

Article

Nested Grover's Algorithm for Tree Search

Andreas Wichert 

Department of Computer Science and Engineering, INESC-ID & Instituto Superior Técnico, University of Lisbon, 2740-122 Porto Salvo, Portugal; andreas.wichert@tecnico.ulisboa.pt

Abstract

We investigate optimizing quantum tree search algorithms by employing a nested Grover Algorithm. This approach seeks to enhance results compared to previous Grover-based methods by expanding the tree of partial assignments to a specific depth and conducting a quantum search within the subset of remaining assignments. The study explores the implications and constraints of this approach, providing a foundation for quantum artificial intelligence applications. Instead of utilizing conventional heuristic functions that are incompatible with quantum tree search, we introduce the partial candidate solution, which indicates a node at a specific depth of the tree. By employing such a function, we define the concatenated oracle, which enables us to decompose the quantum tree search using Grover's algorithm. With a branching factor of 2 and a depth of m , the costs of Grover's algorithm are $O(2^{m/2})$. The concatenated oracle allows us to reduce the cost to $O(m \cdot 2^{m/4})$ for m partial candidate solutions.

Keywords: quantum tree search; Grover's algorithm; heuristics; nested search

1. Introduction

We investigate the optimization of quantum tree search algorithms by employing a nested Grover Algorithm. We explore the implications of this approach and identify the resulting constraints. Our findings demonstrate the potential for enhanced results compared to previously developed Grover-based methods. Instead of adopting the hybrid divide-and-conquer methods that are predicated on quantum backtracking and quantum walk, as indicated in [1,2], we will proceed with the initial conception of a nested version of Grover's nested search, as described in [3]. In a nested version of Grover search, the complete tree of partial assignments is expanded to a specific depth, and then a quantum search is conducted within the subset of partial assignments that have not yet been eliminated.

It is claimed that, for certain reasonable distributions of problems, the average complexity of the original nested version of Grover search [3] will be less than that obtained from Grover search. This claim is supported by references [1,3]. Building upon the initial concept of a nested version of Grover search, we will adopt a combinatorial approach to address the limitations inherent in the original work, as described in [3]. For simplicity, we assume that the tree is uniform and its size is known beforehand. We will demonstrate our approach on simple examples using Qiskit, see [4]. The main contributions of this paper are as follows:

- We introduce the definition of a partial candidate solution and the contented oracles.
- We analyze the constraints of the original Grover's nested search as introduced in [3];
- We introduce the iterative approach with a quadratic speedup in comparison to the original Grover's algorithm;



Academic Editor: Pan Zhang

Received: 3 December 2025

Revised: 20 December 2025

Accepted: 23 December 2025

Published: 24 December 2025

Copyright: © 2025 by the author.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

- We investigate the possibilities of dividing the original space into two disentanglement subspaces and the resulting consequences;
- We introduce the concept of performing a permutation on a subspace and the corresponding constraint.

Our approach imposes specific constraints that elucidate the concepts of partial candidate solutions within the context of quantum tree search. This work provides the foundation for quantum artificial intelligence applications, including quantum problem-solving and quantum production systems.

2. Quantum Tree Search

In this section, we outline the application of Grover's algorithm to the tree search problem. Nodes and edges in a search tree represent states and transitions between states [5]. The initial state is the root, and from each state, either B states can be reached or it is a leaf. From a leaf no other state can be reached. b represents the branching factor of a node, indicating the number of possible choices. A leaf can either be the goal of the computation or an impasse when there are no valid transitions to a succeeding state. Each node except the root has a unique parent node, which is called the parent. Every node and its parent are connected by an edge. Each parent has b children. If $b = 2$ and the depth of the tree is m , each of the m questions has a yes/no reply and can be represented by a bit (as shown in Figure 1). The m answers are collectively represented by a binary number of length m .

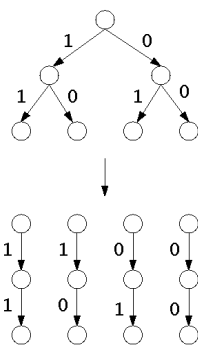


Figure 1. Search tree for $b = 2$ and depth $m = 2$. Each question can be represented by a bit. Each binary number (11, 10, 01, 00) represents a path descriptor m from the root to the leaf.

There are $n = 2^m = b^m$ possible binary numbers of length m . Each binary number represents a path from the root to a leaf. For each goal, a specific binary number indicates the solution. This binary number is called a path descriptor, since it describes a path from the root of the tree to the leaf, see Figure 2 with $b = 2$ and depth $m = 5$.

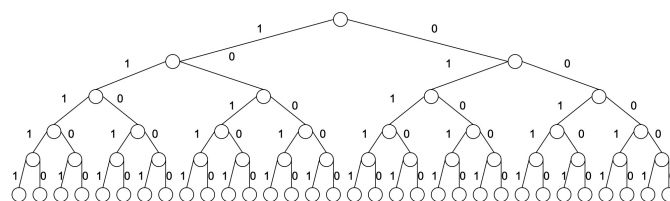


Figure 2. Search tree for $b = 2$ and depth $m = 5$. Each question can be represented by a bit.

For a constant branching factor $b > 2$, each question has b possible answers. The m answers can be represented by a base- b number with m digits. In a quantum computation, we can simultaneously represent all possible path descriptors. There is one path descriptor for each leaf off the tree. Using Grover's algorithm, we search through all possible paths and verify whether each path leads to the goal state. This type of procedure is known as a

quantum tree search [6]. For $n = b^m$ possible paths, the costs are (approximately) $\sqrt{n} = b^{\frac{m}{2}}$, corresponding to the reduced branching factor $b_q = \sqrt{b}$.

Grover's Algorithm

For a function $o(x)$

$$o_{\xi}(x) = \begin{cases} 1 & \text{if } x = \xi \\ 0 & \text{else} \end{cases} \quad (1)$$

we seek to identify the value of x representing a path descriptor for which $o(x) = 1$, where $x = \xi$, the path descriptor leading to the solution. This task is equivalent to a decision problem with a binary answer: 1 indicating a successful search and 0 indicating an unsuccessful search. The instance x is the input to the problem.

Grover's amplification algorithm implements exhaustive search in $O(\sqrt{n})$ steps in an n -dimensional Hilbert space [7–10]. This algorithm is derived from the Householder reflection of the state $|x\rangle$ of m qubits, where $n = 2^m$. Grover's amplification algorithm is optimal for exhaustive search, as demonstrated by the lower bound $\Omega(\sqrt{n})$ established by [11]. One can prove that a better algorithm cannot exist [12,13]. It follows that using a quantum computer, *NP-complete* problems remain *NP-complete*. Grover's amplification algorithm provides a quadratic speedup over classical computers, which would require n steps to solve the problem.

For k solutions and more than two qubits ($m > 2$, $n = 2^m$) the number r of iterations is the largest integer

$$r = \left\lfloor \frac{\pi}{4} \cdot \sqrt{\frac{2^m}{k}} \right\rfloor. \quad (2)$$

The value of r depends on the relation of n versus k . For $n = 4$ and $k = 1$ we need only one rotation, and we also need only one rotation for

$$\frac{n}{4} = k$$

to find **one** of the k solutions. The iterative amplification algorithm requires the value of k in order to determine the number of iterations. We can determine the value of k by the quantum counting algorithm [14–16].

Uniform distributions are essential for Grover's algorithm. If the distribution is non-uniform, the algorithm may not function properly or require adaptation which leads to the same complexity $O(\sqrt{n})$ [3,17–19]. Consequently, we cannot simply mark potential solutions by assigning higher amplitude values to speed up the search. For instance, an adapted algorithm for sparse distributions where the majority of amplitudes are zero must generate a uniform distribution for the unmarked sets, with the resulting complexity $O(\sqrt{n})$ [17–19].

In our analysis in relation to the O notation $O(\sqrt{\frac{n}{k}})$ and for simplification, we will assume the approximate complexity of Grover's algorithm as

$$\sqrt{\frac{2^m}{k}} = \sqrt{\frac{n}{k}}. \quad (3)$$

instead of using the accurate Equation (2).

3. Nested Grover's Search

Is it possible to integrate nested Grover's search techniques into the quantum tree search algorithm to enhance the search process? Can the nested Grover's search be based on heuristic functions? A heuristic function $\mu(y)$ estimates the cheapest cost from a node

to the goal. The local path descriptor y describes the path from the root to the node. Therefore, each node in the tree can be described by a local path descriptor. However, we cannot employ a heuristic function to generate a non-uniform distribution, as it does not confer any advantage in Grover's search. Furthermore, we cannot compare branches represented by distinct path descriptors. This is also the reason why the min-max algorithm cannot be applied to quantum tree search in games. The reason for this is that distinct branches correspond to distinct superpositions, and these superpositions interact solely through interference.

3.1. Partial Candidate Solution

Within the context of quantum tree search, instead of employing a heuristic function $\mu(y)$, we will utilize the function $h(y)$, which represents a partial candidate solution, and y denotes a path from the root to the corresponding node. In contrast to the heuristic function $\mu(y)$, which can mark multiple nodes, $h(y)$ should exclusively mark a single partial solution that includes the node indicated by the path y .

3.2. Decomposition

The idea of saving the costs is based on the inequality that represents the decomposition of the Hilbert space $\mathcal{H}$ of dimension $n = 2^m$ into subspaces $\mathcal{L}$ and $\mathcal{U}$ of dimension 2^g and 2^{m-g} , see Figure 3. For the value g we assume

$$2 \leq g < m.$$

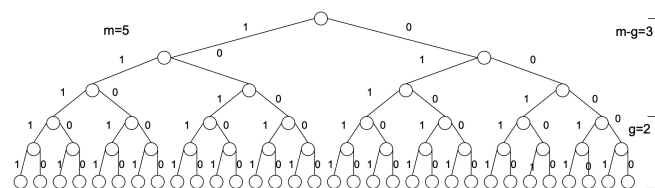


Figure 3. The decomposition of the Hilbert space $\mathcal{H}$ of dimension $n = 2^m$ into subspaces $\mathcal{L}$ and $\mathcal{U}$ of dimension 2^g and 2^{m-g} . In our example, $m = 5$ and $g = 2$.

The costs of Grover's algorithm for the quantum tree search algorithm are

$$\sqrt{2^m} = \sqrt{2^g} \cdot \sqrt{2^{m-g}}.$$

However, the costs associated with Grover's algorithm for the quantum tree of each subspace would be significantly reduced to

$$\sqrt{2^g} + \sqrt{2^{m-g}}$$

(see Figure 4) with

$$\sqrt{2^g} + \sqrt{2^{m-g}} < \sqrt{2^g} \cdot \sqrt{2^{m-g}}, \quad (4)$$

since

$$1 < \sqrt{2^g} \cdot \sqrt{2^{m-g}} - \sqrt{2^g} + \sqrt{2^{m-g}} + 1$$

$$1 < (\sqrt{2^g} - 1) \cdot (1 - \sqrt{2^{m-g}})$$

because

$$1 < (\sqrt{2^g} - 1), \quad 1 < (1 - \sqrt{2^{m-g}}).$$

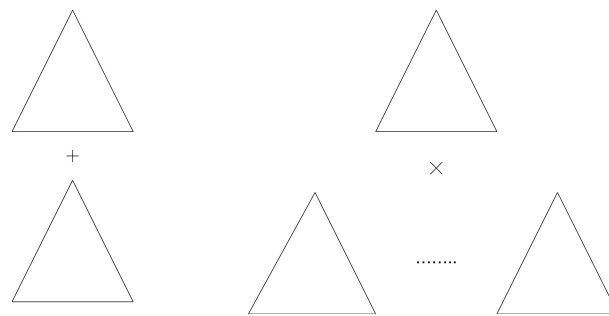


Figure 4. The concept of cost savings is predicated on the inequality that decomposes the Hilbert space $\mathcal{H}$ of dimension $n = 2^m$ into subspaces $\mathcal{H} = \mathcal{U} \otimes \mathcal{L}$ of dimensions 2^g and 2^{m-g} , respectively. On the left side, the sum of the costs of Grover's algorithm, with leaves for addition resulting in $\sqrt{2^g} + \sqrt{2^{m-g}}$, is represented. On the right side, the original costs of Grover's algorithm are $\sqrt{2^m} = 2^g \cdot 2^{m-g}$, corresponding to the multiplication operation.

3.3. Nested Grover's Search and Entanglement

Following, alternatively, the idea that Grover should speed up the search on Hilbert space $\mathcal{H}$ of dimension $n = 2^m$, a set $\aleph$ of v partial candidate solutions

$$h^{(k)}(y) \text{ with } k \in \{1, 2, \dots, v\}$$

is defined that acts on g lower qubits with

$$h^{(k)}(y) = \begin{cases} 1 & \text{if } y = h_k \\ 0 & \text{else} \end{cases} \quad (5)$$

on the subspace $\mathcal{L}$ of dimension 2^g , with y being a local path descriptor. Throughout the paper, we will employ the little-endian notation, which stores the least-significant byte at the smallest address. Qubits are represented from the most significant bit (MSB) on the left to the least significant bit (LSB) on the right (little-endian), similar to binary numbers or bit-string representation on classical computers. We illustrate the fundamental concept of nested Grover search through a simple SAT example, characterized by $v = 2$, $g = 3$, and $m = 5$.

$$h^{(1)}(y) = \neg x_3 \wedge x_2 \wedge x_1, \quad h^{(2)}(y) = x_3 \wedge \neg x_2 \wedge x_1$$

$$y = x_3, x_2, x_1; \quad x_k \in \{0, 1\},$$

$$o(x) = x_5 \wedge \neg x_4 \wedge x_3 \wedge \neg x_2 \wedge x_1, \quad x = x_5, x_4, \dots, x_1;$$

$$x_k \in \{0, 1\},$$

as indicated by the circuit in Figure 5a, with the results of measurement shown in Figure 5b. In the initial step, we determine the set $\aleph$ using Grover's algorithm on the subspace $\mathcal{L}$. In the subsequent step, we anticipate utilizing this set to expedite the search for the solution indicated by the oracle $o_{\xi}(x)$, which operates on m qubits. Grover's amplification is applied to the subspace $\mathcal{U}$ defined by the upper $m - g$ qubits. However, due to entanglement, we are unable to employ Grover's algorithm on the upper subspace $\mathcal{U}$ defined by $m - g$ upper qubits, as both subspaces $\mathcal{L}$ and $\mathcal{U}$ become entangled.

To comprehend the behavior we introduce concatenated oracles.

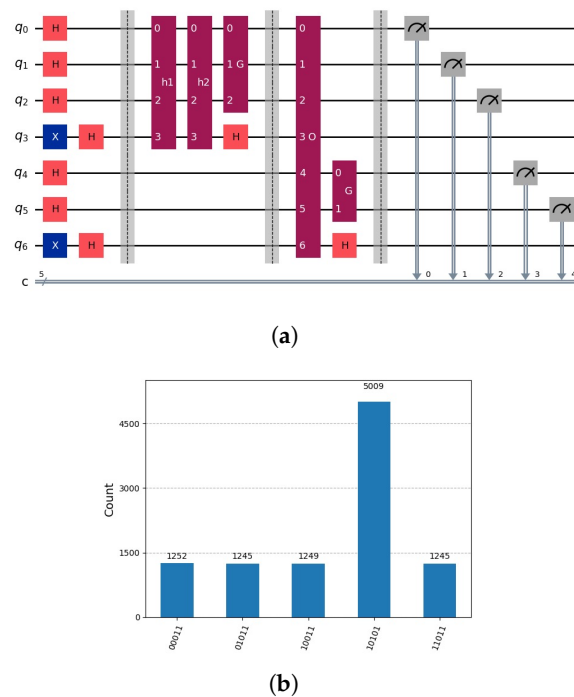


Figure 5. (a) Circuit using Grover's amplification representing oracles $h^{(1)}(y) = \neg x_3 \wedge x_2 \wedge x_1$, $h^{(2)}(y) = x_3 \wedge \neg x_2 \wedge x_1$ for lower qubits and the global oracle $o(x) = x_5 \wedge \neg x_4 \wedge x_3 \wedge \neg x_2 \wedge x_1$. Note, qubits 3 and 5 are auxiliary qubits used in Grover's algorithm to mark the solution. We will employ the little-endian notation as used in Qiskit, see [4]. Notation: *H* indicates Hadamard gate, *X* indicates NOT Gate, *h1* and *h2* oracle circuits that marks the solution in the auxiliary qubits. *O* represents the global oracle circuits and *G* Grover's algorithm circuits. Composed quantum circuits are indicated dark red, quantum gates with colours orange and blue. (b) Due to the entanglement of the lower qubits with the upper qubits, the solution indicated by 10101 is indicated only approximately by 50%.

3.4. Concatenated Oracles

Concatenation is an associative operation but not a commutative operation. We use for the operation concatenation the specific symbol $\parallel$. If we assume that x is represented by m bits with the notation

$$x = x_m, x_{m-1}, x_{m-2} \cdots, x_2, x_1; \quad x_k \in \{0, 1\},$$

then concatenation is often simply the placement of the binary numbers next to each other,

$$(x_m, x_{m-1}, \cdots, x_g) \parallel (x_{g-1}, \cdots, x_1) = x_m, x_{m-1}, \cdots, x_2, x_1$$

For example, for a binary number of 10 and 11 we get

$$10 \parallel 11 = 1011.$$

We extend the concatenation for the oracle function

$$o_{\xi}(x_m, x_{m-1}, x_{m-2} \cdots, x_1) =$$

$$u(x_m, x_{m-1}, \cdots, x_g) \parallel l = (x_{g-1}, \cdots, x_1). \quad (6)$$

Using compact notation,

$$z = (x_m, x_{m-1}, \cdots, x_g), \quad y = (x_{g-1}, \cdots, x_1),$$

$$x = (x_m, x_{m-1}, \dots, x_g) \parallel (x_{g-1}, \dots, x_1),$$

$$x = z \parallel y,$$

and the oracle function

$$o_{\xi}(x) = u(z) \parallel l(y)$$

with

$$o_{\xi}(x) = \begin{cases} 1 & \text{if } x = \xi \\ 0 & \text{else} \end{cases}$$

$$u(z) = \begin{cases} 1 & \text{if } z = u \\ 0 & \text{else} \end{cases} \quad l(y) = \begin{cases} 1 & \text{if } y = l \\ 0 & \text{else} \end{cases} \quad (7)$$

solution being

$$\xi = u \parallel l,$$

with $u = u(z)$ being the upper part of the oracle function that acts on the upper qubits, describing the upper subspace $\mathcal{U}$ of the dimension $2^{(m-g)}$ and $l = l(y)$ being the lower part of the oracle function that acts on the lower qubits, describing the lower subspace $\mathcal{L}$ of the dimension 2^g with

$$o(x) = o(z, y) = u(z) \parallel l(y).$$

Note that $l(y)$ and $u(z)$ do not indicate partial candidate solutions; they are exact copies of the oracle function, and it can be challenging to specify them precisely. Also, u indicates the path descriptor. An example is represented in Figure 6.

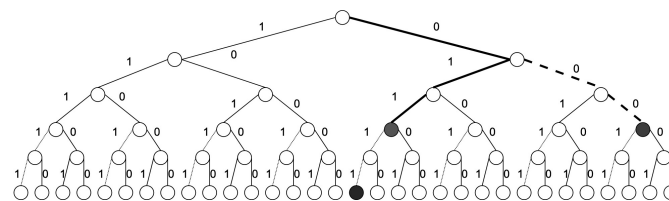


Figure 6. Search tree for $B = 2$ and depth $m = 5$. With $h_1 = 011$, $h_2 = 000$, $u = 11$, $l = 011$ and $\xi = 01111$. Bold lines denote the path to the solution. The dotted line represents the path to the partial candidate solution that does not lead to a solution. Upper-field circles signify partial candidate solutions, while the bold circle in the leaf signifies the solution, which is composed of $h_1 = l$ and u .

In the case we constraint y values to some values h_k with

$$y \in \{h_1, h_2, \dots, h_v\}$$

the constraint oracle can be expressed as $l = h_k$ with

$$o(x) = o(z, h_k) = u(z) \parallel l(h_k).$$

3.5. Not Entangled Subspaces

Imagine we know $u_{\mu}(z)$ that acts on the upper subspace $\mathcal{U}$ defined by $(m - g)$ lower qubits and given set $\aleph$ of t partial candidate solutions that act on the subspace $\mathcal{L}$ defined by upper g qubits,

$$h^{(k)}(y) \text{ with } k \in \{1, 2, \dots, v\}.$$

If we apply both oracles and use Grover's algorithm to each Hilbert space defined by the subspace $\mathcal{L}$ and $\mathcal{U}$, the results are not entangled since both subspaces are not entangled. In the first step, we determine $\aleph$ using Grover's algorithm on the Hilbert space defined by the

subspace $\mathcal{L}$. In the second step, we determine $u(z)$ using Grover's algorithm on the upper qubits of the upper subspace $\mathcal{U}$ with

$$\mathcal{H} = \mathcal{U} \otimes \mathcal{L}. \quad (8)$$

If we measure both subspaces, we see that each solution in the lower subspace $\mathcal{L}$ defines a new combination of the one solution in the upper subspace $\mathcal{U}$. One of the combination could correspond to the global solution of the space as described by the oracle $o_{\xi}(x)$, however we do not know if this is the case.

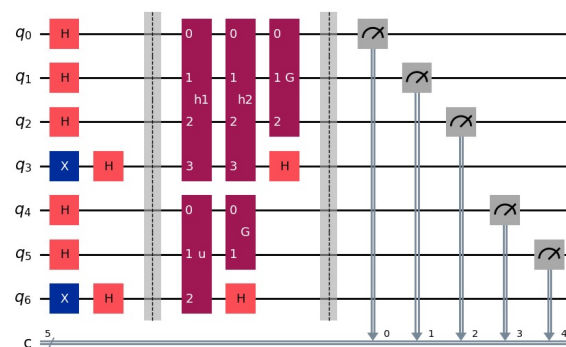
For example in the preceding example we introduce with $v = 2$ and $g = 3$

$$h^{(1)}(y) = \neg x_3 \wedge x_2 \wedge x_1, \quad h^{(2)}(z) = x_3 \wedge \neg x_2 \wedge x_1$$

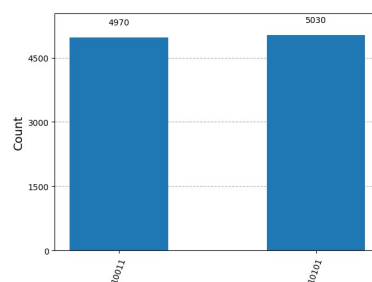
and the oracle for the $(5 - 3)$ upper qubits

$$u(z) = x_5 \wedge \neg x_4$$

as indicated by the circuit in Figure 7a, with the results of measurement shown in Figure ??b.



(a)



(b)

Figure 7. (a) Circuit employing Grover's amplification to represent oracles $h^{(1)}(z) = \neg x_3 \wedge x_2 \wedge x_1$, $h^{(2)}(y) = x_3 \wedge \neg x_2 \wedge x_1$ for lower qubits, and the oracle for the upper qubits $u(z) = x_5 \wedge \neg x_4$. When we measure, we are not guaranteed to obtain the solution $\xi = o(x)$, since $o(x)$ was not checked and the possible combinations may be not present. Notation: H indicates Hadamard gate, X indicates NOT Gate, $h1$ and $h2$ partial candidate solution circuits that marks the solution in the auxiliary qubit, u represents upper part of the oracle function circuit and G Grover's algorithm circuits. Composed quantum circuits are indicated dark red, quantum gates with colours orange and blue. (b) Due to the combinatorial problem that results in our case, we yield two solutions with equal probability, corresponding to the number of partial candidate solutions, since $(v = 2)$ with $\frac{1}{\sqrt{v}} \sum_{k=1}^v |u, h_k\rangle = \frac{1}{\sqrt{2}} \cdot (|10011\rangle + |10101\rangle)$.

The approach can be broadly characterized as

$$\frac{1}{\sqrt{m-g}} \cdot \sum_{z \in B^m} (-1)^{u(z)} \cdot |z\rangle, \quad \frac{1}{\sqrt{g}} \cdot \sum_{y \in B^g} (-1)^{h^{(k)}(y)} \cdot |y\rangle \quad (9)$$

$$\left(\frac{1}{\sqrt{m-g}} \cdot \sum_{z \in B^m} (-1)^{u(z)} \cdot |z\rangle \right) \otimes \left(\frac{1}{\sqrt{g}} \cdot \sum_{y \in B^g} (-1)^{h^{(k)}(y)} \cdot |y\rangle \right) \quad (10)$$

after applying Grover's algorithm to subspace $\mathcal{L}$

$$\left(\frac{1}{\sqrt{m-g}} \cdot \sum_{z \in B^m} (-1)^{u(z)} \cdot |z\rangle \right) \otimes \left(\frac{1}{\sqrt{v}} \sum_{k=1}^v |h_k\rangle \right) \quad (11)$$

and the subspace $\mathcal{U}$

$$|u\rangle \otimes \left(\frac{1}{\sqrt{v}} \sum_{k=1}^v |h_k\rangle \right) = \frac{1}{\sqrt{v}} \sum_{k=1}^v |u, h_k\rangle \quad (12)$$

since both subspaces $\mathcal{L}$ and $\mathcal{U}$ are not entangled.

3.6. Entangled Subspaces

In original nested Grover's search (see Section 3.3), after applying Grover's algorithm to the subspace $\mathcal{L}$ we get

$$\left(\frac{1}{\sqrt{m-g}} \cdot \sum_{z \in B^{(m-g)}} |z\rangle \right) \otimes \left(\frac{1}{\sqrt{v}} \sum_{k=1}^v |h_k\rangle \right) \quad (13)$$

both subspaces $\mathcal{L}$ and $\mathcal{U}$ are entangled because

$$\begin{aligned} |x\rangle &= |z, y\rangle, \\ o(x) &= o(z, y) = o(z, h_k) = u(z) \parallel h^{(k)}(y) = u(z) \parallel l(y), \\ &\left(\frac{1}{\sqrt{m-g}} \cdot \frac{1}{\sqrt{v}} \cdot \sum_{z \in B^{(m-g)}} \sum_{k=1}^v (-1)^{o(z, h_k)} \cdot |z, h_k\rangle \right). \end{aligned} \quad (14)$$

Applying Grover's algorithm to the subspace $\mathcal{U}$ results in the creation of entanglement.

$$\left(\frac{1}{\sqrt{m-g}} \cdot \frac{1}{\sqrt{v}} \cdot \sum_{z \neq u} \sum_{h_k \neq l} |z, h_k\rangle \right) + \frac{1}{\sqrt{v}} |\tilde{c}\rangle. \quad (15)$$

In our preceding example with $v = 2$ (see Figure 5) after applying Grover's algorithm to the subspace $\mathcal{L}$

$$\frac{1}{2} \cdot \frac{1}{\sqrt{2}} \cdot \sum_{z \in B^2} \sum_{k=1}^2 (-1)^{o(z, h_2)} \cdot |z, h_k\rangle \quad (16)$$

$$\left(\frac{1}{\sqrt{8}} \cdot \sum_{z \in B^2} (-1)^{o(z, h_2)} \cdot |z, h_1\rangle \right) + \left(\frac{1}{\sqrt{8}} \cdot \sum_{z \in B^2} (-1)^{o(z, h_2)} \cdot |z, h_2\rangle \right) \quad (17)$$

$$\left(\frac{1}{\sqrt{8}} \cdot \sum_{z \in B^2} |z, h_1\rangle \right) + \left(\frac{1}{\sqrt{8}} \cdot \sum_{z \in B^2} (-1)^{o(z, h_2)} \cdot |z, h_2\rangle \right) \quad (18)$$

After applying Grover's algorithm to the subspace $\mathcal{U}$ we get

$$\left(\frac{1}{\sqrt{8}} \cdot \sum_{z \in B^2} |z, h_1\rangle \right) + \frac{1}{\sqrt{2}} \cdot |\xi\rangle. \quad (19)$$

with the state

$$\begin{aligned} & \frac{1}{\sqrt{8}} \cdot (|00, h_1\rangle + |01, h_1\rangle + |10, h_1\rangle + |11, h_1\rangle) + \\ & \frac{1}{\sqrt{8}} \cdot (|00, h_2\rangle - |01, h_2\rangle + |10, h_2\rangle + |11, h_2\rangle), \end{aligned}$$

After applying Grover's algorithm to the subspace $\mathcal{U}$ we get

$$\frac{1}{\sqrt{8}} \cdot (|00, h_1\rangle + |01, h_1\rangle + |10, h_1\rangle + |11, h_1\rangle) + \frac{1}{\sqrt{2}} \cdot |\xi\rangle,$$

see Figure 5b. Grover's algorithm operates on the initial two qubits (upper qubits). It operates on two subspaces of $\mathcal{U}$, one for h_1 and the other for h_2 . In the subspace defined by h_1 , there is no marking. In the other subspace, there is a marking. As a result of Grover's amplification, the first subspace exhibits a uniform distribution, while the second subspace contains a marking indicating the solution. Each of the two subspaces defined by h_1 and defined by h_2 has a probability of being measured of 0.5. In the first subspace, the uniform distribution of each state is represented by $0.5/4 = 1/8 = 0.125$. In the second subspace, the solution is represented by 0.5, as indicated in Figure 5b. The number of partial candidate solutions, denoted as v , is inversely proportional to the probability of measuring the solution, which is represented as $1/v$. How can this problem be effectively addressed? Two potential solutions to this problem are the iterative approach and the idea of the disentanglement of the two subspaces $\mathcal{L}$ and $\mathcal{U}$.

4. The Iterative Approach

With the set $\aleph$ of v partial candidate solutions

$$h^{(k)}(y) = h_k \text{ with } k \in \{1, 2, \dots, v\}$$

and the global oracle $o_\xi(x)$, we built a circuit with one possible could-be solution of the set $\aleph$ and the global oracle. We determine in the first step one solution of one $h^{(k)}(y)$ using Grover's algorithm on the subspace $\mathcal{L}$, and in the second step we the global oracle $o_\xi(x)$ acts on m qubits. Subsequently, we apply Grover's amplification to the subspace $\mathcal{U}$, which is defined by the upper $m - g$ qubits. Figure 8 indicates an example representing the decomposed search tree for $v = 4$.

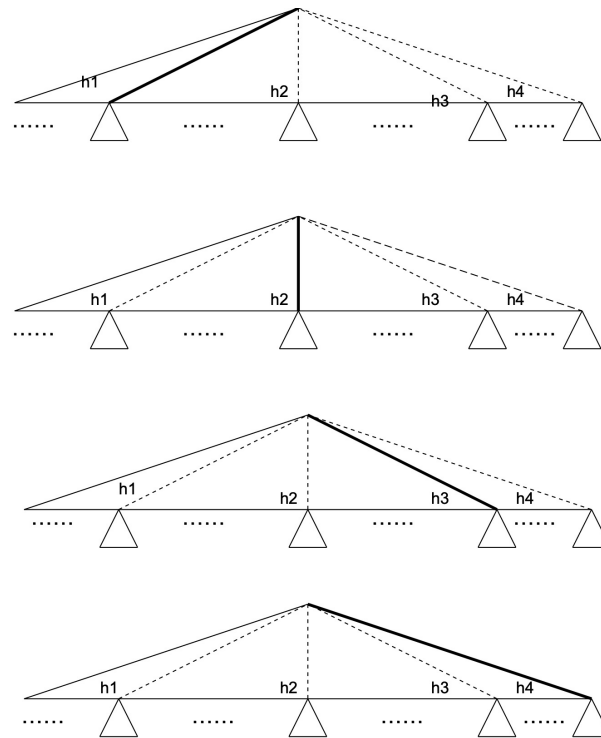


Figure 8. Example for $v = 4$. Determine the solution h_1 by Grover's algorithm on the subspace $\mathcal{L}$. Verify if the solution indicated by the oracle $o_{\xi}(x)$ exists. Repeat the procedure for h_2, h_3, h_4 . The bold line signifies the verification of the partial candidate solution.

If a solution exists, the subspaces $\mathcal{L}$ and $\mathcal{U}$ are not entangled with $\mathcal{H} = \mathcal{U} \otimes \mathcal{L}$. In this case, by measuring $\mathcal{H}$, we obtain the solution. This is because the number of partial candidate solutions represented in our circuit is “one,” $v^* = 1$, and the probability of measuring a solution, if it exists, is $1/v^* = 1$. We measure the state and test whether a solution is present. If it is, we terminate the process; otherwise, we select another potential solution from the set $\aleph$. For the simple preceding SAT example with $v = 2$ and $g = 3$ and $m = 5$ we get two circuits, see Figure 9.

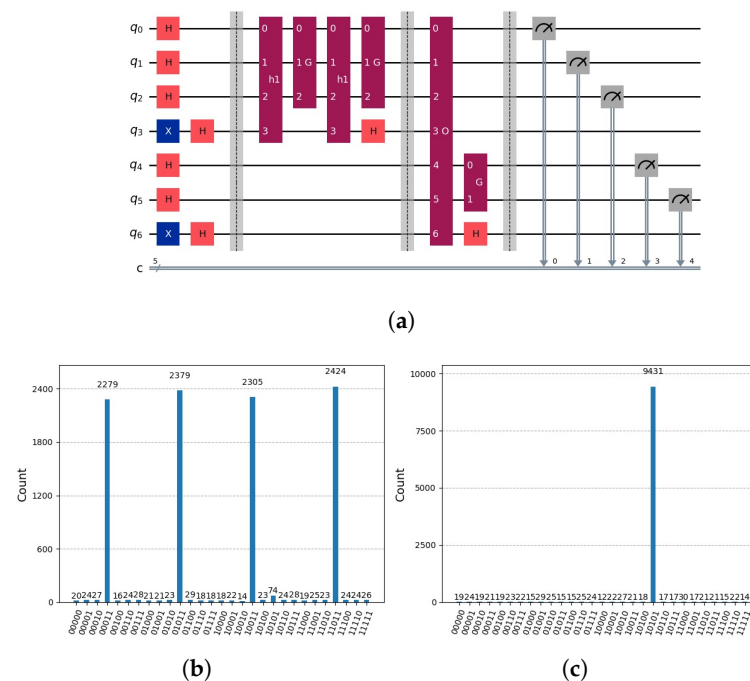


Figure 9. (a) Circuit using Grover's amplification representing oracle $h^{(1)}(y) = \neg x_3 \wedge x_2 \wedge x_1$ for lower qubits and the global oracle $o(x) = x_5 \wedge \neg x_4 \wedge x_3 \wedge \neg x_2 \wedge x_1$. Notation: H indicates Hadamard gate, X indicates NOT Gate, $h1$ and $h2$ partial candidate solution circuits that marks the solution in the auxiliary qubits. O represents the global oracle circuits and G Grover's algorithm circuits. Composed quantum circuits are indicated dark red, quantum gates with colours orange and blue. (b) Since the solution does not exist, when measuring we get the combination of h_1 with all four possible combinations of the two upper qubits x_5 and x_6 with the probability $1/8$. (c) The solution for a circuit $\zeta = u \parallel l$ with $l = h_2$. It is the same circuit as shown in (a), only $h^{(1)}(y)$ is replaced with $h^{(2)}(z) = x_3 \wedge \neg x_2 \wedge x_1$.

4.1. Algorithm

Given set $\aleph$ of v partial candidate solutions

$$h^{(k)}(y) = h_k \text{ with } k \in \{1, 2, \dots, v\}$$

and the global oracle $o_{\zeta}(x)$.

For $k = 1$ to v :

1. For the could-be solution $h^{(k)}(y) = h_k$ build a circuit;
2. Determine the solution h_k by Grover algorithm on the subspace $\mathcal{L}$;
3. Verify if the solution indicated by the oracle $o_{\zeta}(x)$ exists;
4. If solution exists success, exit the loop, otherwise, continue the loop.

Since both subspaces $\mathcal{L}$ and $\mathcal{U}$ are not entangled, if a solution exists, it is readily apparent. We measure the state and verify whether it is a valid solution. If it is a solution, we conclude the process. Otherwise, we select another partial candidate solution from the set $\aleph$.

4.2. Optimal Dimension

How can we choose an optimal dimension of the two subspaces $\mathcal{U}$ and $\mathcal{L}$? The optimal dimension of the two subspaces results in the minimal costs represented by the minimum of the function

$$\sqrt{2^{m \cdot (1-a)}} + \sqrt{2^{m \cdot a}} \text{ with } a \in [0, 1] \quad (20)$$

which is equivalent to

$$e^{m \cdot (1-a)} + e^{m \cdot a} \text{ with } a \in [0, 1]$$

$$\frac{\partial}{\partial a} \cdot (e^{m \cdot (1-a)} + e^{m \cdot a}) = -e^{m \cdot (1-a)} \cdot m + e^{m \cdot a} \cdot m; \quad a \in [0, 1] \quad (21)$$

with the solution

$$a = \frac{1}{2}.$$

4.3. Costs

Within the subspace $\mathcal{U}$, the costs associated with Grover's algorithm are $\sqrt{2^{m/2}}$, while within the subspace $\mathcal{L}$, the corresponding costs remain $\sqrt{2^{m/2}}$. We need to verify the proposed solution leads to a global solution. Subsequently, we must repeat the procedure in the worst case v times, leading to the cost

$$v \cdot (\sqrt{2^{m/2}} + \sqrt{2^{m/2}} + 1) = v \cdot (2 \cdot 2^{m/4} + 1) \quad (22)$$

with the constraint for v that the cost are below $2^{m/2}$ with

$$v < \frac{n^{1/2}}{2 \cdot n^{1/4} + 1} \approx n^{1/4} = 2^{m/4}. \quad (23)$$

since $n = 2^m$. We observe that maximal possible value for v is approximately equal to the number of nodes located at one-fourth of the depth of the search tree. We can express the cost of iterative search it in O notation, given $v \approx m$, as

$$O(v \cdot n^{1/4}) = O(\log n \cdot n^{1/4}). \quad (24)$$

5. Disentanglement of $\mathcal{L}$ and $\mathcal{U}$

Can we disentangle the spaces $\mathcal{L}$ and $\mathcal{U}$ with $\mathcal{H} = \mathcal{U} \otimes \mathcal{L}$ by knowing the function $u(z)$? We test the global oracle $o_\xi(x)$ over the whole space of dimension m that defines the Hilbert space $n = 2^m$. The test should not entangle both subspaces $\mathcal{L}$ and $\mathcal{U}$. We rely on the assumption that ξ exists and can be decomposed as

$$\xi = u \parallel l.$$

Knowing $o_\xi(x)$, assuming the $u(z)$ is correct and with set $\aleph$ the task is to verify if

$$o(x) = o(z, h_k) = u(z) \parallel l(h_k)$$

is true. If it is true, then we can decompose after Grover's algorithm on the subspace $\mathcal{L}$

$$\frac{1}{\sqrt{2^m}} \frac{1}{\sqrt{v}} \cdot \sum_{z \in B^m} \sum_{k=1}^v (-1)^{o(z, h_k)} \cdot |z, h_k\rangle = \quad (25)$$

$$\frac{1}{\sqrt{2^m}} \frac{1}{\sqrt{v}} \cdot \left(\sum_{z \in B^m} \sum_{h_k \neq l} (-1)^{o(z, h_k)} \cdot |z, h_k\rangle + (-1)^{o(z, h_k)} \cdot |\xi\rangle \right) = \quad (26)$$

$$\frac{1}{\sqrt{2^m}} \frac{1}{\sqrt{v}} \cdot \sum_{z \in B^m} \sum_{h_k \neq l} |z, h_k\rangle - \frac{1}{\sqrt{2^m}} \frac{1}{\sqrt{v}} \cdot |\xi\rangle \quad (27)$$

After Grover's search on the subspace $\mathcal{U}$, if true,

$$\frac{1}{\sqrt{v}} \cdot \left(|u\rangle \otimes \sum_{h_k \neq l} |h_k\rangle \right) + \frac{1}{\sqrt{v}} \cdot |\xi\rangle = \quad (28)$$

$$|u\rangle \otimes \left(\frac{1}{\sqrt{v}} \sum_{i=k}^v |h_k\rangle \right)$$

If not true, for $v > 2$

$$\frac{1}{\sqrt{v-2}} \cdot |u\rangle \otimes \frac{1}{\sqrt{(v-2) \cdot 2^m}} \left(\sum_{z \in B^m} |z\rangle \right) \otimes \left(\frac{1}{\sqrt{v}} \sum_{i=k}^v |h_k\rangle \right)$$

and for $v = 2$

$$\frac{1}{\sqrt{2^m}} \left(\sum_{z \in B^m} |z\rangle \right) \otimes \left(\frac{1}{\sqrt{v}} \sum_{i=k}^v |h_k\rangle \right).$$

In the event that $|u\rangle$ does not exist for any z

$$\left(\frac{1}{\sqrt{2^m}} \frac{1}{\sqrt{v}} \cdot \sum_{z \in B^m} \sum_{k=1}^v |z, h_k\rangle \right) = \left(\frac{1}{\sqrt{2^m}} \sum_{z \in B^m} |z\rangle \right) \otimes \left(\frac{1}{\sqrt{v}} \sum_{k=1}^v |h_k\rangle \right). \quad (29)$$

Since we do not know which $h_k = l$ we have to test all combinations, but we can reuse the set $\aleph$ once computed.

$$\begin{aligned} & \left(\frac{1}{\sqrt{2^m}} \frac{1}{\sqrt{v}} \cdot \sum_{z \in B^m} \sum_{h_k \neq h_1} (-1)^{o(z, h_k)} \cdot |z, h_k\rangle + \frac{1}{\sqrt{2^m}} \frac{1}{\sqrt{v}} \cdot (-1)^{o(z, h_k)} \cdot |\xi\rangle \right) \\ & \otimes \left(\frac{1}{\sqrt{2^m}} \frac{1}{\sqrt{v}} \cdot \sum_{z \in B^m} \sum_{h_k \neq h_2} (-1)^{o(z, h_k)} \cdot |z, h_k\rangle + \frac{1}{\sqrt{2^m}} \frac{1}{\sqrt{v}} \cdot (-1)^{o(z, h_k)} \cdot |\xi\rangle \right) \\ & \dots \dots \dots \\ & \otimes \left(\frac{1}{\sqrt{2^m}} \frac{1}{\sqrt{v}} \cdot \sum_{z \in B^m} \sum_{h_k \neq h_v} (-1)^{o(z, h_k)} \cdot |z, h_k\rangle + \frac{1}{\sqrt{2^m}} \frac{1}{\sqrt{v}} \cdot (-1)^{o(z, h_k)} \cdot |\xi\rangle \right) = \quad (30) \\ & \left(\frac{1}{\sqrt{2^m}} \frac{1}{\sqrt{v}} \cdot \sum_{z \in B^m} \sum_{h_k \neq h_1} |z, h_k\rangle - \frac{1}{\sqrt{2^m}} \frac{1}{\sqrt{v}} \cdot |\xi\rangle \right) \\ & \otimes \left(\frac{1}{\sqrt{2^m}} \frac{1}{\sqrt{v}} \cdot \sum_{z \in B^m} \sum_{h_k \neq h_2} |z, h_k\rangle - \frac{1}{\sqrt{2^m}} \frac{1}{\sqrt{v}} \cdot |\xi\rangle \right) \\ & \dots \dots \dots \\ & \otimes \left(\frac{1}{\sqrt{2^m}} \frac{1}{\sqrt{v}} \cdot \sum_{z \in B^m} \sum_{h_k \neq h_v} |z, h_k\rangle - \frac{1}{\sqrt{2^m}} \frac{1}{\sqrt{v}} \cdot |\xi\rangle \right). \quad (31) \end{aligned}$$

After Grover's algorithm on the subspace $\mathcal{U}$,

$$\begin{aligned} & \left(\left(\frac{1}{\sqrt{v-2}} \cdot |u\rangle + \frac{1}{\sqrt{(v-2) \cdot 2^m}} \sum_{z \in B^m} |z\rangle \right) \otimes \left(\frac{1}{\sqrt{v}} \sum_{i=k}^v |h_k\rangle \right) \right) \\ & \otimes \left(|u\rangle \otimes \left(\frac{1}{\sqrt{v}} \sum_{i=k}^v |h_k\rangle \right) \right) \\ & \dots \dots \dots \\ & \otimes \left(\left(\frac{1}{\sqrt{v-2}} \cdot |u\rangle + \frac{1}{\sqrt{(v-2) \cdot 2^m}} \sum_{z \in B^m} |z\rangle \right) \otimes \left(\frac{1}{\sqrt{v}} \sum_{i=k}^v |h_k\rangle \right) \right) \end{aligned}$$

and if u is not present we get a general superposition. We arrive at the decomposition

$$\mathcal{H} = \mathcal{U}_{h_k \neq h_1} \otimes \mathcal{U}_{h_k \neq h_2} \otimes \dots \otimes \mathcal{U}_{h_k \neq h_v} \otimes \mathcal{L} \quad (32)$$

We have determined the correct $h^k(y)$ by measuring $|u\rangle$ with a probability of approximately 1 in $\mathcal{U}$. Subsequently, we can estimate the path descriptor $h_k = l$ using Grover's algorithm on the subspace $\mathcal{L}$. The probability measurement of $\frac{1}{v-2}$ indicates that $h^k(y)$ does not equal l and does not constitute a valid solution. However, for $v > 4$, it necessitates a substantial number of measurements to ascertain that the probability deviates from 1.

5.1. Example with $v = 2$

Our previous example with $v = 2$

$$\begin{aligned} & \left(\frac{1}{\sqrt{2^3}} \frac{1}{\sqrt{2}} \cdot \sum_{z \in B^3} (-1)^{o(z, h_k)} \cdot |z, h_1\rangle + \frac{1}{\sqrt{2^3}} \frac{1}{\sqrt{2}} \cdot (-1)^{o(z, h_k)} \cdot |\xi\rangle \right) \\ & \otimes \left(\frac{1}{\sqrt{2^3}} \frac{1}{\sqrt{2}} \cdot \sum_{z \in B^3} (-1)^{o(z, h_k)} \cdot |z, h_2\rangle + \frac{1}{\sqrt{2^3}} \frac{1}{\sqrt{2}} \cdot (-1)^{o(z, h_k)} \cdot |\xi\rangle \right) = \\ & \left(\frac{1}{\sqrt{2^3}} \frac{1}{\sqrt{2}} \cdot \sum_{z \in B^3} |z, h_1\rangle - \frac{1}{\sqrt{2^3}} \frac{1}{\sqrt{2}} \cdot |\xi\rangle \right) \\ & \otimes \left(\frac{1}{\sqrt{2^3}} \frac{1}{\sqrt{2}} \cdot \sum_{z \in B^3} |z, h_2\rangle - \frac{1}{\sqrt{2^3}} \frac{1}{\sqrt{2}} \cdot |\xi\rangle \right). \end{aligned}$$

After Grover's algorithm on the subspace $\mathcal{U}$ we get the decomposition

$$\mathcal{H} = \mathcal{U}_{h_k \neq h_1} \otimes \mathcal{U}_{h_k \neq h_2} \otimes \mathcal{L} = \mathcal{U}_{h_2} \otimes \mathcal{U}_{h_1} \otimes \mathcal{L}, \quad (33)$$

with

$$\begin{aligned} & \left(\frac{1}{2^3} \sum_{z \in B^m} |z\rangle \right) \otimes \left(\frac{1}{2} \cdot (|\xi\rangle + |u\rangle|h_2\rangle) \right) \otimes \left(\frac{1}{2} \cdot (|h_1\rangle + |h_2\rangle) \right) = \\ & \left(\frac{1}{2^3} \sum_{z \in B^m} |z\rangle \right) \otimes \left(\frac{1}{2} \cdot (|u\rangle|h_1\rangle + |u\rangle|h_2\rangle) \right) \otimes \left(\frac{1}{2} \cdot (|h_1\rangle + |h_2\rangle) \right) = \\ & \left(\frac{1}{2^3} \sum_{z \in B^m} |z\rangle \right) \otimes |u\rangle \otimes \frac{1}{2} \cdot (|h_1\rangle + |h_2\rangle). \end{aligned}$$

Because of the equality $|\xi\rangle = |u\rangle|h_1\rangle$ we know that $h^1(y)$ is correct and can estimate the path descriptor $h_1 = l$ by Grover's algorithm on the subspace $\mathcal{L}$.

5.2. Costs

Within the subspace $\mathcal{L}$, the cost associated with Grover's algorithm is $\sqrt{(2^{m/2}/v)}$. For v combinations in the subspace $\mathcal{U}$, the corresponding cost is $v \cdot \sqrt{2^{m/2}}$. Additionally, given h_k , we need to determine the path descriptor $h_k = l$ by Grover's algorithm in the subspace $\mathcal{L}$. The complete costs are

$$\sqrt{\frac{2^{m/2}}{v}} + \sqrt{2^{m/2}} + v \cdot \sqrt{2^{m/2}} = \sqrt{2^{m/2}} \cdot \left(\frac{1}{\sqrt{v}} + 1 + v \right) \quad (34)$$

We can express the cost of iterative search in O notation, given that $v \approx m$ as before, with

$$O(v \cdot n^{1/4}) = O(\log n \cdot n^{1/4}).$$

For $1 < v$, the actual costs are less than the cost of the iterative algorithm with

$$\sqrt{\frac{2^{m/2}}{v}} + \sqrt{2^{m/2}} + v \cdot \sqrt{2^{m/2}} < v \cdot (\sqrt{2^{m/2}} + \sqrt{2^{m/2}} + 1) \quad (35)$$

$$\sqrt{2^{m/2}} \cdot \left(\frac{1}{\sqrt{v}} + 1 \right) < v \cdot \sqrt{2^{m/2}} < v \cdot (\sqrt{2^{m/2}} + 1) \quad (36)$$

$$\frac{1}{\sqrt{v}} + 1 < v, \quad (37)$$

and they are cheaper in relation to *times*.

$$times = \frac{v \cdot (2 \cdot \sqrt{2^{m/2}} + 1)}{\sqrt{2^{m/2}} \cdot \left(\frac{1}{\sqrt{v}} + 1 + v \right)}, \quad (38)$$

For $v > 4$, a substantial number of measurements is necessary. A practical approach would be to select $v^* = 4$, resulting in approximately 1.45 faster costs

$$times = \frac{8 \cdot \sqrt{2^{m/2}} + 4}{\sqrt{2^{m/2}} \cdot 5.5} \approx 1.45$$

compared to the iterative approach with $v/4 = v/v^*$ possible iterations.

Implementation

For the $\aleph$ of v partial candidate solution functions that acts on g lower qubits on the subspace $\mathcal{L}$

$$h^{(k)}(y) \text{ with } k \in \{1, 2, \dots, v\}$$

we introduce a $flag_k$ variable

$$h^{(k)}(y) = \begin{cases} 1 = flag_k & \text{if } y = h_k \\ 0 = flag_k & \text{else} \end{cases} \quad (39)$$

and extend the upper oracle in relation to the $flag_k$ variable

$$\tilde{u}^{(k)}(z) = \begin{cases} 1 & \text{if } (z = u) \wedge (flag_k = 1) \\ 0 & \text{else} \end{cases} \quad (40)$$

For example, for the case $l = h_s$, $\tilde{u}^{(s)}(z) = o_{\tilde{c}}(x)$. To determine a possible solution we have to measure v subspaces $\mathcal{U}$.

For example, in the preceding SAT example with $v = 2$ and $g = 3$ we introduce the $flag_k$ variable

$$h^{(1)}(y) = \neg x_3 \wedge x_2 \wedge x_1, \quad h^{(2)}(z) = x_3 \wedge \neg x_2 \wedge x_1$$

$$u(z)^{(k)} = (\neg x_6 \wedge \neg x_5 \wedge x_4) \wedge flag_k$$

$$o(x) = \neg x_6 \wedge \neg x_5 \wedge x_4 \wedge \neg x_3 \wedge x_2 \wedge x_1$$

where $o_{\tilde{c}}(x)$ corresponds to $u(y)^1$. The decomposition

$$\mathcal{H} = \mathcal{U}_{h_2} \otimes \mathcal{U}_{h_1} \otimes \mathcal{L},$$

is represented by the quantum circuit in Figure 10a.

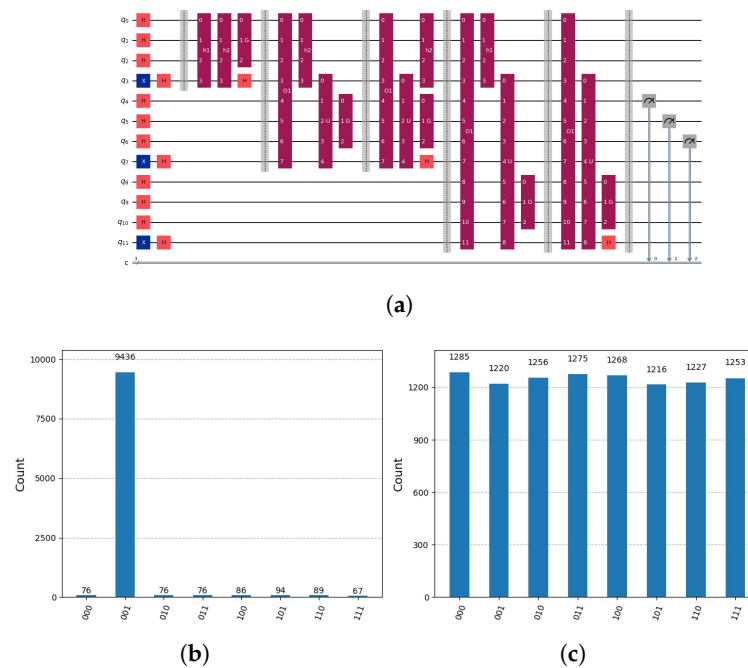


Figure 10. (a) Circuit employing Grover's amplification to represent oracles $h^{(1)}(z) = \neg x_3 \wedge x_2 \wedge x_1$, $h^{(2)}(z) = x_3 \wedge \neg x_2 \wedge x_1$ representing the decomposition $\mathcal{U}_{h_2} \otimes \mathcal{U}_{h_1} \otimes \mathcal{L}$. The subspace $\mathcal{L}$ is represented by the qubits 0–2, subspace $\mathcal{U}_{h_1}$ is represented by the qubits 4–6 and subspace $\mathcal{U}_{h_2}$ is represented by the qubits 8–10. Notation: H indicates Hadamard gate, X indicates NOT Gate, h_1 and h_2 partial candidate solution circuits that marks the solution in the auxiliary qubits. O represents the global oracle circuits and G Grover's algorithm circuits. Composed quantum circuits are indicated dark red, quantum gates with colours orange and blue. Notation: H indicates Hadamard gate, X indicates NOT Gate, h_1 and h_2 partial candidate solution circuits that marks the solution in the auxiliary qubits. O represents the global oracle circuits, U upper part of the oracle function and G Grover's algorithm circuits. Composed quantum circuits are indicated dark red, quantum gates with colours orange and blue. (b) We measure the subspace $\mathcal{U}_{h_1}$ representing the solution. (c) We measure the subspace $\mathcal{U}_{h_2}$ representing a non-solution.

6. Permutation in the Subspace $\mathcal{L}$

We can map the set of solutions determined by the function of the set $\mathcal{N}$ into in the subspace $\mathcal{L}$ into another ordering that uses only the corresponding subset that could lead to a decomposition. For example, the superposition

$$\frac{1}{\sqrt{2}} \cdot (|011\rangle + |101\rangle)$$

is mapped by a permutation matrix P

$$P \cdot \left(\frac{1}{\sqrt{2}} \cdot (|011\rangle + |101\rangle) \right) = \frac{1}{\sqrt{2}} \cdot (|000\rangle + |001\rangle) = |00\rangle \otimes \frac{1}{\sqrt{2}} \cdot (|0\rangle + |1\rangle) \quad (41)$$

into the decomposition $|00\rangle \otimes \frac{1}{\sqrt{2}} \cdot (|0\rangle + |1\rangle)$, see Figure 11.

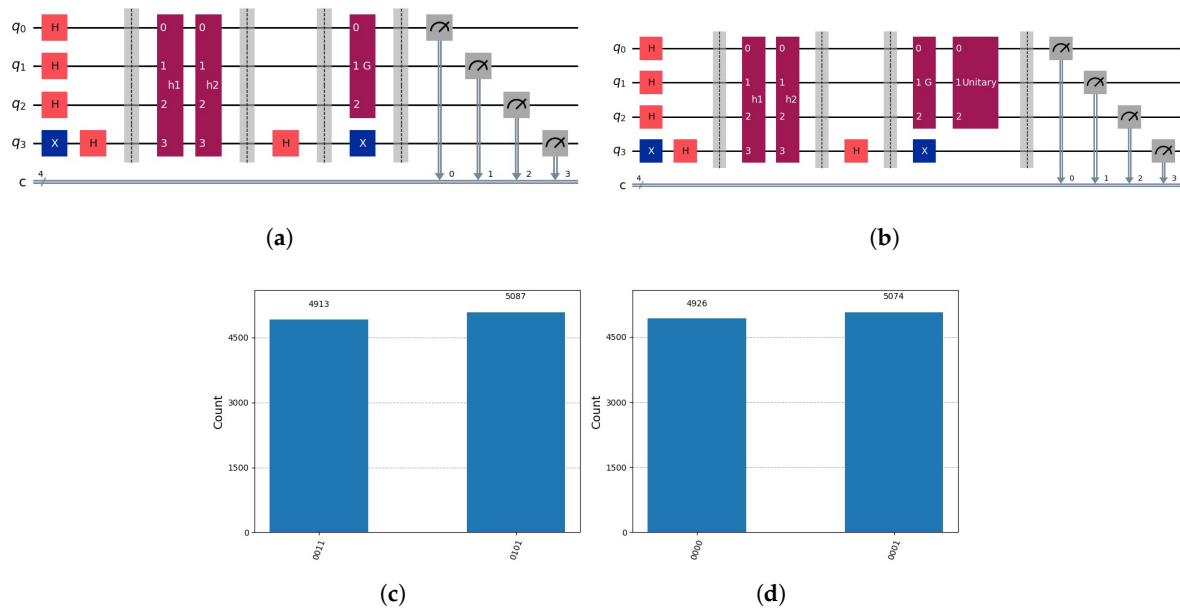


Figure 11. (a) After applying Grover's algorithm, with two solutions indicated by $h^1(y)$ and $h^2(y)$. (b) Mapping by permutation operator P indicated by the gate "Unitary" of the register $|y\rangle = \frac{1}{\sqrt{2}} \cdot (|011\rangle + |101\rangle)$ into $|x\rangle = |00\rangle \otimes \frac{1}{\sqrt{2}} \cdot (|0\rangle + |1\rangle)$. Notation: H indicates Hadamard gate, X indicates NOT Gate, $h1$ and $h2$ partial candidate solution circuits that marks the solution in the auxiliary qubits. G indicates Grover's algorithm circuits and *Unitary* the permutation operator P circuit. Composed quantum circuits are indicated dark red, quantum gates with colours orange and blue. (c) Measuring the circuit (a). (d) Measuring the circuit (b) representing $P \cdot \left(\frac{1}{\sqrt{2}} \cdot (|011\rangle + |101\rangle) \right) = |00\rangle \otimes \frac{1}{\sqrt{2}} \cdot (|0\rangle + |1\rangle)$.

We can represent the permutation matrix P as

$$\begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}.$$

The operation corresponds to the mapping by permutation operator P of a register $|y_k\rangle$ of m qubits into a register $|x_k\rangle$ of m qubits, of which the first g qubits are $|0\rangle^{\otimes g}$, with

$$\frac{1}{\sqrt{v}} \sum_{k=1}^v P \cdot |y_k\rangle = |0\rangle^{\otimes g} \otimes \left(\frac{1}{\sqrt{v}} \sum_{k=1}^v |x_k\rangle \right). \quad (42)$$

Since Qiskit uses little-endian notation we indicate the inverse order by a permutation matrix P^*

$$P^* \cdot \left(\frac{1}{\sqrt{2}} \cdot (|011\rangle + |101\rangle) \right) = \frac{1}{\sqrt{2}} \cdot (|000\rangle + |100\rangle) = \frac{1}{\sqrt{2}} \cdot (|0\rangle + |1\rangle) \otimes |00\rangle \quad (43)$$

into the decomposition $\frac{1}{\sqrt{2}} \cdot (|0\rangle + |1\rangle) \otimes |00\rangle$. We can represent the permutation matrix P^* , and due to little-endian notation we get

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}.$$

The operation corresponds to the mapping by permutation operator P^* of a register $|y_k\rangle$ of m qubits into a register $|x_k\rangle$ of m qubits, of which the last g qubits are $|0\rangle^{\otimes g}$, with

$$\frac{1}{\sqrt{v}} \sum_{k=1}^v P^* \cdot |y_k\rangle = \left(\frac{1}{\sqrt{v}} \sum_{k=1}^v |x_k\rangle \right) \otimes |0\rangle^{\otimes g}. \quad (44)$$

We can apply such a permutation operator P^* for a nested Grover's algorithm for tree search, as indicated in Figure 12. Grover's algorithm is applied on the qubits 2, 4 and 5.

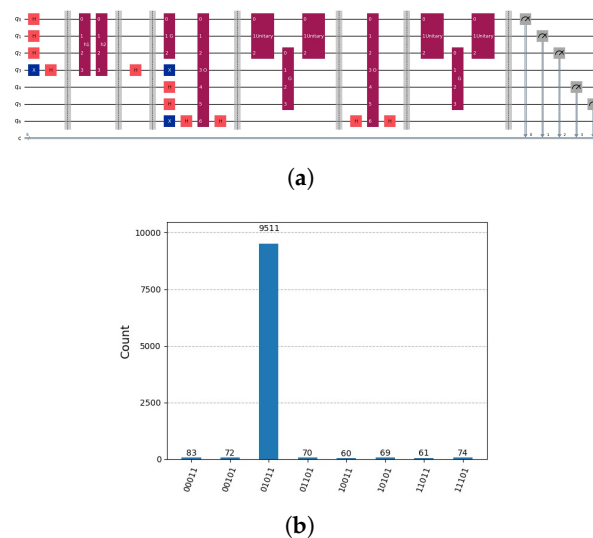


Figure 12. (a) The corresponding circuit; the permutation operator P^* is indicated by the gate “Unitary”. After applying Grover's algorithm with two solutions indicated by $h^1(y)$ and $h^2(y)$, using the oracle on the original space $\frac{1}{\sqrt{2}} \cdot (-|011\rangle + |101\rangle)$ to mark the solution and mapping $P^* \cdot \left(\frac{1}{\sqrt{2}} \cdot (-|011\rangle + |101\rangle) \right) = \frac{1}{\sqrt{2}} \cdot (-|0\rangle + |1\rangle) \otimes |00\rangle$. Qubit 2 indicates the marked solution describing the subspace $\mathcal{L}$ by permuted values h_1 and h_2 . Qubits 4 and 5 correspond to the subspace $\mathcal{U}$. Grover's algorithm is applied on the qubits 2, 4 and 5. We the transpose P^* to determine the solution by the oracle $o_{\xi}(x)$ in the original space. This is quite elegant, but until now, there appears to be no straightforward method to ascertain the mapping represented by P^* . Notation: H indicates Hadamard gate, X indicates NOT Gate, h_1 and h_2 partial candidate solution circuits that marks the solution in the auxiliary qubits. O represents the global oracle circuits, G indicates Grover's algorithm circuits and *Unitary* the permutation operator P circuit. Composed quantum circuits are indicated dark red, quantum gates with colours orange and blue. (b) Determining the correct solution in the original space.

6.1. Costs

Ignoring the costs of the permutation operator P with the preparation costs v by basis encoding in subspace $\mathcal{L}$, as described by [20–22], the costs corresponding to the Grover's algorithm applied to $m/2 + \log_2 v$ qubits are

$$v + \sqrt{2^{m/2 + \log_2 v}} = v + \sqrt{2^{m/2}} \cdot \sqrt{v}. \quad (45)$$

With the preparation costs $\sqrt{\frac{2^{m/2}}{v}}$ using Grover's algorithm in subspace $\mathcal{L}$ the costs are

$$\sqrt{\frac{2^{m/2}}{v}} + \sqrt{2^{m/2 + \log_2 v}} = \sqrt{2^{m/2}} \cdot \left(\frac{1}{\sqrt{v}} + \sqrt{v} \right) \quad (46)$$

in O notation, given $v \approx m$, with

$$O(v^{1/2} \cdot n^{1/4}) = O((\log n)^{1/2} \cdot n^{1/4}).$$

6.2. Permutation Operator P

There appears to be no straightforward method to ascertain the mapping represented by P . The permutation operator P is defined by the v paths h_k to partial could-be solutions. Due to entanglement we cannot determine the operator P using $h^k(z)$ and the resulting h_k paths. Due to entanglement we cannot determine the permutation operator P without knowing the h_k paths that indicate which nodes indicate partial could-be solutions. Knowing the h_k paths we can determine the permutation operator P by controlled not gates, see Figure 13a. Using the regularities, permutation operator P can be considerably compressed, see Figure 13c, leading to reduced costs.

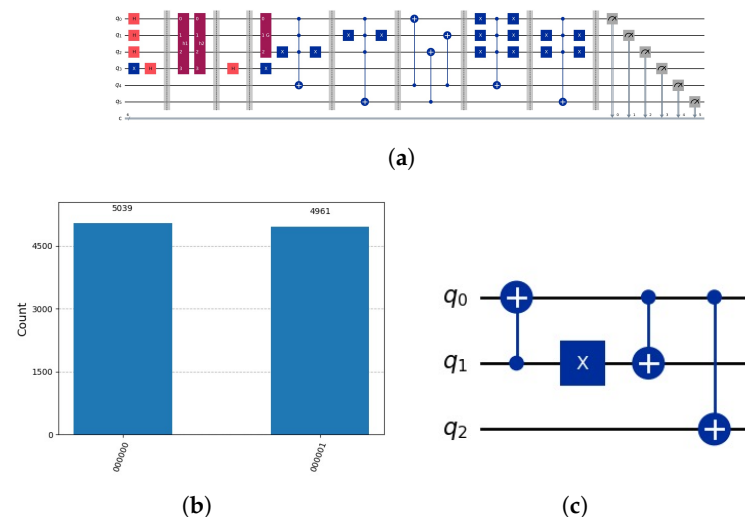


Figure 13. (a) Representation of the permutation operator $P \cdot \left(\frac{1}{\sqrt{2}} \cdot (|01\rangle + |10\rangle) \right) = |00\rangle \otimes \frac{1}{\sqrt{2}} \cdot (|0\rangle + |1\rangle)$ by controlled not gates. First we indicate the two paths h_k in superposition by two flag qubits 4 and 5. We used the two flag qubits and the knowledge of the two paths h_1, h_2 to perform the permutation mapping. Knowing the permutation mapping, we disentangle the two flag qubits 4 and 5 by mapping them to zero. Notation: H indicates Hadamard gate, X indicates NOT Gate and is used in as well in controlled not gates, h_1 and h_2 partial candidate solution circuits that marks the solution in the auxiliary qubits. G indicates Grover's algorithm circuit. (b) The correct mapping $|00\rangle \otimes \frac{1}{\sqrt{2}} \cdot (|0\rangle + |1\rangle)$ without entanglement of the additional qubits 3, 4 and 5 that are mapped to 0. (c) A compressed representation of the permutation operator P .

Knowing h_k paths defines distinct nodes of the same depth simultaneously. For instance, in the quantum search tree depicted in Figure 6, the two extended oracles would be $o_1(011, u)$ and $o_2(000, u)$, with $g = 3$, and we would seek the solution u . This approach is appropriate for a tree search where we presume that certain nodes at a depth g signify potential solutions. Alternatively, we could implement the two extended oracles directly through a quantum circuit using concatenation.

7. Discussion

7.1. Partial Candidate Solution $h(y)$

Within the context of quantum tree search, instead of employing a heuristic function $\mu(y)$, we will utilize the function $h(y)$, which signifies a potential solution, and y represents a path from the root to the corresponding node. In contrast to the heuristic function $\mu(y)$, which can mark multiple nodes, $h(y)$ should exclusively mark a single potential solution for the node. A set of partial candidate solutions can be identified by determining their similarity or distance to the goal state. However, how can we pinpoint a single potential solution, a single partial candidate solution? One approach to achieve this is to introduce an additional constraint that narrows the set to a single function, denoted by $h_k(y)$. Such a constraint could, for instance, specify the possible position within a subtree, thereby reducing the set to a single function. It is important to note that this approach is highly probabilistic and may occasionally yield non-existent partial candidate solutions or values exceeding one. Additionally, we have to take care that v is a constraint, as indicated by the Equation (23). In addition, further research should investigate the partial candidate solution $h_k(y)$ function on examples, as indicated in the paper [5].

7.2. Generalized Quantum Tree Search

We must have prior knowledge of the depth m off the search tree. This constraint can be circumvented through iterative deepening. In an iterative deepening search, we progressively increase the search limit from one to two to three to four and continue until a goal is discovered. For each limit, a search is conducted from the root to the maximum depth of the search tree. Should the search prove unsuccessful, a new search is initiated with a deeper limit. During the iterative deepening search, states are generated multiple times [23,24]. The time complexity of an iterative deepening search is comparable to that of a search to the maximum depth [23]. A quantum iterative deepening search is equivalent to an iterative deepening search [6]. For each limit max , a quantum tree search is performed from the root, where max represents the maximum depth of the search tree. The potential solutions are determined through measurement.

For a non-constant branching factor, the maximal branching factor B_{max} has to be used for the quantum tree search [6]. For the maximum value of B_{max} , the quantum algorithm using qubit representation outperforms the classical tree search described by the effective branching factor b in the specified case

$$b > b_q = \sqrt{B_{max}}. \quad (47)$$

For a large number of instances with varying initial and goal states, the effective branching factor converges to the average branching factor for an uninformed tree search, as shown in [5].

Consider each branching factor as a potential local path. For a non-constant branching factor, the quantum tree search utilizes the maximal branching factor, denoted by B_{max} . If, in a node v , $B_v < B_{max}$, only B_v are required, and $B_{max} - B_v$ local paths are not utilized. To address this limitation, we augment the path descriptor by repeating the paths. This

approach ultimately leads to k solutions that converge to the effective branching factor, as demonstrated in [5].

8. Conclusions

A primary outcome of the investigation is that conventional heuristic functions are unsuitable for quantum tree search due to the inability to disentangle the subspaces utilized by Grover's algorithm. Furthermore, the theoretical concepts of the potential permutations within the subspace $\mathcal{L}$ suggest that due to entanglement, we cannot determine the permutation operator P without knowing the paths that represent the values of the heuristic functions.

Consequently, only partial candidate solutions possess the capability to disentangle these subspaces. We assume the existence of a set of partial candidate solutions, denoted by $\aleph$, that can be determined, for instance, by their similarity to the goal state. We impose the constraint that within this set, only one maximal partial candidate solution, denoted by $h_k(y)$, identifies a potential solution. Each partial candidate solution of the set $\aleph$ can be constrained to be valid only in a specific subtree. It is important to note that a partial candidate solution is not a position in the search tree itself, as its values are unknown and are determined by Grover's algorithm and may occasionally yield non-existent partial candidate solutions or values exceeding one candidate solution.

With the existence of the partial candidate solution $h(y)$, the iterative approach represents a considerable speedup with the costs

$$v \cdot \left(\sqrt{2^{m/2}} + \sqrt{2^{m/2}} + 1 \right) = v \cdot \left(2 \cdot 2^{m/4} + 1 \right) \quad (48)$$

We can express the cost of iterative search in O notation, given $v \approx m$, since $n = 2^m$ as

$$O(v \cdot n^{1/4}) = O(\log n \cdot n^{1/4}).$$

Combinatory approach with the possible existence of the upper oracle $u(z)$ allows us to disentangle $\mathcal{L}$ and $\mathcal{U}$ using

$$\mathcal{H} = \mathcal{U}_{h_k \neq h_1} \otimes \mathcal{U}_{h_k \neq h_2} \otimes \cdots \otimes \mathcal{U}_{h_k \neq h_v} \otimes \mathcal{L}$$

which speeds up the costs minimally.

The work suggests a novel direction based on a partial candidate solution rather than heuristic functions that are unsuitable for quantum tree search. All Qiskit (v2.2) examples are presented in a Jupyter Notebooks that can be freely downloaded at <https://github.com/andrzejwichert/Nested-Quantum-Tree-Search> (accessed on 16 December 2025).

Funding: This work was supported by national funds through FCT, Fundação para a Ciência e a Tecnologia, under project UIDB/50021/2020. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Institutional Review Board Statement: The funders had no role in the study design, data collection and analysis, decision to publish, or preparation of the manuscript. This article does not contain any studies with human participants or animals performed by any of the authors.

Data Availability Statement: The data presented in this study are openly available in [Nested-Quantum-Tree-Search] at <https://github.com/andrzejwichert/Nested-Quantum-Tree-Search> (accessed on 16 December 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Montanaro, A. Quantum-Walk Speedup of Backtracking Algorithms. *Theory Comput.* **2018**, *14*, 1–24.
2. Rennela, M.; Brand, S.; Laarman, A.; Dunjko, V. Hybrid divide-and-conquer approach for tree search algorithms. *Quantum* **2023**, *7*, 959.
3. Cerf, N.J.; Grover, L.K.; Williams, C.P. Nested quantum search and structured problems. *Phys. Rev. A* **2000**, *61*, 032303.
4. Qiskit-Contributors. Qiskit: An Open-Source Framework for Quantum Computing. 2023. Available online: <https://zenodo.org/records/8190968> (accessed 16 December 2023).
5. Wichert, A. *Quantum Artificial Intelligence with Qiskit*; CRC Press: Boca Raton, FL, USA, 2024.
6. Tarrataca, L.; Wichert, A. Quantum Iterative Deepening with an application to the Halting problem. *PLoS ONE* **2013**, *8*, e57309.
7. Grover, L.K. A fast quantum mechanical algorithm for database search. In *STOC '96: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*; ACM: New York, NY, USA, 1996; pp. 212–219. <https://doi.org/10.1145/237814.237866>.
8. Grover, L.K. Quantum Mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.* **1997**, *79*, 325.
9. Grover, L.K. A framework for fast quantum mechanical algorithms. In *STOC '98: Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*; ACM: New York, NY, USA, 1998; pp. 53–62. <https://doi.org/10.1145/276698.276712>.
10. Grover, L.K. Quantum Computers Can Search Rapidly by Using Almost Any Transformation. *Phys. Rev. Lett.* **1998**, *80*, 4329–4332. <https://doi.org/10.1103/PhysRevLett.80.4329>.
11. Aharonov, D. Noisy Quantum Computation. Ph.D. Thesis, Hebrew University, Jerusalem, Israel, 1999.
12. Bennett, C.H.; Bernstein, E.; Brassard, G.; Vazirani, U. Strengths and Weaknesses of Quantum Computing. *SIAM J. Comput.* **1997**, *26*, 1510–1523. <https://doi.org/10.1137/S0097539796300933>.
13. Boyer, M.; Brassard, G.; Hoyer, P.; Tapp, A. Tight bounds on quantum searching. *Fortschritte Phys.* **1998**, *46*, 493.
14. Brassard, G.; Hoyer, P.; Tapp, A. Quantum Counting. In *Automata, Languages and Programming*; Springer: Berlin/Heidelberg, Germany, 1998.
15. Brassard, G.; Hoyer, P.; Mosca, M.; Tapp, A. Quantum Amplitude Amplification and Estimation. *arXiv* **2000**, arXiv:quant-ph/0005055.
16. Kaye, P.R.; Laflamme, R.; Mosca, M. *An Introduction to Quantum Computing*; Oxford University Press: New York, NY, USA, 2007.
17. Ventura, D.; Martinez, T. Quantum associative memory with exponential capacity. In *Proceedings of the 1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence*, Anchorage, AK, USA, 4–9 May 1998; Volume 1, pp. 509–513.
18. Ventura, D.; Martinez, T. Quantum associative memory. *Inf. Sci.* **2000**, *124*, 273–296.
19. Tay, N.; Loo, C.; Perus, M. Face Recognition with Quantum Associative Networks Using Overcomplete Gabor Wavelet. *Cogn. Comput.* **2010**, *2*, 297–302. <https://doi.org/10.1007/s12559-010-9047-2>.
20. Trugenberger, C.A. Probabilistic Quantum Memories. *Phys. Rev. Lett.* **2001**, *87*, 067901.
21. Trugenberger, C.A. Quantum Pattern Recognition. *Quantum Inf. Process.* **2003**, *1*, 471–493.
22. Diamantini, M.C.; Trugenberger, C.A. Mirror modular cloning and fast quantum associative retrieval. *arXiv* **2022**, arXiv:2206.01644.
23. Korf, R.E. Depth-first iterative-deepening: An optimal admissible tree search. *Artif. Intell.* **1985**, *27*, 97–109. [https://doi.org/10.1016/0004-3702\(85\)90084-0](https://doi.org/10.1016/0004-3702(85)90084-0).
24. Russell, S.; Norvig, P. *Artificial Intelligence: A Modern Approach*; Prentice Hall series in artificial intelligence; Prentice Hall: Hoboken, NJ, USA, 2010.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.