Date of publication DEC-31, 2022, date of current version NOV-28, 2022
www.computingonline.net/ computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.21.4.2776

Many Known Quantum Algorithms Are Optimal: Symmetry-Based Proofs

VLADIK KREINOVICH[1], OSCAR GALINDO[1], OLGA KOSHELEVA[2]
[1]Department of Computer Science, University of Texas at El Paso, El Paso, TX 79968, USA (e-mails: vladik@utep.edu, ogalindomo@miners.utep.edu)
[2]Department of Teacher Education, University of Texas at El Paso, El Paso, TX 79968, USA (e-mail: olgak@utep.edu)

Corresponding author: Vladik Kreinovich (e-mail: vladik@utep.edu).

This work was supported in part by the National Science Foundation grants 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science), and HRD-1834620 and HRD-2034030 (CAHSI Includes), and by the AT&T Fellowship in Information Technology.

ABSTRACT Many quantum algorithms have been proposed which are drastically more efficient that the best of the non-quantum algorithms for solving the same problems. A natural question is: are these quantum algorithms already optimal – in some reasonable sense – or they can be further improved? In this paper, we review recent results showing that many known quantum algorithms are actually optimal. Several of these results are based on appropriate invariances (symmetries). Specifically, we show that the following algorithms are optimal: Grover's algorithm for fast search in an unsorted array, teleportation algorithm – which is important for parallel quantum computations, and quantum annealing optimization algorithm. This covers many algorithms related to quantum computing. We also mention that algorithms for quantum communication and Deutsch-Josza algorithm – for fast checking whether a bit affect computation results – are optimal. In all these cases, optimality is shown not just for one specific optimality criterion, but for all possible optimality criteria that satisfy the natural invariance requirement.

KEYWORDS quantum computing; optimal algorithms; invariance; symmetry.

I. FORMULATION OF THE PROBLEM

A. NEED FOR QUANTUM COMPUTING

Modern computers are extremely fast, but still there are many practical problem that require even faster computations. For example, high-performance computers, after computing for several hours, help us come up with a reasonably accurate prediction of tomorrow's weather. It turns out that similar algorithms can help us predict where a tornado will turn in the next 15 minutes – but this computation also requires several hours on modern computers, too late for this prediction to be practically useful.

How can we make computer faster? There are many interesting engineering ideas how to do it, but there is also a fundamental limitation – that, according to relativity theory, nothing can travel faster than the speed of light $c = 300000$ km/sec; see, e.g., [9], [33]. For a usual laptop which is about 30 cm in size this means that it takes 10^{-9} seconds – 1 nanosecond – for a signal to go from one side of the laptop to the other. During this time, a usual 4 GHz laptop already performs 4 operations. From this viewpoint, the only way to make computer substantially faster is to make them significantly smaller.

Already in modern computers, each memory cell is very small – up to 10 nanometers (nm), comparable with the nm size of a single molecule. As a result, each cell contains several thousand molecules. If we make cells even smaller, their size will be comparable with the size of a single molecule. At such sizes, we can no longer use Newtonian mechanics, we need to take into account that the micro-world is governed by different equations – the equations of quantum physics [9], [33]. Computing on such a level is known as *quantum computing*

B. NEED FOR QUANTUM ALGORITHMS

One of the important challenges of quantum computing is that in quantum physics – in contrast to Newtonian physics – the results are non-deterministic: we can only predict the probabilities of different outcomes. The classical example of such a probabilistic uncertainty is radioactivity, one of the first observed quantum phenomena: we can predict the

probability that an atom will decay – and thus, accurately predict the amount of radiation – but we cannot predict at which moment of time each individual atom will decay.

Because of this probabilistic uncertainty, we cannot simply use the usual algorithms on the micro-level: we will then, in general, get different results with different probabilities, while in computations, we usually want to come up with a single result. Thus, we need to develop new algorithms.

C. QUANTUM ALGORITHMS: SUCCESSES

Quantum algorithms have indeed been successfully developed for solving all fundamental aspects of computation needs; see, e.g., [26], [35]. Not only the resulting algorithms produce deterministic (or almost deterministic) results, many of them compute these results even faster than the best non-quantum algorithms for solving the same problems.

To briefly describe these successes, let us recall what are the fundamental computation needs. To enumerate these needs, let us recall what we humans want.

- We want to understand how the world works, to predict what will happen – this is, crudely speaking, what science is about. For example, we want to predict where the tornado will turn.
- We also want to understand how can we improve the situation – this is, crudely speaking, what engineering is about. For example, how can we make tornadoes change their course? How can we make houses less vulnerable to tornadoes?
- Finally, we want to communicate – or not – with others, so we need to develop techniques for communication only with the intended folks.

Quantum algorithms are useful in solving all these main problems of science and engineering:

- In the general prediction problem, we need to find a model that fits all the observations. In a usual engineering problem, we need to find a design and/or a control that satisfies a given specification. In most of these problems, once we have a model, a design, or a control, it is computationally feasible to check whether this model, design, etc. satisfies the given specifications, it is searching for a satisfactory model, design, etc. which is computationally intensive. There exists a quantum algorithm that speeds us such a search. An algorithm – proposed by Lev Grover – finds an element in an unsorted list in time $\sqrt{n}$, which is much faster than n steps needed in the non-quantum case [16], [17], [26], [35]. Quantum algorithms are also useful in optimization.
- An additional way to speed up computations comes from the fact that in prediction problems – such as predicting tomorrow's weather – to be on the safe side, we take into account today's meteorological data in all nearby locations, even though most of this data is actually irrelevant. To speed up computations, it is desirable to decide which inputs and relevant and which are not. In this analysis, quantum computing also help – namely, we can use Deutsch-Jozsa algorithm; see, e.g., [26], [35].
- Finally, special algorithms have been developed for quantum communications – which is especially important since it is known that by using quantum computing, we can break the RSA encryption (and similar encryptions) – and these encryptions are behind most of the current computer security techniques [26], [29], [30], [35].

D. QUANTUM ALGORITHM: REMAINING CHALLENGES AND WHAT WE COVER IN THIS PAPER

As we have mentioned, the existing quantum algorithms work very well. However, a next natural question is: are these algorithms optimal – in some reasonable sense – or we can do better? In this paper, we overview several results that show that many quantum algorithms are indeed optimal. These proofs are based on the invariance (symmetry) techniques.

Of course, these results are just the beginning of the study. Quantum computing is a developing field, many new algorithm are being developed all the time, and, as quantum computers will become practical, this will definitely further boost the invention of new algorithms. We hope that the results reviewed in this paper will help researchers to analyze the optimality of other quantum algorithms as well – and in some cases, lead to the discovery of new optimal algorithms.

Comment. In this paper, we summarize results of several papers of ours in which optimality was proven for specific empirically effective quantum algorithms. From this viewpoint, this paper can be viewed as an extended version of several of our previous published papers, in particular, our quantum annealing paper [14].

E. STRUCTURE OF THIS PAPER

We start, in Section 2, with a brief reminder of the quantum basics – basics which are needed to understand the main ideas behind the existing quantum algorithms and behind the proofs of their optimality. In Section 3, we describe the relation between optimality – that we want to prove – and symmetries – i.e., invariance with respect to different transformations. After that, we present the proofs of optimality of different quantum algorithms for quantum data processing: Grover's algorithm in Section 4, parallel-related teleportation algorithm in Section 5, and an optimization-related quantum annealing algorithm in Section 6.

It should be mentioned that other quantum algorithms are also known to be optimal: optimality of Deutsch-Josza algorithm is proven in [20], and optimality of quantum communication algorithm in [15].

II. QUANTUM BASICS

A. QUANTUM STATES

In "classical" (= non-quantum) physics, each object, each system can be in different states s, s', ...In quantum physics, such classical state are denoted by $|s\rangle$, $|s'\rangle$, etc. An unusual feature of quantum physics is that, in addition to such states, we can also have *superpositions* of such states, i.e., states of the type

$$c \cdot |s\rangle + c' \cdot |s'\rangle + \ldots, \tag{1}$$

where c, c', ...are complex numbers for which

$$|c|^2 + |c'|^2 + \ldots = 1, \tag{2}$$

where, as usual, for a complex number $c = a + b \cdot i$, its modulus $|c|$ is defined as $|c| = \sqrt{a^2 + b^2}$. If the system is in the state (1), and we use a classical measurement instrument to measure the state, then:

- we will get state s with probability $|c|^2$,
- we will get state s' with probability $|c'|^2$, etc.

These probabilities should add up to 1, which explains the formula (2).

In particular, a quantum analogue of a *bit* (binary digit) – i.e., of a system that can be in two different states 0 and 1 – is a *quantum bit* (*qubit*, for short) that can be in any state

$$c_0 \cdot |0\rangle + c_1 \cdot |1\rangle, \tag{3}$$

where c_0 and c_1 are complex numbers for which

$$|c_0|^2 + |c_1|^2 = 1. \tag{4}$$

In the state (3), the probability that we will observe 0 is $|c_0|^2$, and the probability that we will observe 1 is equal to $|c_1|^2$.

Similarly, for a 2-bit system – which in classical physics, can be in 4 different states 00, 01, 10, and 11 – a general quantum state is equal to

$$c_{00} \cdot |00\rangle + c_{01} \cdot |01\rangle + c_{10} \cdot |10\rangle + c_{11} \cdot |11\rangle. \tag{5}$$

In principle, we can have general complex numbers. Interestingly, in most quantum algorithms, only real-valued coefficients c, c', ...are used. An explanation of this is provided, e.g., in [2].

B. QUANTUM MEASUREMENTS

In general, if we have n classical states $s_1, \ldots, s_n$, and we want to detect, in a quantum state $\sum_i \alpha_i \cdot |s_i\rangle$, which of these states we are in, we get each s_i with probability $|\alpha_i|^2$ – and once the measurement process detects the state s_i, the actual state turns into s_i.

Instead of the classical states $s_1, \ldots$, we can use any other sequence of states $s'_i = \sum_j t_{ij} \cdot |s_j\rangle$, as long as they are *orthonormal* (= orthogonal and normal) in the sense that:

- for each i, we have $\|s'_i\|^2 = 1$, where $\|s'_i\|^2 \stackrel{def}{=} \sum_j |t_{ij}|^2$ (*normal*), and
- for each $i \neq j$, we have $s'_i \perp s'_{i'}$, i.e., $\langle s'_i | s'_{i'} \rangle = 0$, where $\langle s'_i | s'_{i'} \rangle \stackrel{def}{=} \sum_j t_{ij} \cdot t^*_{i'j}$ (*orthogonal*).

In this case, if we have a state $\sum_i \alpha'_i \cdot |s'_i\rangle$, then with probability $|\alpha'_i|^2$, the measurement result is s'_i and the state turns into s'_i.

In general, instead of a sequence of orthogonal vectors, we can have a sequence of orthogonal linear spaces L_1, L_2, ...– where $L_i \perp L_j$ means that $s_i \in L_i$ and $s_j \in L_j$ implies $s_i \perp s_j$. In this case, every state s can be represented as a sum $s = \sum_i s_i$ of the vectors $s_i \in L_i$. As a result of the measurement, with probability $\|s_i\|^2$, we conclude that the state is in the space L_i, and the original state turns into a new state $s_i/\|s_i\|$.

C. COMPOSITE SYSTEMS

A 2-bit system is the simplest example of a *composite system*, when we consider two independent subsystems as a single system. In classical physics, if the first system is in one of the states s, s', ..., and the second system is in one of the states t, t', ..., then the set of all possible states of the composite system is the set of all the pairs (s, t) – which is also known as a *Cartesian product* $S \times T$ of the set $S = \{s, s', \ldots\}$ of possible states of the first system and the set $T = \{t, t', \ldots\}$ of possible states of the second system.

In quantum physics, if the first system was in the general quantum state (1) and the second system is in a similar quantum state

$$a \cdot |t\rangle + a' \cdot |t'\rangle + \ldots, \tag{6}$$

then the state of the composite system – known as the *tensor product* of the states (1) and (6):

$$(c \cdot |s\rangle + c' \cdot |s'\rangle + \ldots) \otimes (a \cdot |t\rangle + a' \cdot |t'\rangle + \ldots), \tag{7}$$

is equal to

$$c \cdot a \cdot |s, t\rangle + c \cdot a' \cdot |s, t'\rangle + \ldots + c' \cdot a \cdot |s', t\rangle + c' \cdot a' \cdot |s', t'\rangle + \ldots \tag{8}$$

In particular, for classical states, e.g., when $c = a = 1$ and $c' = \ldots = a' = \ldots = 0$, we get $|s\rangle \otimes |t\rangle = |s, t\rangle$.

Comment. It should be mentioned that the transformation of two states of subsystems into a single state of a composite system is linear in each of the values c, c', ..., and a, a', ...This linearity comes from the need to make sure that for the independent subsystems, the probability of observing (s, t) is equal to the product of the probabilities of observing s and t. This is true for the formula (8), when this equality follows from the fact that for every two complex numbers c and a, we have $|c \cdot a|^2 = |c|^2 \cdot |a|^2$.

D. HOW QUANTUM STATES CHANGE

States may change with time. In quantum physics, all changes are linear – for the same reason why composition of two states is linear. In other words, each state

$$c_1 \cdot |s_1\rangle + \ldots + c_n \cdot |s_n\rangle \tag{9}$$

is transformed into the state

$$c'_1 \ \ j s_1 i + \ldots + c'_n \ \ j s_n i, \tag{10}$$

for which

$$c'_i = \sum_{j=1} T_{ij} \ \ c_j$$

for some coefficient T_{ij}. The matrix $T = kT_{ij}k$ is *unitary*: $T^\dagger T = TT^\dagger = I$, where I is the unit matrix, and $T^\dagger_{ij} \stackrel{def}{=} T^*_{ji}$, where c^* denotes *complex conjugate*:

$$(a+b \ \ i)^* \stackrel{def}{=} a \ \ \ b \ \ i.$$

Note that every such transformation is reversible: once we apply the transformation T, we can then apply the transformation $T^\dagger$ and, due to the property $T^\dagger T = I$, get back the original state.

For 1-qubit systems, one of such transformation is *Hadamard transformation* H for which

$$H(j0i) = j0'i \stackrel{def}{=} \frac{1}{2} \ \ j0i + \frac{1}{2} \ \ j1i;$$

$$H(j1i) = j0'i \stackrel{def}{=} \frac{1}{2} \ \ j0i \ \ \ \frac{1}{2} \ \ j1i. \tag{11}$$

E. HOW FUNCTIONS ARE REPRESENTED IN QUANTUM ALGORITHMS

In this section, we will deal only with functions $y = f(x_1, \ldots, x_n)$ of boolean (0-1) variables – since these are the basic functions implemented by different "gates", of which computers are built. We cannot simple represent these functions as transforming n boolean values x_i into a single boolean value y, since such transformation is, in general, irreversible. For example, for the "and"-function $y = f(x_1, x_2) = x_1 \,\&\, x_2$, if we know that $y = 0$, we cannot uniquely reconstruct the original pair (x_1, x_2):

- we could have $(x_1, x_2) = (0, 0)$,
- we could have $(x_1, x_2) = (0, 1)$, or
- we could have $(x_1, x_2) = (1, 0)$.

To make the corresponding transformation reversible, a function $y = f(x_1, \ldots, x_n)$ is represented as

$$T_f(x_1, \ldots, x_n, y) = (x_1, \ldots, x_n, y \ \ \ f(x_1, \ldots, x_n)), \tag{12}$$

where $a \ \ \ b$ is exclusive "or" – or, what is the same, addition modulo 2, an operation for which $0 \ \ \ 0 = 1 \ \ \ 1 = 0$ and $0 \ \ \ 1 = 1 \ \ \ 0 = 1$. One can check that thus defined transformation is reversible: namely, if we apply the transformation T_f twice, we get back the original state $(x_1, \ldots, x_n, y)$ – simply because $a \ \ \ a = 0$ for all a.

Comment. While this is the prevailing representation of functions in quantum computing, it should be mentioned in some cases, a different representation is preferable; see, e.g., [11].

III. RELATION BETWEEN OPTIMALITY AND INVARIANCE (SYMMETRY)

A. WHAT IS INVARIANCE (SYMMETRY)

In many cases, there are some natural transformations that does not change the system. This "not changing" is called *invariance*. For example, suppose that we have an unsorted list, and we are looking for an element with a ceratin property in this list. For convenience, we can denote one of the list's elements by s_1, another one by s_2, etc., but in this problem, it does not matter which element is called s_1, which s_2, etc. – any permutation

$$\pi : f1, \ldots, ng \ \ ! \ \ f1, \ldots, ng$$

would retain the problem. Thus, in this problems, permutations are invariances. In other problems, we will have other natural invariances.

In physics, invariance is called *symmetry* – since it naturally generalizes geometric invariances (symmetries); see, e.g., [9], [33].

IV. WHAT DOES "OPTIMAL" MEAN

Usually, when we talk about "optimal", we mean that on the set of all alternatives A, A', ..., there is an *objective function* describing the quality of different alternatives, and we are looking for the alternative A with the largest (or sometimes the smallest) value of this objective function. For example, when we select between different quantum – i.e., in general, probabilistic – algorithms for solving a given problem, we may want to maximize that probability $f(A)$ that the algorithm A will lead to the desired solution.

However, this is a somewhat simplified description of what we usually mean by optimality. Often, there are several alternatives A_1, A_2, etc., with the exact same largest value $f(A_1) = f(A_2) = \ldots$ of the objective function. In this case, we can use this non-uniqueness to optimize something else. For example, in the above case, we can minimize the average time needed for the algorithm to finish. This means, in effect, that the original optimality criterion is not final, we can modify it and come up with a new, more complex criterion according to which an alternative A is better than the alternative A' – we will denote it by $A < A'$– if:

- either $f(A') < f(A)$
- or we have $f(A) = f(A')$ and also $g(A') < g(A)$ for the additional objective function $g(A)$.

If this still leads to several equally optimal alternatives, we can use this non-uniqueness to optimize something else – until we get to the *final* optimality criterion, for which there is exactly one optimal alternative.

To simplify the analysis, it is useful to ignore all these objective functions $f(A)$, $g(A)$, etc., and to only consider what really matters: for each pair (A, A'), according to this criterion:

- when we have A' better than A $(A < A')$,
- when we have A better than A' $(A' < A)$, and
- when A and A' are equally good; we will denote it by $A \ \ \ A'$.

In precise terms, by an *optimality criterion* that the set A of all alternatives, we mean a pair of relations $<$ and with the following natural properties:

- if $A < A'$ and $A' < A''$, then $A < A''$;
- if $A < A'$ and A' A'', then $A < A''$;
- if A A' and $A' < A''$, then $A < A''$;
- if A A' and A' A'', then A A'';
- if A A', then A' A; and
- if $A < A'$, then we cannot have A A'.

This pair of relations is known as a *pre-order*: it is similar to order, with the main difference that we can have A A' without having $A = A'$.

An alternative A_{opt} is called *optimal* if for every other alternative A, we have $A < A_{opt}$ or A A_{opt}. An optimality criterion is called *final* if there is exactly one alternative which is optimal with respect to this criterion.

A. FOR INVARIANT CRITERIA, OPTIMAL ALTERNATIVE IS ALSO INVARIANT

In many cases, there exists a reversible transformation T : A ! A – e.g., permutation – which does not change the situation. In this case, it makes sense to require that this transformation will not change which alternative is better. In precise terms, we say that an optimality criterion is T-*invariant* if the following conditions are satisfied:

- if $A < A'$, then $T(A) < T(A')$;
- if A A' then $T(A)$ $T(A')$.

Many results from this paper used the following lemma (see, e.g., [25]):

Lemma. *For every final T-invariant optimality criterion, its optimal alternative A_{opt} is also T-invariant, i.e.,*

$$T(A_{opt}) = A_{opt}.$$

Proof. The fact that A_{opt} means that for every A 2 A, we have:

- either $A < A_{opt}$
- or A A_{opt}.

In particular, this is true for $T^{-1}(A)$, i.e.:

- either $T^{-1}(A) < A_{opt}$
- or $T^{-1}(A)$ A_{opt}.

Due to T-invariance, we can conclude that:

- either $A < T(A_{opt})$
- or A $T(A_{opt})$.

This is true for every alternative A, which means that the alternative $T(A_{opt})$ is also optimal. However, the optimality criterion is final, which means that there is only one optimal criterion. Thus, indeed, $T(A_{opt}) = A_{opt}$. The lemma is proven.

Due to this lemma, if the optimality criterion is T-invariant, then to find optimal alternative, it is sufficient to find T-invariant alternatives. Let us start checking optimality with Grover's algorithm.

V. GROVER'S ALGORITHM IS OPTIMAL

A. FORMULATION OF THE PROBLEM

We are solving the following problem. We have a list of elements $e_1, \ldots, e_n$. We have an algorithm $f(i)$ that, given an element e_i – i.e., in effect, the index i – checks whether this element has the desired property. We want to find an element that has this property. For simplicity, we will consider the case when there is exactly one such element i_0.

Let us describe this problem is quantum computing-related terms. What we want is an index i_0 of the desired element. In quantum computing terms, this means that we want to end up in a state $|i_0\rangle$. As we have mentioned, in general, quantum processes are probabilistic, so instead of the exact state $|i_0\rangle$, we may end up in a superposition state:

$$c_1 \ |1\rangle + \ldots + c_{i_0-1} \ |i_0\ \ 1\rangle + c_{i_0} \ |i_0\rangle +$$

$$c_{i_0+1} \ |i_0 + 1\rangle + \ldots + c_n \ |n\rangle. \qquad (13)$$

In quantum terms, the algorithm that checks whether a given element has the desired property has the form $T_f(|i, y\rangle) = |i, y \ \ f(x)\rangle$, i.e.:

- for i 6= i_0, we have

$$T_f(|i, 0\rangle) = |i, 0\rangle \text{ and } T_f(|i, 1\rangle) = |i, 1\rangle,$$

  while
- for $i = i_0$, we have

$$T_f(|i_0, 0\rangle) = |i_0, 1\rangle \text{ and } T_f(|i_0, 1\rangle) = |i_0, 0\rangle.$$

In terms of the Hadamard states $|0'\rangle$ and $|1'\rangle$, we get the following:

- for $|0'\rangle$, for all i, we have $T_f(|i\rangle \ \ |0'\rangle) = |i\rangle \ \ |0'\rangle$;
- for $|1'\rangle$, for all i 6= i_0, we have

$$T_f(|i\rangle \ \ |1'\rangle) = |i\rangle \ \ |1'\rangle,$$

  while for $i = i_0$, we have

$$T_f(|i_0\rangle \ \ |1'\rangle) = \ \ |i_0\rangle \ \ |1'\rangle.$$

So, for $|0'\rangle$, nothing changes, and for $|1'\rangle$, the additional bit $|1'\rangle$ remains the same, but the previous state (13) changes to:

$$c_1 \ |1\rangle + \ldots + c_{i_0-1} \ |i_0\ \ 1\rangle \ \ c_{i_0} \ |i_0\rangle +$$

$$c_{i_0+1} \ |i_0 + 1\rangle + \ldots + c_n \ |n\rangle. \qquad (14)$$

Let us denote this transformation from (13) to (14) by U.

Our goal is to start with some state, and, by applying this transformation U and some other transformation(s) S, eventually come up with the desired element i_0.

B. INVARIANCE (SYMMETRY): REMINDER

As we have mentioned earlier, in this problem, the natural invariances (symmetries) are invariances with respect to all possible permutations $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$. It is therefore reasonable to require that our optimality criterion is invariant with respect to all permutations. Due to the above Lemma, this implies that the optimal algorithm should also be permutation-invariant, in particular:

- that the initial state should be permutation-invariant, and
- that all transformations S should be permutation-invariant.

C. TOWARDS THE OPTIMAL ALGORITHM: WHICH TRANSFORMATIONS ARE PERMUTATION-INVARIANT?

The fact that the initial state is permutation-invariant means that $c_i = c_{i'}$ for all i and i' – since every two indices i and i' can be obtained from each other by an appropriate permutation. Thus, the initial state must have the form

$$c_1 \cdot |1\rangle + \ldots + c_1 \cdot |n\rangle, \tag{14}$$

for some c_1. Due to the normalization requirement (2), we have $|c_1| = 1/\sqrt{n}$. In quantum mechanics, states differing by a constant are considered the same state, so we can simply take $c_1 = 1/\sqrt{n}$. Then the initial state takes the form:

$$\frac{1}{\sqrt{n}} \cdot |1\rangle + \ldots + \frac{1}{\sqrt{n}} \cdot |n\rangle. \tag{15}$$

This is exactly the initial state of Grover's algorithm.

A general transformation is describes by a matrix S_{ij}. For this matrix, permutation invariance means that all the elements S_{ii} are equal to each other – similar argument as before. Let us denote this common value by a. Similarly, all the elements S_{ij} with $i \neq j$ should also be equal to each other. Let us denote this common value by b. In these terms, the corresponding linear transformation transforms the vector c_i into a new vector

$$c_i' = a \cdot c_i + b \cdot \sum_{j \neq i} c_j. \tag{16}$$

This expression can be equivalently described as

$$c_i' = (a - b) \cdot c_i + b \cdot C, \text{ where } C \stackrel{def}{=} \sum_{j=1}^{n} c_j. \tag{17}$$

We want to make sure that this transformation preserved the fact that the probabilities add up to 1, i.e., that

$$\sum_{j=1}^{n} |c_j'|^2 = 1. \tag{18}$$

As we have mentioned earlier, it is sufficient to consider situations in which all the coefficients c_i' are real numbers. In this case, $|c_j'|^2 = (c_j')^2$, and, due to (17), the condition (18) takes the form

$$(a - b)^2 \cdot \left(\sum_{i=1}^{n} c_i^2 \right) + 2 \cdot (a - b) \cdot b \cdot C^2 + n \cdot b^2 \cdot C^2 = 1, \tag{19}$$

i.e., due to the fact that $\sum_{i=1}^{n} c_i^2 = 1$, that

$$(a - b)^2 + (2 \cdot (a - b) \cdot b + n \cdot b^2) \cdot C^2 = 1. \tag{20}$$

This equality has to hold for all C, so we must have

$$2 \cdot (a - b) \cdot b + n \cdot b^2 = (2 \cdot (a - b) + n \cdot b) \cdot b = 0. \tag{21}$$

If $b = 0$, then $a = -1$, so the transformation S either leaves the state unchanged or multiplies all the coefficient c_i by -1 – i.e., in effect, also leaves the state unchanged. So, to get a non-trivial transformation, we need to take $b \neq 0$. In this case, $2 \cdot (a - b) + n \cdot b = 0$; since, without losing generality, we can take $a - b = 1$, we get $b = -2/n$. Thus, $a = (a - b) + b = 1 - 2/n$, and this transformation takes the form

$$c_i' = -1 \cdot \frac{2}{n} \cdot c_i - \frac{2}{n} \cdot \sum_{j \neq i} c_j.$$

This is also exactly the transformation used in Grover's algorithm!

In what order shall we apply the algorithms U and S? If we apply U twice or S twice, we get back the same state. Thus, it makes sense to apply these two algorithms interchangingly. The first application should be of U, since if we apply S to the initial state, we get the same state multiplied by a constant. Thus, we arrive at the following algorithm:

- we start with the initial state (15);
- then, we apply the transformation U;
- after that, we apply the transformation S;
- then, again we apply U; etc.

This is exactly Grover's algorithm. Thus, the Grover's algorithm is the only permutation-invariant one. And since the optimal algorithm must be permutation-invariant, we therefore conclude that Grover's algorithm is optimal.

VI. PARALLELIZATION: TELEPORTATION ALGORITHM IS OPTIMAL

A. NEED FOR PARALLELIZATION

From the theoretical viewpoint, the fact that, e.g., Grover's algorithm is optimal is interesting. However, from the practical viewpoint, the fact that we cannot improve this algorithm constitutes a limitation on how fast we can compute – even if we use quantum computing. In problems in which the Grover's speed up is not sufficient, we need to use other ideas to achieve a further speedup.

To further speed up computations, a natural idea is to have several quantum computers working in parallel, so that each of them solves a part of the problem. This idea is similar to how we humans solve complex problems: if a task is too difficult for one person to solve – be it building a big house or proving a complex theorem – several people team up and together solve the task.

B. NEED FOR TELEPORTATION

To successfully collaborate, quantum computers need to exchange intermediate states of their computations. Here lies a problem: for complex problems, we would like to use computers located in different geographic areas, but a quantum state gets changed when it is sent far away.

Researchers have come up with a way to avoid this sending, called *teleportation*. There exists a scheme for teleportation [3], [26], [27].

C. WHAT WE DO IN THIS SECTION

A priori, it is not clear how good is the current teleportation scheme: maybe there are other schemes which are faster (or better in some other sense)? In this section, we show that the existing teleportation scheme is, in some reasonable sense, unique – and, in this sense, is the best. This result first appeared in [12].

D. STANDARD QUANTUM TELEPORTATION ALGORITHM: REMINDER

1) Need for communication

At one location, we have a particle in a certain state; we want to send this state to some other location.

Usually, the sender is denoted by A and the receiver by B. In communications, it is common to call the sender Alice, and to call the receiver Bob. States corresponding to Alice are usually described by using a subscript A, and states corresponding to Bob are usually described by using a subscript B.

2) Communication is straightforward in classical physics but a challenge in quantum physics

In classical (pre-quantum) physics, the communication problem has a straightforward solution: if we want to communicate a state, we measure all possible characteristics of this state, send these values to Bob, and let Bob reproduce the object with these characteristics. This is how, e.g., 3D printing works. This solution is based on the fact that in classical (non-quantum) physics we can, in principle, measure all characteristic of a system without changing it.

The problem is that in quantum physics, such a straightforward approach is not possible: as we have mentioned, in quantum physics, every measurement changes the state – and moreover, irreversibly deletes some information about the state. For example, if we start with a state $\alpha_0\ |0\rangle+\alpha_1\ |1\rangle$, all we get after the measurement is either 0 or 1, with no way to reconstruct the values α_0 and α_1 that characterize the original state. Since we cannot use the usual straightforward approach for communicating a state, we need to use an indirect approach. This approach is known as *teleportation*.

3) What we consider in this section

In this section, we consider the simplest possible quantum state – namely, the quantum analogue of the simplest possible non-quantum state. In the non-quantum case, a system can be in several different states. The state passing problem makes sense only when the system can be in at least two different states – otherwise, if we know beforehand what state we want to send, there is no need to send any information, Bob can simply reproduce the known state. The simplest case when communication is needed is when the number of possible states is as small as possible but still larger than 1 – i.e., the case when the system can be in two different states. In the computer, such situation can be naturally described if we associate these two possible states with 0 and 1.

In these terms, the problem is as follows:

- Alice has a state

$$\alpha_0\ |0\rangle + \alpha_1\ |1\rangle \tag{22}$$

  that she wants to communicate to Bob – a person at a different location.
- As a result of this process, Bob should have the same state.

4) Notations

Let us indicate states corresponding to Alice with a subscript A, and states corresponding to Bob with a subscript B. The state (22) is not exclusively Alice's and it is not exclusively Bob's, so to describe this state, we will use the next letter – letter C. In these terms, Alice has a state

$$\alpha_0\ |0\rangle_C + \alpha_1\ |1\rangle_C \tag{23}$$

that she wants to communicate to Bob.

5) Preparing for teleportation: an entangled state

To make teleportation possible, Alice and Bob prepare a special *entangled* state:

$$\frac{1}{\sqrt{2}}\ |0_A 1_B\rangle + \frac{1}{\sqrt{2}}\ |1_A 0_B\rangle. \tag{24}$$

This state is a superposition of two classical states:

- the state $0_A 1_B$ in which A is in state 0 and B is in state 1, and
- the state $1_A 0_B$ in which A is in state 1 and B is in state 0.

6) What is the joint state of A, B, and C at the beginning of the procedure

In the beginning, the state C is independent of A and B. So, the joint state is a tensor product of the AB-state (24) and the C-state (23):

$$\frac{\alpha_0}{\sqrt{2}}\ |0_A 1_B 0_C\rangle + \frac{\alpha_1}{\sqrt{2}}\ |0_A 1_B 1_C\rangle +$$

$$\frac{\alpha_0}{\sqrt{2}}\ |1_A 0_B 0_C\rangle + \frac{\alpha_1}{\sqrt{2}}\ |1_A 0_B 1_C\rangle. \tag{25}$$

7) First stage: measurement

In the first stage of the standard teleportation algorithm, Alice performs a measurement procedure on the parts A and C which are available to her. In general, to describe the possible results of measuring a state s with respect to linear spaces L_i, we need to represent s as the sum

$$s = \sum_i s_i, \tag{26}$$

with $s_i \in L_i$.

In the standard teleportation algorithm, we perform the measurement with respect to the following four linear spaces $L_i = L_B \otimes t_i$, where L_B is the set of all possible linear combinations of $|0\rangle_B$ and $|1\rangle_B$, and the states t_i have the following form:

$$\begin{aligned}
t_1 &= \frac{1}{\sqrt{2}}\,|0\rangle_A|0\rangle_C + \frac{1}{\sqrt{2}}\,|1\rangle_A|1\rangle_C;\\
t_2 &= \frac{1}{\sqrt{2}}\,|0\rangle_A|0\rangle_C - \frac{1}{\sqrt{2}}\,|1\rangle_A|1\rangle_C;\\
t_3 &= \frac{1}{\sqrt{2}}\,|0\rangle_A|1\rangle_C + \frac{1}{\sqrt{2}}\,|1\rangle_A|0\rangle_C;\\
t_4 &= \frac{1}{\sqrt{2}}\,|0\rangle_A|1\rangle_C - \frac{1}{\sqrt{2}}\,|1\rangle_A|0\rangle_C.
\end{aligned} \tag{27}$$

One can easily check that the states t_i are orthonormal, hence the spaces L_i are orthogonal.

To describe the result of measuring the state (25) with respect to these linear spaces, we must first represent the state (25) in the form $s = \sum_i s_i$, with $s_i \in L_i$. For this purpose, we can use the fact that, due to the formulas (27), we have

$$\begin{aligned}
|0\rangle_A|0\rangle_C &= \frac{1}{\sqrt{2}}\,t_1 + \frac{1}{\sqrt{2}}\,t_2;\\
|1\rangle_A|1\rangle_C &= \frac{1}{\sqrt{2}}\,t_1 - \frac{1}{\sqrt{2}}\,t_2;\\
|0\rangle_A|1\rangle_C &= \frac{1}{\sqrt{2}}\,t_3 + \frac{1}{\sqrt{2}}\,t_4;\\
|1\rangle_A|0\rangle_C &= \frac{1}{\sqrt{2}}\,t_3 - \frac{1}{\sqrt{2}}\,t_4.
\end{aligned} \tag{28}$$

Substituting the expressions (28) into the formula (25), we get

$$\begin{aligned}
&\frac{\alpha_0}{\sqrt{2}}\,|1\rangle_B \otimes \frac{1}{\sqrt{2}}\,t_1 + \frac{1}{\sqrt{2}}\,t_2 \;+\\
&\frac{\alpha_1}{\sqrt{2}}\,|1\rangle_B \otimes \frac{1}{\sqrt{2}}\,t_3 + \frac{1}{\sqrt{2}}\,t_4 \;+\\
&\frac{\alpha_0}{\sqrt{2}}\,|0\rangle_B \otimes \frac{1}{\sqrt{2}}\,t_3 - \frac{1}{\sqrt{2}}\,t_4 \;+\\
&\frac{\alpha_1}{\sqrt{2}}\,|0\rangle_B \otimes \frac{1}{\sqrt{2}}\,t_1 - \frac{1}{\sqrt{2}}\,t_2\;,
\end{aligned}$$

thus

$$\frac{\alpha_0}{2}|1\rangle_B|1\rangle + \frac{\alpha_1}{2}|0\rangle_B|1\rangle \otimes t_1 + \frac{\alpha_0}{2}|1\rangle_B|1\rangle - \frac{\alpha_1}{2}|0\rangle_B|1\rangle \otimes t_2 +$$

$$\frac{\alpha_1}{2}|1\rangle_B|1\rangle + \frac{\alpha_0}{2}|0\rangle_B|1\rangle \otimes t_3 + \frac{\alpha_1}{2}|1\rangle_B|1\rangle - \frac{\alpha_0}{2}|0\rangle_B|1\rangle \otimes t_4.$$

So, we get a representation of the type (26), with

$$\begin{aligned}
s_1 &= \frac{\alpha_0}{2}\,|1\rangle_B|1\rangle + \frac{\alpha_1}{2}|0\rangle_B|1\rangle \otimes t_1,\\
s_2 &= \frac{\alpha_0}{2}\,|1\rangle_B|1\rangle - \frac{\alpha_1}{2}\,|0\rangle_B|1\rangle \otimes t_2,\\
s_3 &= \frac{\alpha_1}{2}\,|1\rangle_B|1\rangle + \frac{\alpha_0}{2}\,|0\rangle_B|1\rangle \otimes t_3,\\
s_4 &= \frac{\alpha_1}{2}\,|1\rangle_B|1\rangle - \frac{\alpha_0}{2}\,|0\rangle_B|1\rangle \otimes t_4.
\end{aligned}$$

Here, for each i, we have

$$\|s_i\|^2 = \left(\frac{\alpha_0}{2}\right)^2 + \left(\frac{\alpha_1}{2}\right)^2 = \frac{1}{4}\,(|\alpha_0|^2 + |\alpha_1|^2) = \frac{1}{4},$$

thus $\|s_i\| = \frac{1}{2}$.

So, with equal probability of $\frac{1}{4}$, we get one of the following four states – and Alice knows which one it is:

$$\begin{aligned}
&(\alpha_0\,|1\rangle_B|1\rangle + \alpha_1\,|0\rangle_B|1\rangle) \otimes t_1;\\
&(\alpha_0\,|1\rangle_B|1\rangle - \alpha_1\,|0\rangle_B|1\rangle) \otimes t_2;\\
&(\alpha_1\,|1\rangle_B|1\rangle + \alpha_0\,|0\rangle_B|1\rangle) \otimes t_3;\\
&(\alpha_1\,|1\rangle_B|1\rangle - \alpha_0\,|0\rangle_B|1\rangle) \otimes t_4.
\end{aligned} \tag{29}$$

8) Second stage: communication

On the second stage, Alice sends to Bob the measurement result. As a result, Bob knows in which the four states (29) the system is.

9) Final stage: Bob "rotates" his state and thus, get the original state teleported to him

On the final stage, Bob performs an appropriate transformation of his state B.

- In the first case, he uses a unitary transformation that swaps $|0\rangle_B$ and $|1\rangle_B$, for which $t_{01} = t_{10} = 1$ and $t_{00} = t_{11} = 0$.
- In the second case, he uses a unitary transformation for which
$$t_{01} = 1,\ t_{10} = -1 \text{ and } t_{00} = t_{11} = 0.$$
- In the third case, he already has the desired state.
- In the fourth case, he uses a unitary transformation for which $t_{00} = -1$, $t_{11} = 1$, and $t_{01} = t_{10} = 0$.

As a result, in all fours cases, he gets the original state

$$\alpha_0\,|0\rangle_B + \alpha_1\,|1\rangle_B.$$

E. THE MAIN RESULT OF THIS SECTION: THE STANDARD QUANTUM TELEPORTATION ALGORITHM IS, IN SOME REASONABLE SENSE, UNIQUE

1) Formulation of the problem

Teleportation is possible because we have prepared an *entangled* state (24), i.e., a state s_{AB} in which the states of Alice and Bob are not independent, i.e., a state that does not have a form $s_A \quad s_B$. However, (24) is not the only possible entangled state. Let us consider, instead, a general joint state of two qubits:

$$a_{00} \quad |0_A 0_B\rangle + a_{01} \quad |0_A 1_B\rangle + a_{10} \quad |1_A 0_B\rangle + a_{11} \quad |1_A 1_B\rangle. \tag{24a}$$

What will happen if we use this more general entangled state instead of the one that is used in the known teleportation algorithm?

2) Analysis of the problem

For the state (24a), the joint state of all three subsystems has the form

$$\begin{aligned} &\alpha_0 \quad a_{00} \quad |0_A 0_B 0_C\rangle + \alpha_1 \quad a_{00} \quad |0_A 0_B 1_C\rangle + \\ &\alpha_0 \quad a_{01} \quad |0_A 1_B 0_C\rangle + \alpha_1 \quad a_{01} \quad |0_A 1_B 1_C\rangle + \\ &\alpha_0 \quad a_{10} \quad |1_A 0_B 0_C\rangle + \alpha_1 \quad a_{10} \quad |1_A 0_B 1_C\rangle + \\ &\alpha_0 \quad a_{11} \quad |1_A 1_B 0_C\rangle + \alpha_1 \quad a_{11} \quad |1_A 1_B 1_C\rangle. \end{aligned} \tag{25a}$$

Substituting expressions (28) into this formula, we get

$$\begin{aligned} &\frac{\alpha_0}{\sqrt{2}} \quad a_{00} \quad |0_B\rangle \quad (t_1 + t_2) + \\ &\frac{\alpha_1}{\sqrt{2}} \quad a_{00} \quad |0_B\rangle \quad (t_3 + t_4) + \\ &\frac{\alpha_0}{\sqrt{2}} \quad a_{01} \quad |1_B\rangle \quad (t_1 + t_2) + \\ &\frac{\alpha_1}{\sqrt{2}} \quad a_{01} \quad |1_B\rangle \quad (t_3 + t_4) + \\ &\frac{\alpha_0}{\sqrt{2}} \quad a_{10} \quad |0_B\rangle \quad (t_3 \quad t_4) + \\ &\frac{\alpha_1}{\sqrt{2}} \quad a_{10} \quad |0_B\rangle \quad (t_1 \quad t_2) + \\ &\frac{\alpha_0}{\sqrt{2}} \quad a_{11} \quad |1_B\rangle \quad (t_3 \quad t_4) + \\ &\frac{\alpha_1}{\sqrt{2}} \quad a_{11} \quad |1_B\rangle \quad (t_1 \quad t_2), \end{aligned}$$

thus $s = S_1 \quad t_1 + S_2 \quad t_2 + \ldots$, where

$$S_1 = \frac{\alpha_0}{\sqrt{2}} \frac{a_{00}}{\sqrt{2}} + \frac{\alpha_1}{\sqrt{2}} \frac{a_{10}}{\sqrt{2}} \quad |0_B\rangle + \frac{\alpha_0}{\sqrt{2}} \frac{a_{01}}{\sqrt{2}} + \frac{\alpha_1}{\sqrt{2}} \frac{a_{11}}{\sqrt{2}} \quad |1_B\rangle,$$

and $S_2, \ldots$ are described by similar expressions.

This means that after the measurement, Bob will have the normalized state $S_1/\|S_1\|$. To perform teleportation, we need to transform this state into the original state $\alpha_0 \quad |0_B\rangle + \alpha_1 \quad |1_B\rangle$. Thus, the transformation from the resulting state $S_1/\|S_1\|$ to the original state must be unitary. It is known that the inverse transformation to a unitary one is also unitary. In general, a unitary transformation transforms orthonormal states into orthonormal ones.

So, the inverse transformation that:

- maps the state $|0_B\rangle$ (corresponding to $\alpha_0 = 1$ and $\alpha_1 = 0$) into a new state

$$|1'_B\rangle \stackrel{def}{=} \text{const} \quad (a_{00} \quad |0_B\rangle + a_{01} \quad |1_B\rangle),$$

and

- maps the state $|1_B\rangle$ (corresponding to $\alpha_0 = 0$ and $\alpha_1 = 1$) into a new state

$$|0'_B\rangle \stackrel{def}{=} \text{const} \quad (a_{10} \quad |0_B\rangle + a_{11} \quad |1_B\rangle),$$

transforms two original orthonormal vectors $|0_B\rangle$ and $|1_B\rangle$ into two new orthonormal ones $|0'_B\rangle$ and $|1'_B\rangle$.

In terms of these new states, the entangled state (24a) takes the form

$$\text{const} \quad (|0_A\rangle \quad |1'_B\rangle + |1_B\rangle \quad |0'_B\rangle).$$

From the requirement that the sum of the squares of absolute values of all the coefficients add up to 1, we conclude that $2 \quad \text{const}^2 = 1$. Then $\text{const} = \frac{1}{\sqrt{2}}$ and the entangled state takes the familiar form

$$\frac{1}{\sqrt{2}} \quad (|0_A\rangle \quad |1'_B\rangle + |1_B\rangle \quad |0'_B\rangle). \tag{24}$$

This is exactly the entangled state used in the standard teleportation algorithm. So, we can make the following conclusion.

3) Conclusion of this section

From the technical viewpoint, the only entangled state that leads to a successful teleportation is the state (24) corresponding to the standard quantum teleportation algorithm – for some orthornomal states $|0'_B\rangle$ and $|1'_B\rangle$.

Thus, we have shown that, indeed, the existing quantum teleportation algorithm is unique – so we should not waste our time and effort looking for more efficient alternative quantum teleportation algorithms.

VII. OPTIMIZATION: QUANTUM ANNEALING SCHEDULES ARE OPTIMAL

A. QUANTUM ANNEALING: IDEAS, SUCCESSES, AND CHALLENGES

One of the important practical problems is optimization. An important challenge is that often, the existing optimization techniques lead to a local optimum. One way to avoid local optima is *annealing*: whenever we find ourselves in a possibly local optimum, we jump out with some probability and continue search for the true optimum. Since quantum processes are probabilistic, a natural way to organize such a probabilistic perturbation of the deterministic optimization

is to use quantum effects, i.e., to perform quantum annealing. This idea was first proposed in [10], [19] and has been used successfully since then.

It turns out that often, quantum annealing works much better than non-quantum one; see, e.g., see, e.g., [4]–[6], [8], [18], [21], [22], [24], [28], [31], [32], [34]. Quantum annealing is the main technique behind the only commercially available computational devices that use quantum effects – D-Wave computers; see, e.g., [4], [21], [32].

The efficiency of quantum annealing depends on the proper selection of the annealing schedule, i.e., schedule that describes how the perturbations decrease with time. Researchers have found that empirically, the following two schedules work best: power law and exponential ones [7], [22], [23]. In this section, following [14], we describe the method and corresponding schedules in some detail, and prove that these two schedules are indeed optimal (in some reasonable sense).

Comment. It is important to emphasize that in this section – as well as in other sections of this paper – we are not proposing a new method, we are providing a theoretical explanation for the empirical effectiveness of previously proposed methods.

B. FORMULATION OF THE PROBLEM AND THE PHYSICAL MEANING OF ANNEALING

Traditional computers operate in discrete time. Specifically, data processing in traditional computers is performed by elementary processing units (called gates):

- there are “and”-gates that perform the logical “and”-operation $a \,\&\, b$,
- there are “or”-gates that perform logical “or”-operation $a \vee b$,
- there are “nand”-gates that, given two bits a and b, compute: $\neg(a \,\&\, b)$, etc.

Most computer algorithms consist of a clearly defined sequence of such discrete-time steps.

Many quantum algorithms – including the algorithms that we analyzed in the previous sections – are like that: the state of the system in the next moment of time is determined by its state at the previous moment of time.

However, there are also other algorithms that simulate continuous-time processes. For example, algorithms for simulating a plant – e.g., digital twins – are like that. Many other algorithms are like that, algorithms in which simulation is not the ultimate objective as for digital twins, it is a way to achieve some other objective. For example, many optimization algorithms have this form – e.g., gradient descent, the most well known optimization technique.

Of course, in a computer, each such algorithm is implemented in discrete time anyway, but, in contrast to effectively discrete-time algorithms like search and sorting, in many continuous-time algorithms, the selection of the time step barely affects the computation result. For example, if we want to simulate a plant, and we get a reasonable picture by re-computing its state every hour, we can get a slightly more accurate picture if we instead simulate this plant minute-by-minute, but the change will be relatively small.

So, when we analyze such systems, then, in the first approximation, we can safely ignore the existence of a discrete time step – provided, of course, that this time step is sufficiently small – and consider the algorithm as actually operating in continuous time. To analyze quantum analogues of such algorithms, we need to be able to describe how the quantum state changes in continuous time. We have already mentioned that at any given moment of time, the quantum state can be described as a complex-valued linear combination $\sum_i s_i$. In quantum physics, the dynamics of a quantum state is described by an equation that goes back to Schroedinger, one of the founders of quantum physics:

$$i\hbar \cdot \frac{\partial}{\partial t} = H. \tag{30}$$

Here, as before, $i \stackrel{\text{def}}{=} \sqrt{-1}$ and H is a corresponding linear operator – in line with the fact that, as we have mentioned earlier, all changes of quantum states are described by linear transformations.

It can be shown that the linear operator H used in Schroedinger's equation has a clear physical meaning: it represents potential energy. From this viewpoint, the use of quantum dynamics in optimization is in line with the general use of simulations of physical phenomena as a way to solve optimization problems. This use is based on the fact that in nature, a physical system tends to be in the state with the smallest possible value of potential energy. For example, if rain falls on the mountaintop, the water rolls down, to places where the potential energy is lower. So, a natural idea is to have the energy operator H proportional to the values of the objective function that we want to minimize. This way, no matter in what state we start, the quantum system will reach a state with lower energy – and thus, a state with the smaller value of the objective function.

The problem with this idea is that in practice, the physical system does not go all the way down: for example, water from a rain can get stuck in hole, forming a puddle. In optimization terms, the system gets stuck in a local minimum of the optimized function. To reach the global minimum, we need to push the system out of the local minimum, so that it will continue its descent. For example, if a ball rolls down from the top of a hill and gets stuck in a hole, we need to push it out of the hole, then it will continue rolling down.

It is important to take into account that we want the system to eventually reach the global minimum and stay there. Thus, eventually, we should stop pushing – otherwise, the system will continue changing and will never stabilize. So, the intensity of pushing should eventually decrease. This “pushing” is the main idea behind an algorithm known as simulated annealing.

Pushing means that instead of letting the system follow the original trajectory – as determined by the gravitational forces – we apply additional forces. In quantum physics, the dynamics is described by Schroedinger's equation, i.e., by the operator H. Thus, the only way to describe additional pushes is to modify this operator. So, quantum annealing means adding an additional term – decreasing with time – to the operator H.

In non-quantum annealing, a push is characterized by one parameter – its intensity (t). As we have mentioned, this intensity should decrease with time, tending to 0 as t increases. Similarly, in the quantum case, it is natural to describe annealing by a scalar equantity (t), i.e., to replace the original equation (30), in which H is exactly proportional to the minimized objective function, with a modi ed equation

$$i \sim \frac{@}{@t} = H + (t) H_0; \qquad (31)$$

where H_0 describes the "unit push", and (t) monotonically tends to 0 as t increases. The function (t) is called an annealing schedule

In general, the effectiveness of simulated annealing depends on the selection of the annealing schedule:

- if a function (t) decreases too fast, then by the time the system reaches a local minimum, the intensity of the push will be not suf cient to move the system out of this local minimum – and the system will get stuck in a local minimum;
- on the other hand, if a function (t) decreases too slowly, then for a long time, the force of the pushes may be stronger than the force pushing the system down the hill – so the system will oscillate instead of going down, and the objective function will not decrease for a long time, it will start decreasing only when the intensity of pushes will get smaller.

The situation is similar with quantum annealing – the effectiveness of quantum annealing strongly depends on the selection of the annealing schedule:

- for some annealing schedules, quantum annealing works wonders and helps minimize complex objective functions, while
- for other annealing schedules, quantum annealing is not effective at all.

In quantum annealing, there are, at present, no theoretically justi ed recommendations on what annealing schedule to select, but – due to the fact that, as we have mentioned, quantum annealing is actually used in commercially available quantum computing devices – there is a lot of empirical data comparing the effectiveness of different annealing schedules. Empirically, the following two types of annealing schedules work the best:

- the power law annealing schedule, when $(t) = A \cdot t^a$, for some A and $a < 0$; see, e.g., [22], [23]; and
- the exponential annealing schedule, when

$$(t) = A \cdot \exp(a \cdot t)$$

for some A and $a < 0$; see, e.g., [7], [23].

In this section, we provide a theoretical proof that these schedules are indeed optimal. As we have mentioned, this result rst appeared in [14]

C. WE NEED TO SELECT A FAMILY OF ANNEALING SCHEDULES

The effectiveness of quantum annealing also depends on the selection of a unit push H_0. There is no xed unit of energy. If we select, for measuring energy, a measuring unit which is C times larger than the previous one, then the new unit push $H_0' = C \cdot H_0$. So, if we apply the same annealing schedule (t) but with the new push, we will get the equation

$$i \sim \frac{@}{@t} = H + (t) H_0' = H + (t) \cdot C \cdot H_0: \qquad (32)$$

This equation can be represented in an equivalent form

$$i \sim \frac{@}{@t} = H + (C \cdot (t)) \cdot H_0: \qquad (33)$$

This is equivalent to using the original unit push H_0, but with a new annealing schedule $'(t) = C \cdot (t)$.

The quality of quantum annealing should not depend on what unit we use to measure energy. Since, as we have just shown, changing the unit energy is equivalent to replacing the original annealing schedule (t) with the new schedule $C \cdot (t)$, this means that these two annealing schedules must have the same effectiveness. Thus, for every annealing schedule (t), all functions of the type $C \cdot (t)$ have the same quality.

So, we cannot select a single annealing schedule (t) as the best – since for this schedule, all schedules of the type $C \cdot (t)$ will also be equally optimal. Since all the schedules from the whole family of functions $\{C \cdot (t)\}$ have the same quality – so we cannot distinguish between different functions from this family – what we need to do is select the optimal family.

D. LET US USE THE FACT THAT WE CAN ALSO SELECT DIFFERENT UNITS FOR MEASURING TIME

The equation (31) connects two physical quantity: energy – as described by the operator H – and time. So far, we talked about selecting different measuring units for energy. However, for measuring time, we can also use different units. If, instead of the original time unit, we select a unit which is times smaller, then all numerical values of time are multiplied by : $t \to t$. For example, if we replace seconds with milliseconds, a thousand times smaller measuring unit, then 0.5 seconds become $1000 \cdot 0.5 = 500$ milliseconds.

There is no preferred unit of time, so it make sense to require that the relative quality of two families of annealing schedules should not change if we simply replace the unit for measuring time. In other words, if we have

$$\{C \cdot {}_1(t)\}_{C>0} < \{C \cdot {}_2(t)\}_{C>0};$$

then, for each $C > 0$, we should also have

$$\{f_{C\cdot 1}(t)\}_{C>0} < \{f_{C\cdot 2}(t)\}_{C>0}.$$

Let us show that this natural requirement explains the power law annealing schedule. Indeed, according to the above Lemma, when the final optimality criterion is invariant with respect to the transformation $t \to t$, then the optimal family must be also invariant with respect to this transformation. Thus, for the optimal family, we should have

$$\{f_{C\cdot}(t)\}_{C>0} = \{f_{C\cdot}(t)\}_{C>0}.$$

The equality of the two families means, in particular, that every function from the family on the left-hand side of this equality – in particular, the function (t) corresponding to $C = 1$ – must also be an element of the family on the right, i.e., it must have the form

$$(t) = C(\ )\ (t)$$

for some C depending on . It is known (see, e.g., [1]) that the only monotonic solutions to this functional equation are power laws $(t) = A \cdot t^a$.

So, we conclude that each optimal annealing schedule is described by a power law. This provide a theoretical justification for the above empirical fact.

E. WHAT ABOUT THE EXPONENTIAL ANNEALING SCHEDULE?

So far, we have explained the power law annealing schedule, but where does the exponential annealing schedule come from? To answer this question, let us take into account that for time, there is no fixed starting point. So, it is reasonable to require that the relative quality of two families should not change is we simply select a different starting point for measuring time.

If we replace the original starting point with the one which is t_0 units earlier, then all numerical values t are replaced with shifted values $t + t_0$. So, the above requirement means that if

$$\{f_{C\cdot 1}(t)\}_{C>0} < \{f_{C\cdot 2}(t)\}_{C>0},$$

then we should also have

$$\{f_{C\cdot 1}(t + t_0)\}_{C>0} < \{f_{C\cdot 2}(t + t_0)\}_{C>0}.$$

What can we conclude from this requirement? According to the above Lemma, for every final shift-invariant optimality criterion, the optimal family should also be shift-invariant, so we should have

$$\{f_{C\cdot}(t + t_0)\}_{C>0} = \{f_{C\cdot}(t)\}_{C>0}.$$

This equality implies, in particular, that every element of the first family – in particular, the function $(t + t_0)$ corresponding to $C = 1$ – also belongs to the second family, i.e., has the form

$$(t + t_0) = C(t_0)\ (t)$$

for some C depending on t_0. It is known (see, e.g., [1]) that every monotonic solution to this functional equation has the exponential form $(t) = A \cdot \exp(a \cdot t)$.

Thus, for a final shift-invariant optimality criterion on the class of all families, every optimal annealing schedule has the exponential law form. This result explains the empirical efficiency of the exponential-law annealing schedules.

References

[1] J. Aczél and J. Dhombres, Functional Equations in Several Variables, Cambridge University Press, 2008.

[2] I. Bautista, V. Kreinovich, O. Kosheleva, and H. P. Nguyen, "Why it is sufficient to have real-valued amplitudes in quantum computing", In: Nguyen Hoang Phuong and V. Kreinovich (Eds.), Soft Computing: Biomedical and Related Applications, Springer: Cham, Switzerland, 2021, pp. 131–136. https://doi.org/10.1007/978-3-030-76620-7_11

[3] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, "Teleporting an unknown quantum state via dual classical and Einstein-Podolsky–Rosen channels", Physical Review Letters, vol. 70, pp. 1895–1899, 1993. https://doi.org/10.1103/PhysRevLett.70.1895

[4] S. Boixo, T. F. Ronnow, S. V. Isakov, Z. Wang, D. Wecker, D. A. Lidar, J. M. Martinis, and M. Troyer, "Evidence for quantum annealing with more than one hundred qubits", Nature Physics, vol. 10, pp. 218–224, 2014. https://doi.org/10.1038/nphys2900

[5] J. Brooke, D. Bitko, T. F. Rosenbaum, and G. Aeppli, "Quantum annealing of a disordered magnet", Science, vol. 284, pp. 779–781, 1999. https://doi.org/10.1126/science.284.5415.779

[6] E. Crosson, E. Farhi, C. Y.-Y. Lin, H.-H. Lin, and P. Shor, "Different Strategies for Optimization Using the Quantum Adiabatic Algorithm", 2014, arXiv:1401.7320.

[7] A. Das, B. K. Chakrabarti, and R. B. Stinhcombe, "Quantum annealing in a kinetically constrained system", Physical Review Letters, vol. 72, Paper 026701, 2005. https://doi.org/10.1103/PhysRevE.72.026701

[8] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Ludgren, and D. Preda, "A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem", Science, vol. 292, pp. 472–475, 2001. https://doi.org/10.1126/science.1057726

[9] R. Feynman, R. Leighton, and M. Sands, The Feynman Lectures on Physics, Addison Wesley, Boston, Massachusetts, 2005.

[10] A. B. Finnila, M. A. Gomez, C. Sebenik, C. Stenson, and J. D. Doll, "Quantum annealing: A new method for minimizing multidimensional functions", Chemical Physics Letters, vol. 219, no. 5–6, pp. 343–348, 1994. https://doi.org/10.1016/0009-2614(94)00117-0

[11] O. Galindo, L. Bokati, and V. Kreinovich, "Towards a more efficient representation of functions in quantum and reversible computing", Proceedings of the Joint 11th Conference of the European Society for Fuzzy Logic and Technology EUSFLAT'2019 and International Quantum Systems Association (IQSA) Workshop on Quantum Structures, Prague, Czech Republic, September 9–13, 2019. https://doi.org/10.2991/eusflat-19.2019.10

[12] O. Galindo, O. Kosheleva, and V. Kreinovich, "Towards parallel quantum computing: standard quantum teleportation algorithm is, in some reasonable sense, unique", In: H. Seki, C. H. Nguyen, V.-N. Huynh, and M. Inuiguchi (Eds.), USB Proceedings of the 7th International Symposium on Integrated Uncertainty in Knowledge Modelling and Decision Making IUKM'2019, Nara, Japan, March 27–29, 2019, pp. 23–34.

[13] O. Galindo, O. Kosheleva, and V. Kreinovich, "How to efficiently store intermediate results in quantum computing: theoretical explanation of the current algorithm", In: S. Sriboonchitta, V. Kreinovich, and W. Yamaka (Eds.), Credible Asset Allocation, Optimal Transport Methods, and Related Topics, Springer, Cham, Switzerland, 2022, pp. 51–64. https://doi.org/10.1007/978-3-030-97273-8_5

[14] O. Galindo and V. Kreinovich, "What is the optimal annealing schedule in quantum annealing", Proceedings of the IEEE Series of Symposia on Computational Intelligence SSCI'2020, Canberra, Australia, December 1–4, 2020. https://doi.org/10.1109/SSCI47803.2020.9308407

[15] O. Galindo, V. Kreinovich, and O. Kosheleva, "Current quantum cryptography algorithm is optimal: a proof", Proceedings of the IEEE Symposium on Computational Intelligence for Engineering Solutions CIES'2018, Bangaluru, India, November 18–21, 2018. https://doi.org/10.1109/SSCI.2018.8628876