TUDelft

Delft University of Technology

Explaining Grover's algorithm with a colony of ants: a pedagogical model for making quantum technology comprehensible

Schalkers, M.A.; Dankers, K.T.; Wimmer, M.T.; Vermaas, P.E.

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

PAPER • OPEN ACCESS

Explaining Grover's algorithm with a colony of ants: a pedagogical model for making quantum technology comprehensible

View the article online for updates and enhancements.

You may also like

Explaining Grover's algorithm with a colony of ants: a pedagogical model for making quantum technology comprehensible

Merel A Schalkers[1,*] , **Kamiel Dankers**[2] ,
Michael Wimmer[2] **and Pieter Vermaas**[3]

[1] Department of Applied Mathematics, Delft University of Technology, Delft, The Netherlands
[2] Qutech and Kavli Institute of Nanoscience, Delft University of Technology, Delft, The Netherlands
[3] Department of Philosophy, Delft University of Technology, Delft, The Netherlands

E-mail: m.a.schalkers@tudelft.nl

Abstract

The rapid growth of quantum technologies requires an increasing number of physicists, computer scientists, and engineers who can work on these technologies. For educating these professionals, quantum mechanics should stop being perceived as incomprehensible. In this paper we contribute to this change by presenting a pedagogical model for explaining Grover's search algorithm, a prominent quantum algorithm. This model visualizes the three main steps of Grover's algorithm and, in addition to explaining the algorithm itself, introduces three key principles of quantum mechanics: superposition, interference, and state collapse at measurement. The pedagogical model, visualized by a video, is called the *ant colony maze model*. It represents the

* Author to whom any correspondence should be addressed.

search problems as finding the exit of a maze, and visualizes Grover's search algorithm as a strategy by which a colony of ants finds that exit.

1. Introduction

The rapid growth of quantum technologies requires a substantially increasing number of physicists, computer scientists, and engineers working on these technologies. This development can be expected to remain in place for the coming decade, and necessitates educating large contingents of students who can work on quantum technologies. While being of a different nature than constructing quantum computers, quantum internet, or quantum sensors, scaling up teaching for future quantum technologists is urgent and challenging as well. Having the famous quote by Richard Feynman in mind—*I think I can safely say that nobody understands quantum mechanics* [Fey95]—quantum mechanics, being the basis of quantum technologies, has the connotation of being incomprehensible, framing understanding this basis as a quest only a few privileged can realize. In this paper we aim to contributing to changing this perspective on quantum mechanics and enabling the education of larger contingents of students. Specifically, we contribute to education in quantum technology by presenting a pedagogical model for explaining a quantum algorithm: Grover's search algorithm [Gro96].

We submit the *ant colony maze model* as a pedagogical model for explaining Grover's algorithm in teaching and for introducing key concepts of quantum mechanics. Moreover, by the graphic nature of the model, it can also provide nonprofessionals an informed feel of how quantum computing works, thus also contributing to an understanding of quantum technologies in society at large [Ver17], breaking away from the Feynmanian frame that understanding the quantum realm is a privilege of the few.

Section 2 introduces quantum computing and section 3 describes Grover's algorithm. In section 4 we present our ant colony maze model. Existing visualizations of Grover's algorithm are reviewed in section 5 and compared to our model in section 6. Section 7 is for conclusions.

2. Quantum computers

2.1. Overview

Since its discovery in the 1920s quantum mechanics have played an important role in physics. Not long after, quantum mechanical phenomena were embraced by engineers, with the potential of exploiting rather than just describing its properties.

In 1980 the idea of quantum computation was first suggested by Benioff [Ben80]. Later Yuri Manin and Richard Feynman proposed it for efficiently simulating quantum processes [Man07, Fey82], Quantum computing remained a rather small area of research until in 1994 Peter Shor published an algorithm for efficiently factorizing large numbers with quantum computers, thus highlighting the field's potential.

Nowadays both governmental organizations and private companies have invested millions in the technology with private companies being in the lead in developing quantum computers. Google was the first to claim a *quantum advantage*, i.e. a quantum computer solving a problem faster than a classical computer [Aru+19]. Moreover, IBM currently offering 24 quantum computers online, the largest machine being the IBM Ospray containing 433 physical quantum bits, or *qubits* [IBMb]. Other companies such as D-Wave, Rigetti, Honeywell, and Google, and academic research institutes such as Oxford University and Delft University of Technology, also offer quantum computers for running algorithms via the cloud. Despite the great progress that has been made in the last decades and specifically the last couple of years, there are still significant challenges to be overcome before quantum computing can live up to its promises. Before we can expect large scale

industrial applications more qubits with a higher precision and stability will be required[4].

2.2. Calculating with qubits

Calculating with qubits differs from calculating with classical bits in two central ways, and precisely these differences provide the potential and subtleties of quantum computing.

Just like classical bits, qubits can be in states that correspond to 0 or 1. These states are written as $|0\rangle$ and $|1\rangle$ in quantum computing. A first difference between classical bits and qubits is that the latter can also be in states that are so-called superpositions of 0 and 1. Such states $|\phi\rangle$ are a complex linear combination of the states $|0\rangle$ and $|1\rangle$ that corresponds to 0 and 1:

$$|\phi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle, \tag{1}$$

with α_0 and α_1 complex numbers in $\mathbb{C}$, and $|\alpha_0|^2 + |\alpha_1|^2 = 1$.

For determining whether a qubit has a value 0 or 1, one has to perform a measurement on the qubit in the computational basis[5]. If the qubit is initially in state $|0\rangle$, one finds 0 with the measurement, and when the qubit is in state $|1\rangle$, one finds 1. If however the state of the qubit is a superposition of $|0\rangle$ and $|1\rangle$, there is $|\alpha_0|^2$ probability of finding 0 and $|\alpha_1|^2$ probability of finding 1. The amplitudes α_0 and α_1 in the superposition thus represent the probabilities of finding 0 and 1 upon measurement on the qubit.

When extending the idea of superposition to multiple qubits, the quantum state $|\phi\rangle$ of n qubits is a combination of the states of the separate qubits. For n qubits we have 2^n basis states (as each separate qubit has two basis states $|0\rangle$ and $|1\rangle$), meaning that we can see an n-qubit state as a complex combination of $N = 2^n$ basis states $|i\rangle$. So n qubits have a combined quantum state that spans

a space that is exponentially large in the number of qubits. It is, however, important to keep in mind that only one basis state $|i\rangle$ will be found upon measurement. Mathematically we can express a general n-qubit quantum state as:

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

with $\alpha_i \in \mathbb{C}$ for all $i \in \{0, 1, \ldots, 2^n - 1\}$ and $\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$. Now the probability of finding the state $|i\rangle$ upon measurement is $|\alpha_i|^2$. To translate such a state $|i\rangle$ back to the value of the separate qubits, we can express the number i in binary, to see precisely which qubit has which value in this basis state.

When we start performing calculations on n qubits, it is easiest to interpret the quantum state of these qubits as a large vector in complex space. More specifically, an n-qubit quantum state can be seen as a vector $|\phi\rangle$ in $\mathbb{C}^{2^n}$. We can then interpret an operation on such a quantum state as a multiplication of the state $|\phi\rangle$ by a matrix U in $\mathbb{C}^{2^n \times 2^n}$. Furthermore, since we require that the amplitudes α_i of a quantum state as expressed in equation (2) have the property that their absolute values squared sum to 1, this matrix U must be unitary.

Having introduced the basic qubit properties and how to apply operations to them, it can intuitively be seen how quantum computing can lead to a computational speed-up when properly exploited. Assume we have n qubits. The quantum state $|\phi\rangle$ of these qubits can then be written as a linear combination of $r \leqslant 2^n$ linearly independent vectors $|\phi_i\rangle \in \mathbb{C}^{2^n}$, that is, as $|\phi\rangle = \sum_{i=1}^{r} \alpha_i |\phi_i\rangle$. Now, when applying the unitary operation U to the quantum state $|\phi\rangle$ we get $U|\phi\rangle = \sum_{i=1}^{r} \alpha_i U|\phi_i\rangle$ using basic linear algebra. This implies that quantum computing allows us to perform in parallel r different operations on r different states $|\phi_i\rangle$, with the number r potentially growing exponentially with the number n of qubits. This is often referred to as 'quantum parallelism'. Notice, however, that since we will only find *one* computational basis state upon measurement, this parallelism is not immediately useful. More precisely, a quantum algorithm needs to consist of quantum operations that yield a

---

[4] Experts often make a distinction between noisy physical qubits and theoretical noiseless qubits, called *logical qubits*. Research focuses on composing these logical qubits of large sets of physical qubits, thus increasing the task to create quantum computers with large numbers of physical qubits.

[5] A quantum measurement always needs to be performed with respect to a certain basis. The computational basis is the standard choice consisting of the basis $|0\rangle$, $|1\rangle$ that corresponds to the qubit values 0 and 1.

final state with a very high probability in the desired solution, such that it can be found in a measurement.

By exploiting these properties of quantum computers, quantum algorithms can be designed that solve problems potentially more efficient than algorithms for classical computers.

3. Grover's search algorithm

One quantum algorithm that achieves a speed-up over its classical counterparts is Grover's search algorithm. Grover's algorithm was first described in 1996 [Gro96], and remains an important part of education in quantum technology to this date. This is due to the fact that it is a relatively simple algorithm that elegantly exploits the potential of quantum computing, thus being a good example for learning the tricks of the trade.

Even though Grover's algorithm might be relatively simple, novice students may focus on understanding the mathematics and miss to see the steps in the algorithm that are enabled by quantum mechanics. This paper gives a pedagogical model for explaining these quantum steps in Grover's algorithm in an intuitive way that does not request a deep knowledge of the underlying mathematics. There are three such steps and we describe them as each exploiting a key principle of quantum mechanics. Later we use this to show how our pedagogical model does a better job at introducing these principles than existing models.

3.1. The search problem

The search problem tackled by Grover's algorithm is, in computer science terms, searching an unstructured database: the problem is to find items from a search space that have a certain property. For example, this could be items fulfilling a set of mathematical equations. This search problem can be translated into quantum language as the task of finding within a large space $\mathbb{C}^{2^n}$ spanned by 2^n basis states those basis states $|i\rangle$ that represent the items with the desired property.

The search problem in Grover's algorithm is defined in terms of an *oracle* function. Such a quantum oracle function can recognize the correct states $|i\rangle$ within the large set of all quantum basis states: when giving a general quantum state $|\phi\rangle$ as input, the oracle function recognizes the basis states $|i\rangle$ with the desired property and marks them with a minus sign. It is important here to note that the oracle does not actually 'know' which are the basis states $|i\rangle$ representing items with the desired property; it can only recognize whether a basis state is a 'good' one and which basis state is a 'bad' one, when given an input. Generally, recognizing a correct solution is easier than knowing it—for example it is easier to check if a solution fulfills an equation than to derive the solution for that equation. We denote this quantum oracle by O_X where X is the set of indices i of basis states $|i\rangle$ that represent items with the desired property. This quantum oracle is hence problem-dependent. As such the oracle function is the only part of the quantum circuit of which the implementation will be different depending on problem, the remainder of the circuit of Grover's algorithm is universal.

One could compare the problem of searching an unstructured database to finding in a maze a path that leads to the exit of the maze, and we use this comparison in our pedagogical model for explaining Grover's algorithm. Most paths in a maze do not lead to the exit, but some paths have this special property, and we want one of those. From this example the difference between knowing and recognizing a correct solution can be better understood: knowing which paths in a maze lead to the exit is much more difficult than simply recognizing whether or not a given path does so.

3.2. Description of Grover's algorithm

Grover's algorithm can be divided into three steps. The first can be interpreted as creating a state covering all items in the database. The second step as homing in on the 'good' items with the oracle function. Subsequently the third step reveals one of those 'good' items.

3.2.1. Step 1: spreading out over the space.
We start with our qubits in the state $|0\rangle$, and the first thing we want to do is spread out our quantum state over the full space $\mathbb{C}^{2^n}$. We can do this by applying an operation that is standard in quantum

computing: applying the Hadamard gate[6] H to all n qubits. This operation gives us a new state, called $|\mathcal{U}\rangle$, which is:

$$|\mathcal{U}\rangle = H^{\otimes n}|0\rangle = \frac{1}{\sqrt{2^n}}\sum_{i=0}^{2^n-1}|i\rangle. \qquad (3)$$

This means that our qubits are brought in a state that is an equal superposition of all basis states and thus spread out over the full space $\mathbb{C}^{2^n}$. By being in this state $|\mathcal{U}\rangle$, we know with certainty that all the 'good' basis state corresponding to a solution of our search problem are also present in the quantum state of the n qubits. We can thus write this state $|\mathcal{U}\rangle$ as a linear combination of the 'good' states and of the 'bad' states. The superposition of the 'good' states is $|\mathcal{G}\rangle = \frac{1}{\sqrt{\dim(X)}}\sum_{i:i\in X}|i\rangle$, and the superposition of the 'bad' states is $|\mathcal{B}\rangle = \frac{1}{\sqrt{2^n-\dim(X)}}\sum_{i:i\notin X}|i\rangle$. Then we get:

$$|\mathcal{U}\rangle = \sin(\theta)|\mathcal{G}\rangle + \cos(\theta)|\mathcal{B}\rangle, \qquad (4)$$

with $\theta = \arcsin(\sqrt{\dim(X)/2^n})$.

In this first step Grover's algorithm creates a state that is a linear combination of all basis states. In terms of our maze metaphor this means that after step 1 Grover gives a state that describes all possible paths in the maze: the 'good' ones that lead to the exit, and the 'bad' ones that do not.

3.2.2. Step 2: Grover iterate. After having spread out the quantum state to be in an equal superposition of all possible basis states, we use the oracle function to 'mark' the 'good' basis states that represent the items with the searched for desired property. Specifically the oracle O_X will multiply the 'good' basis states with the factor -1. This leads to the quantum state:

$$|\mathcal{U}_1\rangle = O_X|\mathcal{U}\rangle = \frac{1}{\sqrt{n}}\sum_{i=0}^{2^n-1}(-1)^{x_i}|i\rangle. \qquad (5)$$

In the above equation $x_i = 1$ if $i \in X$ and $x_i = 0$ otherwise. This operation is equivalent to reflecting $|\mathcal{U}\rangle$ through the equal superposition $|\mathcal{B}\rangle$ of all 'bad' states: $O_X = 2|\mathcal{B}\rangle\langle\mathcal{B}| - I$.

[6] $H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$.

Subsequently we apply another reflection operation $H^{\otimes n}R_0 H^{\otimes n} = 2|\mathcal{U}\rangle\langle\mathcal{U}| - I$ on the newly obtained state $O_X|\mathcal{U}\rangle$. Here, R_0 marks all but the $|0\rangle$ state with a factor -1 and combined with the $H^{\otimes n}$ operation, which turns the $|0\rangle$ state into $|\mathcal{U}\rangle$, it can be seen that the total operation is a reflection through the state $|\mathcal{U}\rangle$. Both reflections are together called a *Grover iterate*, and after one such an iterate, we are in the state [Wol22]:

$$|\mathcal{U}_2\rangle = \sin(2\theta + \theta)|\mathcal{G}\rangle + \cos(2\theta + \theta)|\mathcal{B}\rangle. \quad (6)$$

The combination of the two reflections, first around the 'bad' states $|B\rangle$ and finally around the starting state $|U\rangle$ has the combined effect of rotating away from the starting state $|\mathcal{U}\rangle$ by an angle 2θ such that the resulting state is closer to the 'good' states. That the combination of the two reflections results in this rotation can be seen geometrically from figure 2.

Applying this Grover iterate k times brings us to the state:

$$|\mathcal{U}_k\rangle = \sin((2k+1)\theta)|\mathcal{G}\rangle + \cos((2k+1)\theta)|\mathcal{B}\rangle. \qquad (7)$$

Using this expression it can be seen that if we chose k as $\frac{\pi}{4\theta} - \frac{1}{2}$, we get $\sin((2k+1)\theta) = 1$ and $|\mathcal{U}_k\rangle = |\mathcal{G}\rangle$. This choice is possible if the value $\frac{\pi}{4\theta} - \frac{1}{2}$ is an integer; if this is not the case, we take k to be close to this value, and we get $|\mathcal{U}\rangle \approx |\mathcal{G}\rangle$. The idea is that we perform this rotation k times such that the n qubits are in a state that is very close or equal to the state $|\mathcal{G}\rangle$ that constitutes the equal superposition of the 'good' states. In both cases we end up with a state that is very close or even equal to a superposition of the 'good' states.

Returning to the maze: Grover's algorithm brings the state of the n qubits by step 2 to one where most of the weight is on the 'good' paths that lead to the exit. Moreover, the operations of steps 1 and 2 can be carried out in a predetermined amount of time.

3.2.3. Step 3: Measurement. Finally we measure the state of the n qubits. After step 2 the state of the n qubits is $|\mathcal{U}_k\rangle$, which is equal or very close to the superposition $|\mathcal{G}\rangle$ over all possible 'good' basis states. Then, upon measurement, the probability of finding one of the 'good' basis states

that represent an item with the desired property, is equal, or almost equal, to one. Moreover, in quantum mechanics the state of a system collapses after the measurement to the basis state that corresponds to the found outcome. Hence, after the measurement the qubits are with probability 1, or almost 1, in a 'good' state, that is, in a basis state that represents a solution to the search problem.

In terms of the maze: by the measurement in this last step we find one path. The more weight on a path, the larger the chance that we will get this path. And Grover's algorithm ensures that the 'good' paths that lead to the exit have together a weight that is almost 1. So the chance that we find such a successful path becomes very large.

3.3. The three key quantum principles in Grover's algorithm

Each of the three steps of Grover's algorithm makes use of one of the key quantum principles: superposition, interference, and measurement. Hence, explaining the algorithm is a good way to introducing students to the computations means that quantum mechanics offers.

Step 1 of Grover's algorithm introduces superposition as a key quantum principle: the initial state of the quantum bits in the quantum computer is brought in a linear combination of all basis states, where some basis state represents a possible solution within the search and has a certain amplitude. Students then learn that qubits can be in states that are superpositions of other states.

Step 2 introduces the key principle of interference. The quantum state of the qubits is made to evolve in a way that the amplitudes of the basis states that represent 'bad' solutions to the search problem become close to zero, and that the amplitudes of the 'good' basis states grow. Students learn that states of qubits can interfere to fade out or become more present.

Step 3 of Grover's algorithm presents the key quantum principle of measurement. The state of the qubits collapses with a probability close to 1 to a basis state representing a solution within the search. Students learn that at measurements the states of qubits make transitions from superpositions of basis states to single basis states.

4. The ant colony maze model

In our description of Grover's algorithm in section 3 we already used the metaphor of the maze for Grovers algorithm. Now we introduce the ant colony maze model as a pedagogical model for explaining the workings of Grover's search algorithm on a quantum computer. This model comes as a video that can be accessed online[7]. We first introduce the ingredients of the ant colony maze model and how they relate to the elements of quantum computing. Then we explain how the dynamics in the ant colony maze model represents the three steps of Grover's search algorithm.

4.1. The ingredients

The ant colony maze model consists of a maze with different paths, and a colony of ants trying to find the exit of the maze, marked by food (see figure 1(a), which is a still from the video representing the model). The ant colony as a whole represents the quantum computer. The paths in the maze represent the different possible solution states $|i\rangle$ of equation (3) and the path marked with food represents the correct outcome state $|\mathcal{G}\rangle$ for the given instance of Grover's search problem. Each ant represents an equal discrete density, the amount of ants on a certain path i in the maze after k timesteps thus represent the weight of the basis state $|i\rangle$ representing that path in the superposition of equation (7). This means that after k timesteps the amount of ants on a path (see, e.g. the video still in figure 1(b)) represent a discrete probability of measuring that path; in the model it is assumed that upon a measurement of the ant colony, only one ant (each with equal probability) will be found, and that the path this ant is on represents the found state $|i\rangle$ upon measurement. The ants traveling on a path that leads to food will keep travelling on this path and the ants on an unsuccessful path will choose another path to try for food. This mechanism represents the role of the oracle, as it leads to an increase of the ants traveling on the path that leads to the exit of the maze in each timestep.

(a) Video still of the maze in the Ant Colony Maze model; one path leads to the exit.

(b) Video still after step 1, when the ants have spread out evenly over the maze.

(c) Video still after step 2, when almost all ants can be found on the path to the food.

(d) Video still after the measurement in step 3, when one ant is found on the path to the food.
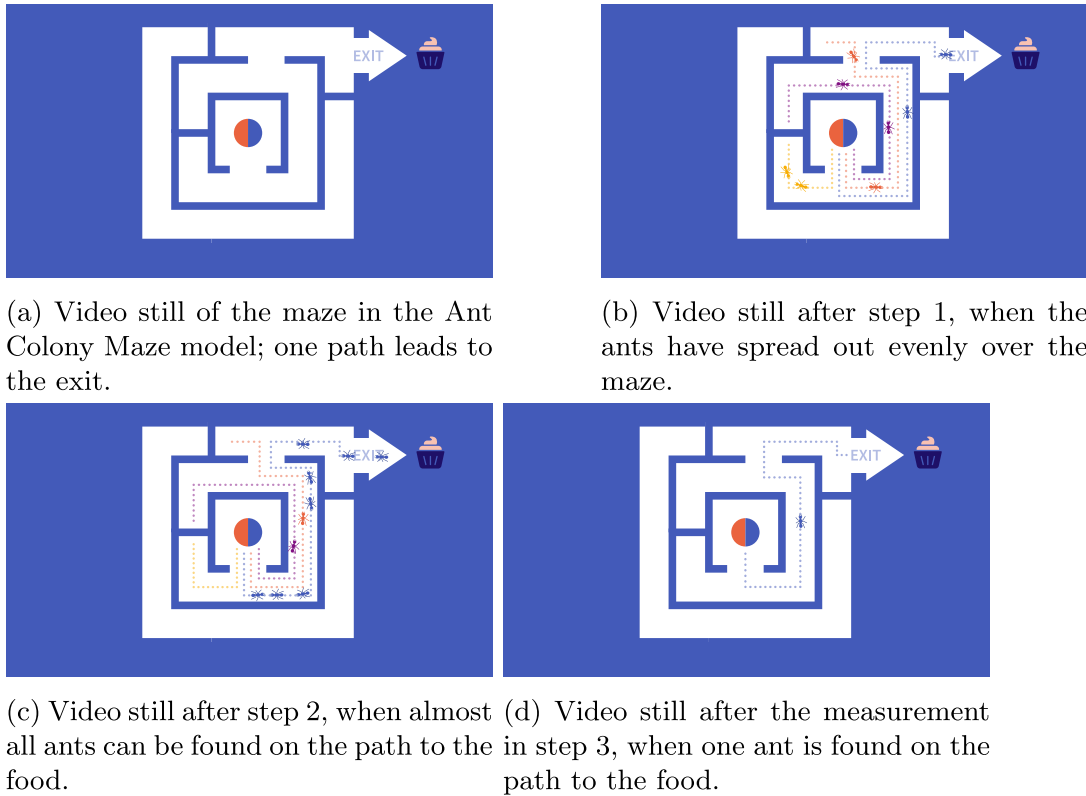
Figure 1. Video stills showing the different steps of the ant colony maze model.

Below we explain how the ants move through the maze over time and how this behavior can be interpreted as the workings of a quantum computer running Grover's search algorithm as described in section 3.

4.2. The three steps

The objective for the ants, of course, is to find food and bring as much as possible back to the ant hill. In the first step the ants spread out evenly over the maze as shown in figure 1(b). This can be interpreted as the start of Grover's algorithm where the quantum state is taken from the $|0\rangle$ state into an equal superposition state $|\mathcal{U}\rangle$, which represents the operation given in equation (3).

At this point in the video we perform a measurement, i.e. we choose a random ant and its corresponding path. Since all ants are distributed randomly over all paths, the chosen ant does with large likelihood not know the correct path. This

strikingly highlights that superposition alone is indeed not useful for finding the solution.

In the second step ants that found food will keep traveling on the same 'good' path, collecting food for the colony. Ants that tried a 'bad' path and did not find food will try different paths until they find one that leads to the food. This represents the redistribution of weight into the correct solution by Grover's algorithm, and at the same time it mimics the phenomenon of ant trails that are known to many people: communicating via pheromones, ants will conglomerate on the path leading to food relatively quickly. Clearly, ant trails are not quantum computing, but can serve as a mental picture.

The way the distribution of the ants changes in each timestep can be interpreted as the way the state of the system $|\mathcal{U}_l\rangle$ changes to $|\mathcal{U}_{l+1}\rangle$ after applying the Grover iterate for the $l+1$-th time. Figure 1(c) shows the maze at the end of step 2 where almost all the ants have ended up on the

right path, this corresponds to the state of the system after applying the Grover iterate k times, as given in equation (7).

Now, in step 3, upon picking by measurement a random ant in the colony, this ant is most likely on the path which ends in food (see figure 1(d)). To bring that back to Grover's search problem, we can interpret picking an ant on a path that leads to food as the probability of a 'good' state to be found after measurement on the state given by equation (7).

5. Existing visualizations of Grover's algorithm

We submit the ant colony maze model as a pedagogical model for explaining Grover's algorithm in teaching and for introducing the key quantum-mechanical principles of superposition, interference and measurement. For supporting these claims we describe in this section current visualizations of Grover's algorithm used in papers and textbooks, and highlight in the next section how the ant colony maze model differs from these visualizations and offers improvements.

5.1. The arrow diagram

A peculiar aspect about Grover's algorithm is that the quantum state of the n qubits after each complete Grover iterate can be written as a linear combination of two states $|\mathcal{G}\rangle$ and $|\mathcal{B}\rangle$ with real coefficients, as shown in equations (4), (6) and (7). Hence, this quantum state can be immediately represented as a unit vector in two-dimensional space, with the x and y-axes representing $|\mathcal{B}\rangle$ and $|\mathcal{G}\rangle$, respectively. As a consequence, Grover's algorithm can be visualized geometrically [NC10] in the form of an 'arrow diagram'. The initial state $|\mathcal{U}\rangle$ has the angle θ with the $|\mathcal{B}\rangle$-axis as shown in equation (4). Hence, the unit vector representing this initial state $|\mathcal{U}\rangle$ is nearly aligned with the $|\mathcal{B}\rangle$-axis, i.e. the angle θ is small, as the dimension of the solution space of the 'good' states is much smaller than 2^n, as shown in figure 2(a). The first Grover iterate then corresponds to rotating the unit vector $|\mathcal{U}\rangle$ by an angle 2θ towards the $|\mathcal{G}\rangle$-axis to the unit vector $|\mathcal{U}_1\rangle$, as given in equation (6).

Hence, the quantum state of the n qubits moves closer to the $|\mathcal{G}\rangle$-axis, as shown in figure 2(b).

The strength of the geometrical visualization is that it represents the mathematical formulas put forward in section 3.2. In particular, the visualization shows that Grover iterates bring the vector $|\mathcal{U}\rangle$ close to the desired solution $|\mathcal{G}\rangle$ in a finite number of steps.

The arrow diagram clearly serves towards a mathematically inclined reader rather than a broader audience. Additionally, this visualization has the disadvantage of hiding the typically vastly different dimension of the solution space compared to the complete Hilbert space. Specifically, the arrow diagram represents the solution space as a 1D subspace in a 2D space. Hence, superposition is represented in an abstract way with the unit vector having components in $|\mathcal{G}\rangle$ and $|\mathcal{B}\rangle$, but the fact that the initial state is a *uniform* superposition of all states is ill-represented. The arrow diagram gives a faithful representation of the Grover iteration mathematics, but the concept of interference is not clearly represented. In fact, rotating the arrow seems to rather emphasize a classical mechanism, like turning a knob. Finally, measurement can be represented as projection on the x and y-axis. However, it again is impeded by the misrepresentation of dimensionality.

5.2. The bar graph

A second commonly employed visualization uses a bar graph to show the (real) amplitude of every quantum state of the n qubits in the Hilbert space. This was used in the original paper by Grover [Gro96], but also for example on the Qiskit website [IBMa] alongside the arrow diagram.

The initial state $|\mathcal{U}\rangle$ is now represented by 2^n bars of equal height, representing the uniform superposition of all states (figure 3(a)). The action of the oracle function flips the bars corresponding to the solution (figure 3(b)), and the amplitude of the solution increases with number of Grover iterates (figures 3(c) and (d)). In fact, a Grover iterate can be understood as a 'reflection around the average' of the amplitudes [Gro96].

The bar graph thus illustrates the concept of superposition well, by showing the equal
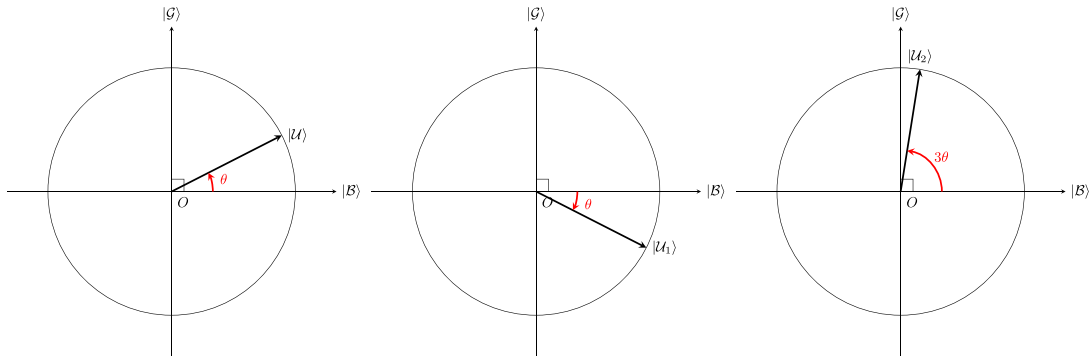
Figure 2. Visual representation of the state $|\mathcal{U}\rangle$ and how it is reflected by a Grover iterate to get closer to the state $|\mathcal{G}\rangle$.
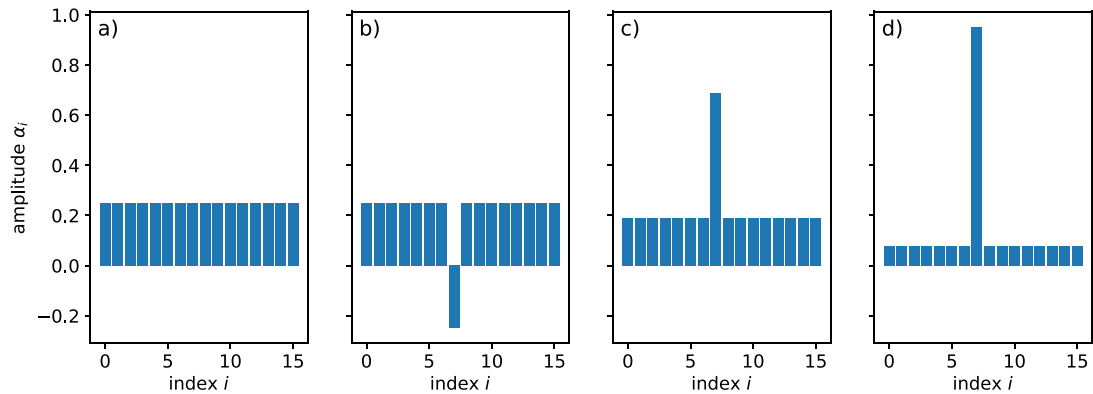

Figure 3. A bar diagram represents the real amplitudes of the wave function of (a) the initial state, (b) after application of the oracle, (c) after one Grover iterate, and (d) after two Grover iterates. There is only one solution in this example.

amplitudes of all quantum states. It also shows the result of interference—the amplitude of the solutions grows as the amplitudes of all other states decrease—though arguably it does not show the process of interference itself.

A disadvantage of the bar graph is the fact that it is a graph and thus not aimed at a broader audience. It is rather a direct representation of the numerical values of quantum mechanical amplitudes. As such, it is faithful to the underlying mathematics, but also requires an understanding of the mathematics for interpretation. This particularly impedes the illustrative power of the bar graph to visualize measurement: the knowledge of quantum mechanical probabilities is necessary to infer the principles of measurement from the graph.

5.3. The Ball Maze model

Both the arrow diagram and the bar graph are abstract visualizations: they are agnostic to the particular search problem and do not aim to use a particular example for illustration purposes. This inherently limits these visualizations to specialists. A frequently used model to illustrate quantum computing for a non-specialist audience is the maze model, for example used in [QuT, Fer, Mic, Vie]. Finding the exit of a maze is a search problem, and can thus be used—as we also did in the ant colony maze model—to illustrate Grover's algorithm quite graphically by means of videos of finding paths in the maze.

We particularly focus on the maze example used by QuTech Academy [QuT] as a typical
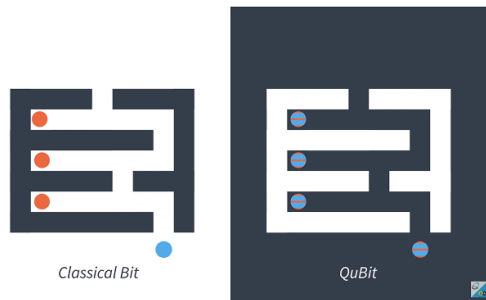
Figure 4. Still from the Ball Maze model in the video from QuTech.

representative of this class of visualization, and will refer to it as the *ball maze model*. In this model a ball needs to find the exit of the maze. The classical computer finds the exit by letting the ball sequentially try out each path until the correct one is discovered. The quantum computer has another way of working: each time an intersection is reached the ball splits in a superposition, and each part in that ball superposition then tries a separate pathway after the intersection. In the video it is made certain that the viewer does not have the misconception that the ball itself searches all paths. The ball becomes a superposition and that superposition is then used to find the right path to the exit (see, e.g. figure 4).

The Ball maze model captures the key principle of superposition in quantum computing in a beautiful and for the student understandable way: the ball represents that the quantum state of the quantum computer evolves to a superposition of the ball, with the parts in the ball superposition rolling along the different paths. Also, by adding an equal color scheme for each ball part in the superposition the principle is made clearer even further. For quantum educated people the wave-like behaviour of quantum states is visible in the color scheme.

The second key principle of quantum interference is not covered by the ball maze model. The different balls all remain present in the model, including the ones that hit a dead end in the maze and thus correspond to solutions to the maze that should fade out as balls that have a close to zero probability of being found in the maze when the search has ended.

Lastly, the third key principle of measurement is also not represented in the ball maze model. At the end of the search, when one ball has found the exit, the other less fortunate balls are still visible suggesting that at the end all parts of the super-position still exist.

6. Comparison of our model to the ball maze model

The ball maze model is most similar to ours both in ideas and audience. However, our model differs in some essential aspects, making it more powerful, which we highlight here.

Similar to the ball maze model, our ant colony maze model captures the idea of superposition by spreading the ant colony over the different paths in a maze. Our model, however, exceeds the ball maze model by incorporating how the 'weights' of the different basis states in the superposition can change over time. By letting the amount of ants walking on a specific path change each timestep, students gain the insight that a super-position must be altered by certain operations to converge towards a final solution.

This concept ties in with how we explain the measurement principle. By changing the 'weight' of the superposition over time before the final measurement, we give students insight in how the sought after solution can actually be recovered using a quantum computer. Our model portrays that measurement is a statistical process, and that the likelihood of finding the right state depends on the distribution of the ants over the paths. This is also highlighted in the video by measuring at two different points in time and showing how the (probability of the) outcomes differ. This is a substantial improvement over the ball maze model; in that model it appears that just creating an equal superposition of states is enough for correctly identifying the sought after state, whereas we show that the altering of the initial super-position is necessary to converge towards a state where finding the correct state upon measurement is likely.

The principle of interference is, however, a bit harder to capture directly due to the pure mathematical nature of the concept. In our model, unlike the maze ball model, we do capture the result of

interference (by altering the amount of ants on the paths) but not so much the process itself, similar to the bar graph and arrow diagram models.

Therefore our model is able to capture two out of the three key principles of quantum computing very well, whilst being able to give insight in the result of the third, while also being a model that does not rely on the purely mathematical visualization of the processes but instead speaks to a broader audience by translation the mathematics of quantum computing to an 'everyday' example.

7. Conclusion

In this paper we have described a new pedagogical model for explaining quantum computing with Grover's search algorithm, and called it the ant colony maze model. In this model, which comes with a video, finding the exit of a maze represents the search problem to be solved. The strategy by which a colony of ants finds that exit represents Grover's algorithm.

Our motivation to develop this pedagogical model is the need to educate increasingly larger numbers of physicists, computer scientists, and engineers for developing quantum technologies. Grover's algorithm is central to quantum computing, and for making it understandable it can be divided in three steps that each make use of one of three key principles of quantum mechanics: superposition, interference, and the collapse of quantum states at measurements. The ant colony maze model represents the three steps and thus not only explains to students how the Grover algorithm works, but also introduces these quantum principles. We therefore submit the ant colony maze model as a valuable contribution to education in quantum technologies.

The ant colony maze model builds on the ball maze model, and improves on it since the latter model represents only the principle of superposition explicitly, while the ant colony maze model represents all three quantum mechanical principles. There still is however room for improvement suggesting further research. The principle of interference is hard to capture due to the pure mathematical nature of the concept. In our model its effect is captured by decreases in

the number of ants that explores dead-end paths in the maze, yet it does not visualize interference directly by some sort of mechanism that creates that decrease.

An additional advantage of the ant colony maze model is that it can be used to explain to students the differences between computations with quantum computers and with classical computers. The video introduces this difference by representing a search with a classical computer as a search by a single ladybug that explores all paths sequentially. Finally, by its graphic nature, the model may be taken as a means for giving laypersons an informed feel of how quantum computing works, thus also contributing to an understanding of quantum technologies in society at large. The ant colony maze contributes in this way to a development that is already set into motion by the emergence of quantum technologies: counter to Feynman's deposition, understanding the quantum realm will become a privilege of the many.

Data availability statement

No new data were created or analysed in this study.

Conflict of interest

The authors have no conflicts to disclose.

ORCID iDs

Merel A Schalkers ⓘ https://orcid.org/0000-0001-7751-9060
Kamiel Dankers ⓘ https://orcid.org/0009-0000-2749-5267
Michael Wimmer ⓘ https://orcid.org/0000-0001-6654-2310
Pieter Vermaas ⓘ https://orcid.org/0000-0002-8175-9179

References

[Aru+19] Arute F *et al* 2019 Quantum supremacy using a programmable superconducting processor *Nature* **574** 505–10

[Ben80] Benioff P 1980 The computer as a physical system: a microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines *J. Stat. Phys.* **22** 563–91

[Fer] Fermilab *Quantum Maze* (available at: www.youtube.com/watch?v=gfUEUhDbGXA) (Accessed 30 January 2023)

[Fey95] Feynman R 1995 *The Character of Physical Law* (The MIT Press) p 129

[Fey82] Feynman R P 1982 Simulating physics with computers *Int. J. Theor. Phys.* **21** 467–88

[Gro96] Grover L K 1996 A fast quantum mechanical algorithm for database search *Proc. 28th Annual ACM Symp. on Theory of Computing (STOC'96)* (Association for Computing Machinery) pp 212–9

[IBMa] IBM *Grover's algorithm* (available at: https://qiskit.org/textbook/ch-algorithms/grover.html) (Accessed 30 January 2023)

[IBMb] IBM *IBM Quantum* (available at: https://quantum-computing.ibm.com/) (Accessed 5 April 2023V)

[Man07] Manin Y I 2007 *Mathematics as Metaphor: Selected Essays of Yuri I. Manin* (American Mathematical Society) pp 77–78

[Mic] Microsoft *Microsoft Ignite Vision Keynote* (available at: https://news.microsoft.com/uploads/2017/09/Satya-Nadella-transcript-Ignite-2017.pdf) (Accessed 30 January 2023)

[NC10] Nielsen M A and Chuang I L 2010 *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press)

[QuT] QuTech *QuTech Academy* (available at: www.youtube.com/watch?v=OQ-NkCiyj_c) (Accessed 30 January 2023)

[Ver17] Vermaas P E 2017 The societal impact of the emerging quantum technologies: a renewed urgency to make quantum theory understandable *Ethics Inform. Technol.* **19** 241–6

[Vie] Vienneau K In the blink of an eye: uOttawa team uses quantum of light to create new quantum simulator (available at: www.uottawa.ca/about-us/media/news/blink-eye-uottawa-team-usesquantum-light-create-new-quantum-simulator) (Accessed 30 January 2023)

[Wol22] Wolf R 2022 *Quantum Computing: Lecture Notes* (The MIT Press)