura 4. Número de portas CNOT e tempo total de execução para a inicialização de um estado bipartido para diferentes quantidades de qubits. É feita uma comparação entre as abordagens ‘ucg’ Bergholm *et al.* [2005], ‘svd’ Plesch and Brukner [2011], ‘lrsp’ Araujo *et al.* [2024] e ‘baa’ Araujo *et al.* [2024] com a abordagem ‘uge’ proposta aqui.

O segundo experimento, por sua vez, consiste na simulação da inicialização de um estado de 12 qubits com diferentes números de componentes desemaranhados utilizando as mesmas abordagens do experimento anterior. Os resultados para

esse experimento estão indicados na figura 5.

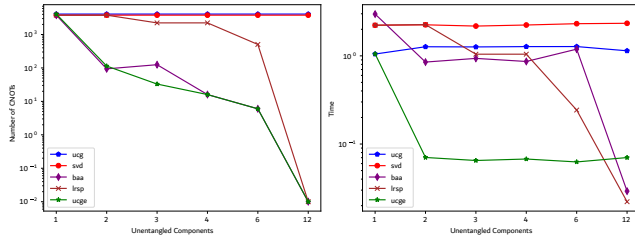


Figura 5. Número de portas CNOT e tempo total de execução para a inicialização de um estado de 12 qubits com diferentes números de partições. É feita uma comparação entre as abordagens ‘ucg’ Bergholm *et al.* [2005], ‘svd’ Plesch and Brukner [2011], ‘lrsp’ Araujo *et al.* [2024] e ‘baa’ Araujo *et al.* [2024] com a abordagem ‘ucge’ proposta aqui.

Analisando a figura 4 verificamos que o número total de CNOTs no circuito gerado com essa técnica é consideravelmente menor do que o respectivo número para as estratégias de QSP que não realizam nenhuma etapa de detecção de nível de emaranhamento. Ademais, a abordagem apresentada também mostra resultados altamente competitivos com os resultados obtidos com o método BAA Araujo *et al.* [2024], que também realiza a detecção de separabilidade, para a inicialização exata de um estado.

Entretanto, a diferença fundamental da nossa abordagem é a detecção eficiente, em termos do número de amplitudes do estado a ser inicializado, dessa separabilidade. O método proposto apresenta uma vantagem considerável no tempo de processamento clássico necessário para gerar o circuito que inicializa um estado onde ocorre um caso não trivial de separabilidade. Esse tempo total mensurado leva em conta o procedimento de transpilação, através do qual o circuito é decomposto em operadores unitários de um qubit e em portas CNOT. Esse procedimento é crucial, tendo em vista que os dispositivos quânticos da atualidade possuem apenas um conjunto limitado de portões quânticos disponíveis.

A abordagem BAA, como já citado anteriormente, também consegue se aproveitar da separabilidade do estado para reduzir a complexidade do circuito final. Esse método faz uso de uma abordagem alternativa que utiliza um esquema de decomposição de operadores através da decomposição de Schmidt Plesch and Brukner [2011]. Além disso, ela realiza o uso de uma aproximação de baixo posto que reduz o número total de CNOTs, introduzindo uma perda de fidelidade arbitrária. Como a decomposição de Schmidt funciona para sistemas bipartidos, o método BAA aplica essa decomposição recursivamente quando está lidando com um sistema multipartido, realizando a decomposição em duas partições por vez. Entretanto, identificar a configuração exata em que o estado a ser inicializado deve ser separado e realizar o cálculo da decomposição de Schmidt leva a um tempo de execução maior para essa técnica. Portanto, a nossa estratégia mostra uma vantagem no tempo total de compilação. Como ilustrado pela figura 4, o tempo total de execução do BAA aumenta com uma taxa de $N/\log_2(N)$ quando comparado com o UCGE.

4 Conclusão

A simplificação de multiplexadores quando aplicada na inicialização de estados quânticos leva a uma redução considerável no número total de portas CNOT necessárias para

essa finalidade e pode ser detectada em um tempo reduzido de processamento clássico. Tal feito é realizado através da análise das árvores sintáticas abstratas que representam a QSP. A demonstração da utilidade dessa abordagem foi realizada através da comparação dos resultados obtidos com essa abordagem e com várias das outras presentes na literatura, para a codificação de um conjunto de dados clássicos através das amplitudes de um estado quântico. Quando o estado a ser inicializado é separável, os ganhos do método são evidentes através do tempo de execução e da diminuição competitiva na complexidade do circuito.

Ademais, a técnica abordada não se limita à inicialização de estados, mas é aplicável a uma gama de rotinas do procedimento de compilação de software quântico que realizam o uso de multiplexadores. Possíveis trabalhos futuros são estudos nesse âmbito, em especial, o uso da abordagem de simplificação na síntese de operadores unitários. Além disso, trabalhos futuros podem envolver uma mescla da rotina de busca por otimizações e outras estratégias de inicialização de estados, como as que usam aproximação de baixo posto e desemaranhamento parcial.

Declarações complementares

Financiamento

Esta pesquisa foi financiada pelo CNPq, CAPES e FACEPE.

Contribuições dos autores

JAC: Análise formal, investigação, programas de computador, escrita. AJS: Conceitualização, análise formal, aquisição de financiamento, investigação, programas de computador, validação, escrita. Todos os autores leram e aprovaram o manuscrito final.

Conflitos de interesse

O conteúdo deste trabalho foi submetido em uma versão estendida para um periódico internacional.

Disponibilidade de dados e materiais

Os conjuntos de dados (e/ou softwares) gerados e/ou analisados durante o estudo atual estão disponíveis em <https://github.com/qcplib/qcplib>.

Referências

- Araujo, I. F., Blank, C., Araújo, I. C. S., and da Silva, A. J. (2024). Low-rank quantum state preparation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 43(1):161–170. DOI: 10.1109/tcad.2023.3297972.
- Araujo, I. F., Park, D. K., Petruccione, F., and da Silva, A. J. (2021). A divide-and-conquer algorithm for quantum state preparation. *Scientific reports*, 11(1):6329. DOI: 10.1038/s41598-021-85474-1.
- Bergholm, V., Vartiainen, J. J., Möttönen, M., and Salomaa, M. M. (2005). Quantum circuits with uniformly controlled one-qubit gates. *Phys. Rev. A*, 71:052330. DOI: 10.1103/PhysRevA.71.052330.
- Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. DOI: 10.48550/arXiv.quant-ph/9605043.
- Hughes, R. J., James, D. F. V., Knill, E. H., Laflamme, R., and Pleschek, A. G. (1996). Decoherence bounds on quantum

- computation with trapped ions. *Phys. Rev. Lett.*, 77:3240–3243. DOI: 10.1103/PhysRevLett.77.3240.
- Kim, Y., Eddins, A., Anand, S., Wei, K., Berg, E., Rosenblatt, S., Nayfeh, H., Wu, Y., Zaletel, M., Temme, K., and Kandala, A. (2023). Evidence for the utility of quantum computing before fault tolerance. *Nature*, 618:500–505. DOI: 10.1038/s41586-023-06096-3.
- Mottonen, M., Vartiainen, J. J., Bergholm, V., and Salomaa, M. M. (2004). Transformation of quantum states using uniformly controlled rotations. DOI: 10.48550/arXiv.quant-ph/0407010.
- Nakaji, K., Uno, S., Suzuki, Y., Raymond, R., Onodera, T., Tanaka, T., Tezuka, H., Mitsuda, N., and Yamamoto, N. (2022). Approximate amplitude encoding in shallow parameterized quantum circuits and its application to financial market indicators. *Physical Review Research*, 4(2):023136. DOI: 10.1103/PhysRevResearch.4.023136.
- Park, D. K., Petruccione, F., and Rhee, J.-K. K. (2019). Circuit-based quantum random access memory for classical data. *Scientific Reports*, 9(1). DOI: 10.1038/s41598-019-40439-3.
- Plesch, M. and Brukner, i. c. v. (2011). Quantum-state preparation with universal gate decompositions. *Phys. Rev. A*, 83:032302. DOI: 10.1103/PhysRevA.83.032302.
- Preskill, J. (2018). Quantum computing in the nisq era and beyond. *Quantum*, 2:79. DOI: 10.22331/q-2018-08-06-79.
- Shende, V., Bullock, S., and Markov, I. (2006). Synthesis of quantum-logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(6):1000–1010. DOI: 10.1109/tcad.2005.855930.
- 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. DOI: 10.1137/s0097539795293172.
- Wootters, W. K., Wootters, W. K., and Zurek, W. H. (1982). A single quantum cannot be cloned. *Nature*, 299:802–803. DOI: 10.1038/299802a0.
- Wu, Y. *et al.* (2021). Strong quantum computational advantage using a superconducting quantum processor. *Phys. Rev. Lett.*, 127:180501. DOI: 10.1103/PhysRevLett.127.180501.