

6-1-2021

Many Known Quantum Algorithms Are Optimal: Symmetry-Based Proofs

Vladik Kreinovich

The University of Texas at El Paso, vladik@utep.edu

Oscar Galindo

The University of Texas at El Paso, ogalindomo@miners.utep.edu

Olga Kosheleva

The University of Texas at El Paso, olgak@utep.edu

Follow this and additional works at: https://scholarworks.utep.edu/cs_techrep



Part of the [Computer Sciences Commons](#), and the [Mathematics Commons](#)

Comments:

Technical Report: UTEP-CS-21-59a

Recommended Citation

Kreinovich, Vladik; Galindo, Oscar; and Kosheleva, Olga, "Many Known Quantum Algorithms Are Optimal: Symmetry-Based Proofs" (2021). *Departmental Technical Reports (CS)*. 1592.

https://scholarworks.utep.edu/cs_techrep/1592

This Article is brought to you for free and open access by the Computer Science at ScholarWorks@UTEP. It has been accepted for inclusion in Departmental Technical Reports (CS) by an authorized administrator of ScholarWorks@UTEP. For more information, please contact lweber@utep.edu.

Many Known Quantum Algorithms Are Optimal: Symmetry-Based Proofs

Vladik Kreinovich , Oscar Galindo, and Olga Kosheleva 

University of Texas at El Paso; vladik@utep.edu, ogalindomo@miners.utep.edu, olgak@utep.edu

* Correspondence: vladik@utep.edu (V.K.)

Abstract: Many quantum algorithms have been proposed which are drastically more efficient than 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).

Keywords: quantum computing; optimal algorithms; invariance; symmetry

1. Formulation of the Problem

1.1. Need for quantum computing

Modern computers are extremely fast, but still there are many practical problems 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*

1.2. 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 probabili-

Citation: Kreinovich, V.; Galindo, O.; Kosheleva, O. Many Known Quantum Algorithms Are Optimal. *Symmetry* **2021**, *13*, 0. <https://doi.org/>

Received:

Accepted:

Published:

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Submitted to *Symmetry* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

39 ties, while in computations, we usually want to come up with a single result. Thus, we
40 need to develop new algorithms.

41 1.3. Quantum algorithms: successes

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

46 To briefly describe these successes, let us recall what are the fundamental computa-
47 tion needs. To enumerate these needs, let us recall what we humans want.

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

56 Quantum algorithms are useful in solving all these main problems of science and engi-
57 neering:

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

78 1.4. Quantum algorithm: remaining challenges and what we cover in this paper

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

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

89 *1.5. Structure of this paper*

90 We start, in Section 2, with a brief reminder of the quantum basics – basics which
 91 are needed to understand the main ideas behind the existing quantum algorithms and
 92 behind the proofs of their optimality. In Section 3, we describe the relation between
 93 optimality – that we want to prove – and symmetries – i.e., invariance with respect
 94 to different transformations. After that, we present the proofs of optimality of differ-
 95 ent quantum algorithms for quantum data processing: Grover’s algorithm in Section
 96 4, parallel-related teleportation algorithm in Section 5, and an optimization-related
 97 quantum annealing algorithm in Section 6.

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

101 **2. Quantum basics**

102 *2.1. Quantum states*

In “classical” (= non-quantum) physics, each object, each system can be in different
 states s, s', \dots . 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

$$c \cdot |s\rangle + c' \cdot |s'\rangle + \dots, \quad (1)$$

where c, c', \dots are complex numbers for which

$$|c|^2 + |c'|^2 + \dots = 1, \quad (2)$$

103 where, as usual, for a complex number $c = a + b \cdot i$, its modulus $|c|$ is defined as
 104 $|c| = \sqrt{a^2 + b^2}$. If the system is in the state (1), and we use a classical measurement
 105 instrument to measure the state, then:

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

108 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

$$c_0 \cdot |0\rangle + c_1 \cdot |1\rangle, \quad (3)$$

where c_0 and c_1 are complex numbers for which

$$|c_0|^2 + |c_1|^2 = 1. \quad (4)$$

109 In the state (3), the probability that we will observe 0 is $|c_0|^2$, and the probability that we
 110 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

$$c_{00} \cdot |00\rangle + c_{01} \cdot |01\rangle + c_{10} \cdot |10\rangle + c_{11} \cdot |11\rangle. \quad (5)$$

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

114 2.2. Quantum measurements

115 In general, if we have n classical states s_1, \dots, s_n , and we want to detect, in a
116 quantum state $\sum \alpha_i \cdot s_i$, which of these states we are in, we get each s_i with probability
117 $|\alpha_i|^2$ – and once the measurement process detects the state s_i , the actual state turns into s_i .

118 Instead of the classical states s_1, \dots , we can use any other sequence of states $s'_i =$
119 $\sum_j t_{ij} \cdot s_j$, as long as they are *orthonormal* (= orthogonal and normal) in the sense that:

- 120 • for each i , we have $\|s'_i\|^2 = 1$, where $\|s'_i\|^2 \stackrel{\text{def}}{=} \sum_j |t_{ij}|^2$ (*normal*), and
- 121 • for each i and i' , we have $s'_i \perp s'_{i'}$, i.e., $\langle s'_i | s'_{i'} \rangle = 0$, where $\langle s'_i | s'_{i'} \rangle \stackrel{\text{def}}{=} \sum_j t_{ij} \cdot t_{i'j}^*$
122 (*orthogonal*).

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

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

130 2.3. Composite systems

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

$$a \cdot |t\rangle + a' \cdot |t'\rangle + \dots, \quad (6)$$

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

$$(c \cdot |s\rangle + c' \cdot |s'\rangle + \dots) \otimes (a \cdot |t\rangle + a' \cdot |t'\rangle + \dots), \quad (7)$$

is equal to

$$c \cdot a \cdot |s, t\rangle + c \cdot a' \cdot |s, t'\rangle + \dots + c' \cdot a \cdot |s', t\rangle + c' \cdot a' \cdot |s', t'\rangle + \dots \quad (8)$$

137 In particular, for classical states, e.g., when $c = a = 1$ and $c' = \dots = a' = \dots = 0$, we get
138 $|s\rangle \otimes |t\rangle = |s, t\rangle$.

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

145 2.4. 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

$$c_1 \cdot |s_1\rangle + \dots + c_n \cdot |s_n\rangle \quad (9)$$

is transformed into the state

$$c'_1 \cdot |s_1\rangle + \dots + c'_n \cdot |s_n\rangle, \quad (10)$$

for which

$$c'_i = \sum_{j=1}^n T_{ij} \cdot c_j \quad (11)$$

146 for some coefficient T_{ij} . The matrix $T = \|T_{ij}\|$ is *unitary*: $T^\dagger T = TT^\dagger = I$, where I is the
147 unit matrix, and $T_{ij}^\dagger \stackrel{\text{def}}{=} T_{ji}^*$, where c^* denotes *complex conjugate*: $(a + b \cdot i)^* \stackrel{\text{def}}{=} a - b \cdot i$.

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

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

$$H(|0\rangle) = |0'\rangle \stackrel{\text{def}}{=} \frac{1}{\sqrt{2}} \cdot |0\rangle + \frac{1}{\sqrt{2}} \cdot |1\rangle; \quad H(|1\rangle) = |0'\rangle \stackrel{\text{def}}{=} \frac{1}{\sqrt{2}} \cdot |0\rangle - \frac{1}{\sqrt{2}} \cdot |1\rangle. \quad (11)$$

151 2.5. How functions are represented in quantum algorithms

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

- 158 • we could have $(x_1, x_2) = (0, 0)$,
- 159 • we could have $(x_1, x_2) = (0, 1)$, or
- 160 • we could have $(x_1, x_2) = (1, 0)$.

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

$$T_f(x_1, \dots, x_n, y) = (x_1, \dots, x_n, y \oplus f(x_1, \dots, x_n)), \quad (12)$$

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

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

168 3. Relation Between Optimality and Invariance (Symmetry)

169 3.1. What is invariance (symmetry)

170 In many cases, there are some natural transformations that does not change the
171 system. This “not changing” is called *invariance*. For example, suppose that we have
172 an unsorted list, and we are looking for an element with a certain property in this list.
173 For convenience, we can denote one of the list’s elements by s_1 , another one by s_2 ,
174 etc., but in this problem, it does not matter which element is called s_1 , which s_2 , etc. –
175 any permutation $\pi : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ would retain the problem. Thus, in this
176 problems, permutations are invariances. In other problems, we will have other natural
177 invariances.

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

180 4. What does “optimal” mean

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

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

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

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

201 To simplify the analysis, it is useful to ignore all these objective functions $f(A)$,
 202 $g(A)$, etc., and to only consider what really matters: for each pair (A, A') , according to
 203 this criterion:

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

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

- 209 • if $A < A'$ and $A' < A''$, then $A < A''$;
- 210 • if $A < A'$ and $A' \sim A''$, then $A < A''$;
- 211 • if $A \sim A'$ and $A' < A''$, then $A < A''$;
- 212 • if $A \sim A'$ and $A' \sim A''$, then $A \sim A''$;
- 213 • if $A \sim A'$, then $A' \sim A$; and
- 214 • if $A < A'$, then we cannot have $A \sim A'$.

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

217 An alternative A_{opt} is called *optimal* if for every other alternative A , we have
 218 $A < A_{\text{opt}}$ or $A \sim A_{\text{opt}}$. An optimality criterion is called *final* if there is exactly one
 219 alternative which is optimal with respect to this criterion.

220 4.1. For invariant criteria, optimal alternative is also invariant

221 In many cases, there exists a reversible transformation $T : \mathcal{A} \rightarrow \mathcal{A}$ – e.g., permuta-
 222 tion – which does not change the situation. In this case, it makes sense to require that
 223 this transformation will not change which alternative is better. In precise terms, we say
 224 that an optimality criterion is *T-invariant* if the following conditions are satisfied:

- 225 • if $A < A'$, then $T(A) < T(A')$;
- 226 • if $A \sim A'$ then $T(A) \sim T(A')$.

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

228 **Lemma.** *For every final T-invariant optimality criterion, its optimal alternative A_{opt} is also*
 229 *T-invariant, i.e., $T(A_{\text{opt}}) = A_{\text{opt}}$.*

230 **Proof.** The fact that A_{opt} means that for every $A \in \mathcal{A}$, we have either $A < A_{\text{opt}}$
 231 or $A \sim A_{\text{opt}}$. In particular, this is true for $T^{-1}(A)$, i.e., either $T^{-1}(A) < A_{\text{opt}}$ or
 232 $T^{-1}(A) \sim A_{\text{opt}}$. Due to T -invariance, we can conclude that either $A < T(A_{\text{opt}})$ or
 233 $A \sim T(A_{\text{opt}})$. This is true for every alternative A , which means that the alternative
 234 $T(A_{\text{opt}})$ is also optimal. However, the optimality criterion is final, which means that
 235 there is only one optimal criterion. Thus, indeed, $T(A_{\text{opt}}) = A_{\text{opt}}$. The lemma is proven.
 236

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

240 5. Grover's Algorithm Is Optimal

241 5.1. Formulation of the problem

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

Let us describe this problem in 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:

$$c_1 \cdot |1\rangle + \dots + c_{i_0-1} \cdot |i_0 - 1\rangle + c_{i_0} \cdot |i_0\rangle + c_{i_0+1} \cdot |i_0 + 1\rangle + \dots + c_n \cdot |n\rangle. \quad (13)$$

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

- 248 • for $i \neq i_0$, we have $T_f(|i, 0\rangle) = |i, 0\rangle$ and $T_f(|i, 1\rangle) = |i, 1\rangle$, while
- 249 • for $i = i_0$, we have $T_f(|i_0, 0\rangle) = |i_0, 1\rangle$ and $T_f(|i_0, 1\rangle) = |i_0, 0\rangle$.

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

- 251 • for $|0'\rangle$, for all i , we have $T_f(|i\rangle \otimes |0'\rangle) = |i\rangle \otimes |0'\rangle$;
- 252 • for $|1'\rangle$, for all $i \neq i_0$, we have $T_f(|i\rangle \otimes |1'\rangle) = |i\rangle \otimes |1'\rangle$, while for $i = i_0$, we have
 253 $T_f(|i_0\rangle \otimes |1'\rangle) = -|i_0\rangle \otimes |1'\rangle$.

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:

$$c_1 \cdot |1\rangle + \dots + c_{i_0-1} \cdot |i_0 - 1\rangle - c_{i_0} \cdot |i_0\rangle + c_{i_0+1} \cdot |i_0 + 1\rangle + \dots + c_n \cdot |n\rangle. \quad (14)$$

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

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

257 5.2. Invariance (symmetry): reminder

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

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

265 5.3. 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

$$c_1 \cdot |1\rangle + \dots + c_1 \cdot |n\rangle, \quad (14)$$

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:

$$\frac{1}{\sqrt{n}} \cdot |1\rangle + \dots + \frac{1}{\sqrt{n}} \cdot |n\rangle. \quad (15)$$

266 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

$$c'_i = a \cdot c_i + b \cdot \sum_{j \neq i} c_j. \quad (16)$$

This expression can be equivalently described as

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

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

$$\sum_{j=1}^n |c'_j|^2 = 1. \quad (18)$$

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

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

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

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

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

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

If $b = 0$, then $a = \pm 1$, 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

$$c'_i = \left(1 - \frac{2}{n}\right) \cdot c_i - \frac{2}{n} \cdot \sum_{j \neq i} c_j. \quad (21)$$

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

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

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

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

280 6. Parallelization: Teleportation Algorithm Is Optimal

281 6.1. Need for parallelization

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

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

292 6.2. Need for teleportation

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

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

299 6.3. What we do in this section

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

304 6.4. Standard quantum teleportation algorithm: reminder

305 6.4.1. Need for communication

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

308 Usually, the sender is denoted by A and the receiver by B . In communications, it is
309 common to call the sender Alice, and to call the receiver Bob. States corresponding to

310 Alice are usually described by using a subscript A , and states corresponding to Bob are
 311 usually described by using a subscript B .

312 6.4.2. Communication is straightforward in classical physics but a challenge in quantum
 313 physics

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

320 The problem is that in quantum physics, such a straightforward approach is not
 321 possible: as we have mentioned, in quantum physics, every measurement changes the
 322 state – and moreover, irreversibly deletes some information about the state. For example,
 323 if we start with a state $\alpha_0 \cdot |0\rangle + \alpha_1 \cdot |1\rangle$, all we get after the measurement is either 0 or
 324 1, with no way to reconstruct the values α_0 and α_1 that characterize the original state.
 325 Since we cannot use the usual straightforward approach for communicating a state, we
 326 need to use an indirect approach. This approach is known as *teleportation*.

327 6.4.3. What we consider in this section

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

337 In these terms, the problem is as follows:

- Alice has a state

$$\alpha_0 \cdot |0\rangle + \alpha_1 \cdot |1\rangle \quad (22)$$

338 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.

340 6.4.4. Notations

Let us indicate states corresponding to Alice with a subscript A , and states corre-
 sponding 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

$$\alpha_0 \cdot |0\rangle_C + \alpha_1 \cdot |1\rangle_C \quad (23)$$

341 that she wants to communicate to Bob.

342 6.4.5. Preparing for teleportation: an entangled state

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

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

343 This state is a superposition of two classical states:

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

346 6.4.6. 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):

$$\frac{\alpha_0}{\sqrt{2}} \cdot |0_A 1_B 0_C\rangle + \frac{\alpha_1}{\sqrt{2}} \cdot |0_A 1_B 1_C\rangle + \frac{\alpha_0}{\sqrt{2}} \cdot |1_A 0_B 0_C\rangle + \frac{\alpha_1}{\sqrt{2}} \cdot |1_A 0_B 1_C\rangle. \quad (25)$$

347 6.4.7. 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

$$s = \sum s_i, \quad (26)$$

348 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:

$$\begin{aligned} t_1 &= \frac{1}{\sqrt{2}} \cdot |0_A 0_C\rangle + \frac{1}{\sqrt{2}} \cdot |1_A 1_C\rangle; & t_2 &= \frac{1}{\sqrt{2}} \cdot |0_A 0_C\rangle - \frac{1}{\sqrt{2}} \cdot |1_A 1_C\rangle; \\ t_3 &= \frac{1}{\sqrt{2}} \cdot |0_A 1_C\rangle + \frac{1}{\sqrt{2}} \cdot |1_A 0_C\rangle; & t_4 &= \frac{1}{\sqrt{2}} \cdot |0_A 1_C\rangle - \frac{1}{\sqrt{2}} \cdot |1_A 0_C\rangle. \end{aligned} \quad (27)$$

349 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 $s = \sum s_i$, with $s_i \in L_i$. For this purpose, we can use the fact that, due to the formulas (27), we have

$$\begin{aligned} |0_A 0_C\rangle &= \frac{1}{\sqrt{2}} \cdot t_1 + \frac{1}{\sqrt{2}} \cdot t_2; & |1_A 1_C\rangle &= \frac{1}{\sqrt{2}} \cdot t_1 - \frac{1}{\sqrt{2}} \cdot t_2; \\ |0_A 1_C\rangle &= \frac{1}{\sqrt{2}} \cdot t_3 + \frac{1}{\sqrt{2}} \cdot t_4; & |1_A 0_C\rangle &= \frac{1}{\sqrt{2}} \cdot t_3 - \frac{1}{\sqrt{2}} \cdot t_4. \end{aligned} \quad (28)$$

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

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

thus

$$\begin{aligned} &\left(\frac{\alpha_0}{2} |1_B\rangle + \frac{\alpha_1}{2} |0_B\rangle \right) \otimes t_1 + \left(\frac{\alpha_0}{2} |1_B\rangle - \frac{\alpha_1}{2} |0_B\rangle \right) \otimes t_2 + \\ &\left(\frac{\alpha_1}{2} |1_B\rangle + \frac{\alpha_0}{2} |0_B\rangle \right) \otimes t_3 + \left(\frac{\alpha_1}{2} |1_B\rangle - \frac{\alpha_0}{2} |0_B\rangle \right) \otimes t_4. \end{aligned}$$

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

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

Here, for each i , we have

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

350 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:

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

351 6.4.8. Second stage: communication

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

354 6.4.9. Final stage: Bob “rotates” his state and thus, get the original state teleported to him

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

- 356 • In the first case, he uses a unitary transformation that swaps $|0\rangle_B$ and $|1\rangle_B$, for which
357 $t_{01} = t_{10} = 1$ and $t_{00} = t_{11} = 0$.
- 358 • In the second case, he uses a unitary transformation for which $t_{01} = 1$, $t_{10} = -1$
359 and $t_{00} = t_{11} = 0$.
- 360 • In the third case, he already has the desired state.
- 361 • In the fourth case, he uses a unitary transformation for which $t_{00} = -1$, $t_{11} = 1$,
362 and $t_{01} = t_{10} = 0$.

363 As a result, in all four cases, he gets the original state $\alpha_0 \cdot |0\rangle_B + \alpha_1 \cdot |1\rangle_B$.

364 6.5. The main result of this section: the standard quantum teleportation algorithm is, in some
365 reasonable sense, unique

366 6.5.1. 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 $s_A \otimes s_B$. However, (24) is not the only possible entangled state. Let us consider, instead, a general joint state of two qubits:

$$a_{00} \cdot |0_A 0_B\rangle + a_{01} \cdot |0_A 1_B\rangle + a_{10} \cdot |1_A 0_B\rangle + a_{11} \cdot |1_A 1_B\rangle. \quad (3a)$$

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

369 6.5.2. Analysis of the problem

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

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

Substituting expressions (28) into this formula, we get

$$\begin{aligned} & \frac{\alpha_0}{\sqrt{2}} \cdot a_{00} \cdot |0\rangle_B \otimes (t_1 + t_2) + \frac{\alpha_1}{\sqrt{2}} \cdot a_{00} \cdot |0\rangle_B \otimes (t_3 + t_4) + \\ & \frac{\alpha_0}{\sqrt{2}} \cdot a_{01} \cdot |1\rangle_B \otimes (t_1 + t_2) + \frac{\alpha_1}{\sqrt{2}} \cdot a_{01} \cdot |1\rangle_B \otimes (t_3 + t_4) + \\ & \frac{\alpha_0}{\sqrt{2}} \cdot a_{10} \cdot |0\rangle_B \otimes (t_3 - t_4) + \frac{\alpha_1}{\sqrt{2}} \cdot a_{10} \cdot |0\rangle_B \otimes (t_1 - t_2) + \end{aligned}$$

$$\frac{\alpha_0}{\sqrt{2}} \cdot a_{11} \cdot |1\rangle_B \otimes (t_3 - t_4) + \frac{\alpha_1}{\sqrt{2}} \cdot a_{11} \cdot |1\rangle_B \otimes (t_1 - t_2),$$

thus $s = S_1 \otimes t_1 + S_2 \otimes t_2 + \dots$, where

$$S_1 = \left(\frac{\alpha_0 \cdot a_{00}}{\sqrt{2}} + \frac{\alpha_1 \cdot a_{10}}{\sqrt{2}} \right) \cdot |0\rangle_B + \left(\frac{\alpha_0 \cdot a_{01}}{\sqrt{2}} + \frac{\alpha_1 \cdot a_{11}}{\sqrt{2}} \right) \cdot |1\rangle_B,$$

370 and S_2, \dots are described by similar expressions.

371 This means that after the measurement, Bob will have the normalized state $S_1 / \|S_1\|$.
 372 To perform teleportation, we need to transform this state into the original state $\alpha_0 \cdot$
 373 $|0\rangle_B + \alpha_1 \cdot |1\rangle_B$. Thus, the transformation from the resulting state $S_1 / \|S_1\|$ to the original
 374 state must be unitary. It is known that the inverse transformation to a unitary one is
 375 also unitary. In general, a unitary transformation transforms orthonormal states into
 376 orthonormal ones.

377 So, the inverse transformation that:

- 378 • maps the state $|0\rangle_B$ (corresponding to $\alpha_0 = 1$ and $\alpha_1 = 0$) into a new state $|1'\rangle_B \stackrel{\text{def}}{=} \text{const} \cdot (a_{00} \cdot |0\rangle_B + a_{01} \cdot |1\rangle_B)$, and
- 380 • maps the state $|1\rangle_B$ (corresponding to $\alpha_0 = 0$ and $\alpha_1 = 1$) into a new state $|0'\rangle_B \stackrel{\text{def}}{=} \text{const} \cdot (a_{10} \cdot |0\rangle_B + a_{11} \cdot |1\rangle_B)$,

382 transforms two original orthonormal vectors $|0\rangle_B$ and $|1\rangle_B$ into two new orthonormal
 383 ones $|0'\rangle_B$ and $|1'\rangle_B$.

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

$$\text{const} \cdot (|0\rangle_A \otimes |1'\rangle_B + |1\rangle_B \otimes |0'\rangle_B).$$

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

$$\frac{1}{\sqrt{2}} \cdot (|0\rangle_A \otimes |1'\rangle_B + |1\rangle_B \otimes |0'\rangle_B). \quad (24)$$

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

386 6.5.3. Conclusion of this section

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

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

393 7. Optimization: Quantum Annealing Schedules Are Optimal

394 7.1. Quantum annealing: ideas, successes, and challenges

395 One of the important practical problems is optimization. An important challenge
 396 is that often, the existing optimization techniques lead to a local optimum. One way to
 397 avoid local optima is *annealing*: whenever we find ourselves in a possibly local optimum,
 398 we jump out with some probability and continue search for the true optimum. Since
 399 quantum processes are probabilistic, a natural way to organize such a probabilistic
 400 perturbation of the deterministic optimization is to use quantum effects, i.e., to perform
 401 *quantum annealing*. This idea was first proposed in [10,19] and has been used successfully
 402 since then.

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

407 The efficiency of quantum annealing depends on the proper selection of the an-
 408 nealing schedule, i.e., schedule that describes how the perturbations decrease with time.
 409 Empirically, it has been found that two schedules work best: power law and exponential
 410 ones. In this section, following [14], we prove that these two schedules are indeed
 411 optimal (in some reasonable sense).

412 7.2. Formulation of the problem

In general, the state of a quantum system is described by a complex-valued function $\psi(t)$ (known as the *wave function*), and the dynamics of a quantum system is described by Schroedinger's equations

$$i \cdot \hbar \cdot \frac{\partial \psi}{\partial t} = H\psi, \quad (30)$$

where, as before, $i \stackrel{\text{def}}{=} \sqrt{-1}$ and H is a corresponding linear operator. In these terms, annealing-type modification means adding additional terms – decreasing with time – to the operator H , i.e., replacing the original equation (30) with the modified equation

$$i \cdot \hbar \cdot \frac{\partial \psi}{\partial t} = H\psi + \gamma(t) \cdot H_0\psi, \quad (31)$$

413 where H_0 describes the deviation, and $\gamma(t)$ monotonically tends to 0 as t increases.

414 The efficiency of quantum annealing strongly depends on the proper selection of
 415 the annealing schedule, i.e., the dependence of $\gamma(t)$ on time t . Empirically, depending
 416 on the specific optimization problem, two scheduled work the best:

- 417 • the power law annealing schedule $\gamma(t) = A \cdot t^a$, for some A and $a < 0$; see, e.g.,
 418 [22,23]; and
- 419 • the exponential annealing schedule $\gamma(t) = A \cdot \exp(a \cdot t)$ for some A and $a < 0$; see,
 420 e.g., [7,23].

421 In this section, we provide a theoretical proof that these schedules are indeed optimal.

422 7.3. Physical meaning of annealing

423 The general idea of using simulating physical phenomena in optimization is that
 424 a physical systems tends to end up in a state with the smallest possible energy. For
 425 example, in a gravitational field, this means getting to as low a position as possible. In
 426 principle, we can place a ball on top of the mountain – which will constitute a local
 427 minimum of energy. However, if a strong wind blows and disturbs the ball, it will start
 428 falling down. It may reach a few local minima along the way, but eventually it will reach
 429 the lowest possible position at the foot of the mountain.

430 So, a natural way to use simulated physical phenomena for optimization is to
 431 simulate a system for which, for all the values of the parameters, its energy is equal to
 432 the value of minimized objective function. In general, in Schroedinger equations, energy
 433 is represented by the operator H , so for quantum annealing, this operator must represent
 434 the desired objective function.

435 7.4. Need to select a family of schedules

436 The original optimization problem is usually not formulated in terms of energy. So,
 437 how we transform it into energy depends in our choice of units. We will get completely
 438 different results if we use a typical macro-world unit like Joule or a typical micro-world
 439 unit like MeV. If we select a different unit, this means that in original unit, instead of H ,
 440 we will have $C \cdot H$, where C is the ratio of the units.

If for the original operator H the best schedule was $\gamma(t)$, then for the new operator $C \cdot H$, the best schedule is $C \cdot \gamma(t)$, since the corresponding equation

$$i \cdot \hbar \cdot \frac{\partial \psi}{\partial t} = C \cdot H\psi + C \cdot \gamma(t) \cdot H_0\psi \quad (32)$$

441 is equivalent to the original equation (31) if we re-scale the time, i.e., consider t/C instead
442 of the original time t .

443 Since, as we have mentioned, the choice of the energy unit is rather arbitrary, this
444 means that we cannot select a single annealing schedule $\gamma(t)$: with each such optimal
445 schedule, in different energy units, a schedule $C \cdot \gamma(t)$ is optimal. Thus, we can only
446 select a *family* $\{C \cdot \gamma(t)\}_{C>0}$ of annealing functions, in which a function $\gamma(t)$ is fixed, and
447 the parameter C can take any positive value.

448 7.5. Need for re-scaling time and the resulting invariances (symmetries)

449 We are looking for the dependence of $\gamma(t)$ on time, but the numerical value of time
450 also depends on the choice of the measuring unit. If we replace the original unit of time
451 by a λ times smaller unit, then, for each moment of time, the original numerical value t
452 is replaced by a new value $\lambda \cdot t$. For example, if we replace minutes by seconds, then 2
453 minutes becomes $60 \cdot 2 = 120$ seconds.

It is reasonable to require that the relative quality of a family not change if we simply change the unit for time: e.g., if

$$\{C \cdot \gamma_1(t)\}_{C>0} < \{C \cdot \gamma_2(t)\}_{C>0},$$

then we should have

$$\{C \cdot \gamma_1(\lambda \cdot t)\}_{C>0} < \{C \cdot \gamma_2(\lambda \cdot t)\}_{C>0}.$$

The numerical value of time also depends on the choice of the starting point. 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 $t + t_0$. It also makes sense to require that the relative quality of two families not depend on the choice of the starting point, i.e., that if

$$\{C \cdot \gamma_1(t)\}_{C>0} < \{C \cdot \gamma_2(t)\}_{C>0},$$

then we should have

$$\{C \cdot \gamma_1(t + t_0)\}_{C>0} < \{C \cdot \gamma_2(t + t_0)\}_{C>0}.$$

454 7.6. Resulting proof

455 According to our Lemma, once we assumed that the optimality criterion is invariant,
456 then the optimal family must be invariant.

For invariance with respect to changing the unit of time, this means that

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

457 This inequality means, in particular, that the function $\gamma(\lambda \cdot t)$ from the first family
458 belongs to the second family, i.e., that for every t and λ , we have $\gamma(\lambda \cdot t) = C(\lambda) \cdot \gamma(t)$
459 for some C depending on λ . It is known (see, e.g., [1]) that the only monotonic solutions
460 to this functional equation are power laws $\gamma(t) = A \cdot t^a$.

461 Similarly, for invariance with respect to a starting point, invariance means that
462 $\{C \cdot \gamma(t + t_0)\}_{C>0} = \{C \cdot \gamma(t)\}_{C>0}$. This inequality means, for every t and t_0 , we have
463 $\gamma(t + t_0) = C(t_0) \cdot \gamma(t)$ for some C depending on t_0 . It is known (see, e.g., [1]) that
464 the only monotonic solutions to this functional equation are exponential laws $\gamma(t) =$
465 $A \cdot \exp(a \cdot t)$.

466 Thus, the power law and the exponential law are indeed the only invariant functions
467 – and thus, the optimal law must be either a power law or an exponential law.

468 **Author Contributions:** All three authors contributed equally to the paper.

469 **Funding:** This work was supported in part by the National Science Foundation grants 1623190
470 (A Model of Change for Preparing a New Generation for Professional Practice in Computer
471 Science), and HRD-1834620 and HRD-2034030 (CAHSI Includes), and by the AT&T Fellowship
472 in Information Technology. It was also supported by the program of the development of the
473 Scientific-Educational Mathematical Center of Volga Federal District No. 075-02-2020-1478.

474 **Institutional Review Board Statement:** Not applicable

475 **Informed Consent Statement:** Not applicable

476 **Data Availability Statement:** Not applicable.

477 **Conflicts of Interest:** The authors declare no conflict of interest.

References

1. Aczél, J.; Dhombres, J. *Functional Equations in Several Variables*; Cambridge University Press, 2008.
2. Bautista, I.; Kreinovich, V.; Kosheleva, O.; Nguyen, H.P. Why it is sufficient to have real-valued amplitudes in quantum computing, In: *Soft Computing: Biomedical and Related Applications*, Nguyen Hoang Phuong; Kreinovich, V., Eds.; Springer: Cham, Switzerland, 2021; pp. 131–136.
3. Bennett, C.H.; Brassard, G.; Crépeau, C.; Jozsa, R.; Peres, A.; Wootters, W.K. Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. *Physical Review Letters* **1993**, *70*, 1895–1899.
4. Boixo, S.; Ronnow, T.F.; Isakov, S.V.; Wang, Z.; Wecker, D.; Lidar, D.A.; Martinis, J.M.; Troyer, M. Evidence for quantum annealing with more than one hundred qubits, *Nature Physics* **2014**, *10*, 218–224.
5. Brooke, J.; Bitko, D.; Rosenbaum, T.F.; Aeppli, G. Quantum annealing of a disordered magnet, *Science* **1999**, *284*, 779–781.
6. Crosson, E.; Farhi, E.; Lin, C.Y.-Y.; Lin, H.-H.; Shor, P. *Different Strategies for Optimization Using the Quantum Adiabatic Algorithm*, 2014, arXiv:1401.7320.
7. Das, A.; Chakrabarti, B.K.; Stinchcombe, R.B. Quantum annealing in a kinetically constrained system, *Physical Review Letters* **2005**, *72*, Paper 026701.
8. Farhi, E.; Goldstone, J.; Gutmann, S.; Lapan, J.; Ludgren, A.; Preda, D. A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem, *Science* **2001**, *292*, 472–475.
9. Feynman, R.; Leighton, R.; Sands, M. *The Feynman Lectures on Physics*; Addison Wesley: Boston, Massachusetts, 2005.
10. Finnila, A.B.; Gomez, M.A.; Sebenik, C.; Stenson, C.; Doll, J.D. Quantum annealing: A new method for minimizing multidimensional functions", *Chemical Physics Letters* **1994**, *219*(5–6), 343–348.
11. Galindo, O.; Bokati, L.; Kreinovich, V. 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.
12. Galindo, O.; Kosheleva, O.; Kreinovich, V. Towards parallel quantum computing: standard quantum teleportation algorithm is, in some reasonable sense, unique, In: Seki, H.; Nguyen, C.H.; Huynh, V.-N.; Inuiguchi, M., 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. Galindo, O.; Kosheleva, O.; Kreinovich, V. How to efficiently store intermediate results in quantum computing: theoretical explanation of the current algorithm; In: *Credible Asset Allocation, Optimal Transport Methods, and Related Topics*; Sriboonchitta, S.; Kreinovich, V.; Yamaka, W., Eds.; Springer: Cham, Switzerland, 2022, to appear.
14. Galindo, O.; Kreinovich, V. 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.
15. Galindo, O.; Kreinovich, V.; Kosheleva, O. Current quantum cryptography algorithm is optimal: a proof, Proceedings of the IEEE Symposium on Computational Intelligence for Engineering Solutions CIES'2018, Bengaluru, India, November 18–21, 2018.
16. Grover, L.K. A fast quantum mechanical algorithm for database search", Proceedings of the 28th ACM Symposium on Theory of Computing, 1996, pp. 212–219.
17. Grover, L.K. Quantum mechanics helps in searching for a needle in a haystack, *Physical Reviews Letters* **1997**, *79*(2), 325–328.
18. Heim, B.; Ronnow, T.F.; Isakov, S.V.; Troyer, M. Quantum versus classical annealing of Ising spin glasses, *Science* **2015**, *348*, 215–217.
19. Kadowaki, T.; Nishimori, H. Quantum annealing in the transverse Ising model, *Physical Reviews E* **1998**, *58*, Paper 5355.
20. Kreinovich, V.; Ceberio, M.; Alvarez, R.; How to use quantum computing to check which inputs are relevant: a proof that Deutsch-Jozsa algorithm is, in effect, the only possibility, Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence SSCI'2019, Xiamen, China, December 6–9, 2019; pp. 828–832.

21. Lanting, T.; Przybysz, A.J.; Smirnov, A.Yu.; Spedalieri, F.M.; Amin, M.H.; Berkley, A.J.; Harris, R.; Altomare, F.; Boixo, S.; Bunyk, P.; Dickson, N.; Enderud, C.; Hilton, J.P.; Hoskinson, E.; Johnson, M.W.; Ladizinsky, E.; Ladizinsky, N.; Neufeld, R.; Oh, T.; Perminov, I.; Rich, C.; Thom, M.C.; Tolkacheva, E.; Uchaikin, S.; Wilson, S.B.; Rose, G. Entanglement in a quantum annealing processor, *Physical Review X* **2014**, *4*(2), Paper 021041.
22. Morita, S.; Nishimori, H. Mathematical foundation of quantum annealing, *Journal of Mathematical Physics* **2008**, *49*, Paper 125210.
23. Mukherjee S.; Chakrabarti, B.K. Multivariable optimization: Quantum annealing and computation", *The European Physical Journal Special Topics* **2015**, *224*, 17–24.
24. Muthukrishnan, D.; Albash, T.; Lidar, D.A. *When Diabatic Trumps Adiabatic in Quantum Optimization*, 2015, arXiv:1505.01249.
25. Nguyen, H.T.; Kreinovich, V. *Applications of Continuous Mathematics to Computer Science*; Kluwer: Dordrecht, 1997.
26. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information*; Cambridge University Press: Cambridge, U.K., 2000.
27. Pirandola, S.; Eisert, J.; Weedbrook, C.; Furusawa, A.; Braunstein, S.L. Advances in quantum teleportation, *Nature Photonics* **2015**, *9*(10), 641–652.
28. Santoro, G.E.; Tosatti, E. Optimization using quantum mechanics: quantum annealing through adiabatic evolution, *Journal of Physics A* **2006**, *39*, Paper R393.
29. Shor, P. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, November 20–22, 1994.
30. Shor, P. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer; *SIAM Journal on Computing* **1997**, *26*, 1484–1509.
31. Smelyanskiy, V.N.; Rieffel, E.G.; Knysch, S.I.; Williams, C.P.; Johnson, M.W.; Thom, M.C.; Macready, W.G.; Pudenz, K.L. *A Near-Term Quantum Computing Approach for Hard Computational Problems in Space Exploration*, 2012, arXiv:1204.2821.
32. Steiger, S.; Heim, B.; Ronnow, T.; Troyer, M. Performance of quantum annealing hardware", In: *Electro-Optical and Infrared Systems: Technology and Applications XII; and Quantum Information Science and Technology*, Huckridge, D.A.; Ebert, R.; Gruneisen, M.T.; Dusek, M.; Rarity J.G., Eds., SPIE Proceedings, 2015, Vol. 9648, Paper 964816.
33. Thorne, K.S.; Blandford, R.D. *Modern Classical Physics: Optics, Fluids, Plasmas, Elasticity, Relativity, and Statistical Physics*; Princeton University Press: Princeton, New Jersey, 2017.
34. Venegas-Andraca, S.E.; Cruz-Santos, W.; McGeoch, C.; Lanzagorta, M. A cross-disciplinary introduction to quantum annealing-based algorithms, *Contemporary Physics* **2018**, *59*(2), 174–196.
35. Williams, C.P.; Clearwater, S.H. *Ultimate Zero and One*; Copernicus: New York, 2000.