

Quantum correlation between the selection of the problem and that of the solution sheds light on the mechanism of the quantum speed-up

Giuseppe Castagnoli*

Via San Bernardo 9/A, I-16030 Pieve Ligure (Genova), Italy

(Received 18 August 2010; published 29 November 2010)

In classical problem solving, there is, of course, correlation between the selection of the problem on the part of Bob (the problem setter) and that of the solution on the part of Alice (the problem solver). In quantum problem solving, this correlation becomes quantum. This means that Alice contributes to selecting 50% of the information that specifies the problem. As the solution is a function of the problem, this gives to Alice advanced knowledge of 50% of the information that specifies the solution. Both the quadratic and exponential speed-ups are explained by the fact that quantum algorithms start from this advanced knowledge.

DOI: [10.1103/PhysRevA.82.052334](https://doi.org/10.1103/PhysRevA.82.052334)

PACS number(s): 03.67.Lx, 03.65.–w

I. OUTLINE OF THE ARGUMENT

Quantum algorithms require fewer computation steps than their classical counterparts. The reason for this quantum speed-up is not well understood. For example, recently Gross *et al.* [1] asserted that the exact reason for it has never been pinpointed. The key to the present explanation of the speed-up is the quantum correlation existing between the selection of the problem on the part of Bob (the problem setter) and that of the solution on the part of Alice (the problem solver). Because of it, all is like Alice contributed to selecting 50% of the information that specifies the problem. Since the solution is a function of the problem, this gives to Alice the advanced knowledge of 50% of the information that specifies the solution. The speed-up is explained by the fact that Alice starts from this advanced knowledge.

The quantum correlation we are dealing with appears at the level of the whole quantum experiment, from the measurement required to put the quantum system in a known state, necessary to prepare the problem, to the measurement required to extract the solution; see also Refs. [2–4].

We focus on Grover's [5] quantum search algorithm. Bob selects a value of $\mathbf{b} \in \{0, 1\}^n$, Alice should find it by computing the Kronecker function $\delta(\mathbf{b}, \mathbf{a})$ for various values of $\mathbf{a} \in \{0, 1\}^n$. We consider the simplest instance $n = 2$. With a classical algorithm, Alice should plan three computations of $\delta(\mathbf{b}, \mathbf{a})$ to be certain of finding the solution, with Grover's algorithm, one computation. There is a quantum speed-up.

In the original Grover's algorithm, a register A , under the control of Alice, contains the value of \mathbf{a} ; the value of \mathbf{b} is hard wired inside the black box that, given in input a value of \mathbf{a} , computes $\delta(\mathbf{b}, \mathbf{a})$. To highlight quantum correlation, we add a register B , under the control of Bob, containing the value of \mathbf{b} . We call \hat{A} the content of register A , \hat{B} that of register B – \hat{A} and \hat{B} are commuting observables.

Initially, register B is in a maximally mixed state. As usual, Alice prepares register A in a uniform, coherent superposition of all the possible values of \mathbf{a} . The initial state of the two

registers is thus:

$$|\psi\rangle = \frac{1}{4}(e^{i\varphi_0}|00\rangle_B + e^{i\varphi_1}|01\rangle_B + e^{i\varphi_2}|10\rangle_B + e^{i\varphi_3}|11\rangle_B) \\ (|00\rangle_A + |01\rangle_A + |10\rangle_A + |11\rangle_A), \quad (1)$$

where the φ_i are independent random phases, each with uniform distribution in $[0, 2\pi]$. We use the random-phase representation of a mixed state, instead of the density operator, to keep the usual state vector representation of the quantum algorithm. The density operator is simply the average over all the φ_i of the product of the ket by the bra: $\langle|\psi\rangle\langle\psi| \rangle_{\varphi_i}$. The two bits von Neumann entropy of the state of B —and of the overall quantum state (1)—corresponds to the complete indeterminacy of the value of \mathbf{b} .

In order to prepare register B in the desired value of \mathbf{b} , say $\mathbf{b} = 00$, Bob measures \hat{B} in state (1), thus randomly selecting a value of \mathbf{b} , say $\mathbf{b} = 01$. This projects state (1) on:

$$P_\alpha|\psi\rangle = \frac{1}{2}|01\rangle_B(|00\rangle_A + |01\rangle_A + |10\rangle_A + |11\rangle_A), \quad (2)$$

here and in the following we denote projection operators by the letter P . The entropy of the quantum state goes to zero with the determination of the value of \mathbf{b} . Then he applies to register B a permutation of the values of \mathbf{b} —a unitary transformation U_B —that changes the randomly selected value into the desired one:

$$U_B P_\alpha|\psi\rangle = \frac{1}{2}|00\rangle_B(|00\rangle_A + |01\rangle_A + |10\rangle_A + |11\rangle_A). \quad (3)$$

The unitary part of the quantum algorithm, U_{BA} , sends state (3) into

$$U_{BA} U_B P_\alpha|\psi\rangle = |00\rangle_B|00\rangle_A. \quad (4)$$

Register A contains the solution, namely the value of \mathbf{b} chosen by Bob. Alice acquires the solution by measuring \hat{A} . Of course there is a one-to-one correlation between the value of \mathbf{b} chosen by Bob and the solution found by Alice. Up to the permutation introduced by U_B , this corresponds to the quantum correlation between the outcome of measuring \hat{B} in (1) and that of measuring \hat{A} in (4). From the standpoint of quantum correlation, which concerns repetitions of the same quantum experiment, U_B should be considered fixed. The fact that Bob chooses the permutation U_B to obtain the desired value of \mathbf{b} belongs to a different film.

*giuseppe.castagnoli@gmail.com

With U_B fixed, all is like Bob's measurement of \hat{B} randomly selected the value $\mathbf{b} = 00$, which becomes in fact a fixed permutation of the randomly selected value $\mathbf{b} = 01$; in this special sense, we will speak of the random selection also of the value of \mathbf{b} chosen by Bob. Moreover, Bob's measurement can be performed indifferently at the beginning or the end of the algorithm. We show the quantum development in the case that this measurement is performed at the end:

$$|\psi\rangle = \frac{1}{4}(e^{i\varphi_0}|00\rangle_B + e^{i\varphi_1}|01\rangle_B + e^{i\varphi_2}|10\rangle_B + e^{i\varphi_3}|11\rangle_B) \\ (|00\rangle_A + |01\rangle_A + |10\rangle_A + |11\rangle_A) \quad (5)$$

$$U_B|\psi\rangle = |\psi\rangle \quad (6)$$

$$U_{BA}U_B|\psi\rangle = \frac{1}{2}(e^{i\varphi_0}|00\rangle_B|00\rangle_A + e^{i\varphi_1}|01\rangle_B|01\rangle_A \\ + e^{i\varphi_2}|10\rangle_B|10\rangle_A + e^{i\varphi_3}|11\rangle_B|11\rangle_A), \quad (7)$$

$$P_\omega U_{BA}U_B|\psi\rangle = |00\rangle_B|00\rangle_A; \quad (8)$$

of course U_B changes the maximally mixed state of register B into itself; for reasons that will become clear, we have assumed that the final measurement of \hat{A} on the part of Alice still randomly projects on $\mathbf{b} = 00$. We can see why Bob's measurement can be deferred at the end: the projection of (7) on (8), back evolved by $U_B^\dagger U_{BA}^\dagger$, becomes the projection of (1) on (2).

Thinking that all measurements are performed in the maximally entangled state (7) makes it more clear that the value of \mathbf{b} is randomly selected by either Bob's or Alice's measurement. Either measurement projects state (7) on the solution eigenstate (8), where both registers contain the randomly selected value of \mathbf{b} ; correspondingly, the two-bit entropy of the quantum state goes to zero.

Unlike measurements, projections are not localized in time. They can be back evolved by the inverse of the time forward unitary evolution. Thus, there is no reason to ascribe the projection on the solution eigenstate, or the corresponding reduction of entropy and random selection of the value of \mathbf{b} , to one measurement rather than the other. Because of the symmetry between the two measurements, we ascribe the determination of 50% of the bits of \mathbf{b} to the measurement performed by Alice, the other 50% to that performed by Bob.

Halving the projection on the solution can be done in many ways. In Sec. IIC we will symmetrize for all the possible ways, here we exemplify one way. We break down \hat{A} into content of first qubit \hat{A}_0 and content of second qubit \hat{A}_1 ; we call a_0 (a_1) the eigenvalue obtained by measuring \hat{A}_0 (\hat{A}_1). We define in a similar way \hat{B}_0 , \hat{B}_1 , b_0 , and b_1 . We ascribe to Alice the measurement of \hat{A}_0 , which selects $a_0 = b_0 = 0$, to Bob the measurement of \hat{B}_1 , which selects $a_1 = b_1 = 0$. Together, the two corresponding projections project on the solution; individually, they halve the projection on the solution.

Summing up, half of the bits of \mathbf{b} are randomly selected by Bob, the other half by Alice. We show that this means that Alice knows in advance 50% of the bits of \mathbf{b} . It suffices to note that states (5) through (8) are the original quantum algorithm—namely states (1) through (4)—with the quantum state relativized to the observer Alice in the sense of relational quantum mechanics [6]. By definition, initially Alice does not know the content of register B . To her, register B is in a maximally mixed state even if Bob has already measured \hat{B} .

The 2 bit entropy of this state—and of the overall quantum state (5)—represents Alice's ignorance of the value of \mathbf{b} . When Alice measures \hat{A} at the end of the algorithm, the quantum state (7) is projected on the solution eigenstate (8). This projection is random to Alice, it is actually on the value of \mathbf{b} chosen by Bob. The entropy of the quantum state goes to zero and Alice acquires full knowledge of the value of \mathbf{b} . Thus, the entropy of the relativized quantum state gauges Alice's ignorance of the value of \mathbf{b} throughout the execution of the algorithm.

As we have said before, when Alice measures \hat{A} at the end of the algorithm, half of the projection on the solution eigenstate is Alice's contribution to the random selection of the value of \mathbf{b} . We back evolve to the beginning of the quantum algorithm (to immediately after the permutation U_B) this halved projection, for example, the projection associated with measuring \hat{A}_0 and obtaining $a_0 = b_0 = 0$; we should apply U_{BA}^\dagger to the projection. This projects the initial state (6) on

$$\frac{1}{2\sqrt{2}}(e^{i\varphi_0}|00\rangle_B + e^{i\varphi_1}|01\rangle_B)(|00\rangle_A + |01\rangle_A + |10\rangle_A + |11\rangle_A), \quad (9)$$

halving the entropy of the state of register B . This means that Alice, before starting the algorithm and “after” this back-evolved half projection, knows that $b_0 = 0$, namely one of the two bits of the solution she will read in the future in register A .

We are at the level of elementary logical operations, where knowing means doing. Alice knows of the advanced information by acting like she knew it, namely by using it to identify classically the missing bit (the value of b_1) with a single computation of $\delta(\mathbf{b}, \mathbf{a})$. Correspondingly, as we showed in Refs. [3,4], the quantum algorithm is the superposition of all the possible ways of taking one bit of information about the solution and, given the advanced knowledge of this bit, classically identifying the missing bit with a single computation of $\delta(\mathbf{b}, \mathbf{a})$. This explains the speed-up from three computations to one computation.

We note that the entangled state (7) is the outcome of the unitary part of any quantum algorithm that starts with a maximally mixed state of register B and solves the database search problem, with or without a quantum speed-up. In fact, the quantum algorithm can do either without or with the advanced knowledge. In the former case, it is isomorphic with a classical algorithm that starts from the usual initial state and yields no speed-up. In the latter, it is isomorphic with a classical algorithm that starts from the initial state “after” the back-evolved half projection on the solution—thus with advanced knowledge of 50% of the bits of the solution.

The above explanation of the speed-up generalizes to $n > 2$ and to the very diverse quantum algorithms that yield an exponential speed-up. In all the cases examined, the quantum algorithm requires the number of function evaluations [computations of $\delta(\mathbf{b}, \mathbf{a})$ in Grover's case] of a classical algorithm that knows in advance 50% of the information about the solution. In a previous work [3], we called this *the 50% rule of the quantum algorithms*.

The 50% rule has a practical interest: It allows us to characterize the problems solvable with a quantum speed-up in an entirely computer science framework with no physics

involved—an important simplification. It should also allow us to identify new quantum speed-ups.

Section II highlights the mechanism of the speed-up in the case of Grover’s algorithm. In Sec. III, we check that the 50% rule holds for a class of quantum algorithms that yield an exponential speed-up. In Sec. IV, we develop a new quantum speed-up out of the 50% rule. In Sec. V we draw the conclusions.

II. THE MECHANISM OF THE QUANTUM SPEED-UP IN GROVER’S ALGORITHM

We develop in detail the line of thinking provided in the former section, first for $n = 2$ then for $n > 2$.

A. Extended representation of Grover’s algorithm

We relativize to Alice the quantum state of the original Grover’s algorithm. With $n = 2$, registers B and A are two qubits each. A one-qubit register V is meant to contain the result of the computation of $\delta(\mathbf{b}, \mathbf{a})$, modulo 2 added to its initial content for logical reversibility. Let us assume that Bob chose $\mathbf{b} = 00$; the initial state of the three registers is in any case:

$$|\psi\rangle = \frac{1}{4\sqrt{2}}(e^{i\varphi_0}|00\rangle_B + e^{i\varphi_1}|01\rangle_B + e^{i\varphi_2}|10\rangle_B + e^{i\varphi_3}|11\rangle_B) \times (|00\rangle_A + |01\rangle_A + |10\rangle_A + |11\rangle_A)(|0\rangle_V - |1\rangle_V). \quad (10)$$

The two-bit entropy of the state of register B represents Alice’s initial ignorance of Bob’s choice.

The computation of $\delta(\mathbf{b}, \mathbf{a})$ is performed in quantum parallelism on each term of the superposition. For example, the input $e^{i\varphi_1}|01\rangle_B|01\rangle_A|0\rangle_V$ means that the input of the computation of $\delta(\mathbf{b}, \mathbf{a})$ is $\mathbf{b} = 01$, $\mathbf{a} = 01$ and that the initial content of register V is 0. The computation yields $\delta(01, 01) = 1$ that, modulo 2 added to the initial content of V , yields the output $e^{i\varphi_1}|01\rangle_B|01\rangle_A|1\rangle_V$ (B and A keep the memory of the input for logical reversibility). Similarly, the input $-e^{i\varphi_1}|01\rangle_B|01\rangle_A|1\rangle_V$ goes into the output $-e^{i\varphi_1}|01\rangle_B|01\rangle_A|0\rangle_V$. More in general, the input $e^{i\varphi_b}|\mathbf{b}\rangle_B|\mathbf{a}\rangle_A(|0\rangle_V - |1\rangle_V)$ goes into the output $-e^{i\varphi_b}|\mathbf{b}\rangle_B|\mathbf{a}\rangle_A(|0\rangle_V - |1\rangle_V)$ if $\mathbf{a} = \mathbf{b}$, remains unaltered otherwise. In the overall, a single computation of $\delta(\mathbf{b}, \mathbf{a})$ sends state (10) into:

$$U_\delta|\psi\rangle = \frac{1}{4\sqrt{2}}[e^{i\varphi_0}|00\rangle_B(-|00\rangle_A + |01\rangle_A + |10\rangle_A + |11\rangle_A) + e^{i\varphi_1}|01\rangle_B(|00\rangle_A - |01\rangle_A + |10\rangle_A + |11\rangle_A) + e^{i\varphi_2}|10\rangle_B(|00\rangle_A + |01\rangle_A - |10\rangle_A + |11\rangle_A) + e^{i\varphi_3}|11\rangle_B(|00\rangle_A + |01\rangle_A + |10\rangle_A - |11\rangle_A)] \times (|0\rangle_V - |1\rangle_V), \quad (11)$$

a maximally entangled state where four orthogonal states of register B , each containing a single value of \mathbf{b} , are correlated with four orthogonal states of register A . To transform entanglement into correlation between measurement

outcomes, we apply to register A the unitary transformation U_A such that:

$$U_A U_\delta |\psi\rangle = \frac{1}{2\sqrt{2}}(e^{i\varphi_0}|00\rangle_B|00\rangle_A + e^{i\varphi_1}|01\rangle_B|01\rangle_A + e^{i\varphi_2}|10\rangle_B|10\rangle_A + e^{i\varphi_3}|11\rangle_B|11\rangle_A) \times (|0\rangle_V - |1\rangle_V). \quad (12)$$

We incidentally note that eliminating register V , like we did in Sec. I, does not alter the unitary character of the transformations. Measuring \hat{A} in state (12), projects it on the solution eigenstate:

$$\frac{1}{\sqrt{2}}|00\rangle_B|00\rangle_A(|0\rangle_V - |1\rangle_V), \quad (13)$$

yielding the eigenvalue $\mathbf{a} = 00$, namely the solution of the problem. Alice acquires full knowledge of the value of \mathbf{b} chosen by Bob and the entropy of the quantum state becomes zero. This entropy gauges Alice’s knowledge of the value of \mathbf{b} throughout the execution of the algorithm.

B. Back evolving 50% of the projection on the solution

We show the consequence of ascribing 50% of the determination of the value of \mathbf{b} to a partial measurement performed by Alice.¹ We adopt the example of Sec. I; we assume that the value of \mathbf{b} chosen by Bob is $\mathbf{b} = 00$ and that the partial measurement is that of \hat{A}_0 . This selects the eigenvalue $a_0 = 0$, projecting (12) on:

$$\frac{1}{2}(e^{i\varphi_0}|00\rangle_B|00\rangle_A + e^{i\varphi_1}|01\rangle_B|01\rangle_A)(|0\rangle_V - |1\rangle_V). \quad (14)$$

We back evolve this projection to the beginning of the quantum algorithm, by applying to state (14) the inverse of the unitary part of the algorithm, namely $U_\delta^\dagger U_A^\dagger$. This projects the initial state of the algorithm, (10), on:

$$\frac{1}{4}(e^{i\varphi_0}|00\rangle_B + e^{i\varphi_1}|01\rangle_B)(|00\rangle_A + |01\rangle_A + |10\rangle_A + |11\rangle_A)(|0\rangle_V - |1\rangle_V). \quad (15)$$

That the state of register B should have the form it has in (15) can be seen more directly as follows. We note that the unitary part of the quantum algorithm is the identity on the reduced density operator of register B that, in the random-phase representation, is $\rho_B = \frac{1}{2}(e^{i\varphi_0}|00\rangle_B + e^{i\varphi_1}|01\rangle_B + e^{i\varphi_2}|10\rangle_B + e^{i\varphi_3}|11\rangle_B)$ in both (10) and (12). By measuring \hat{A}_0 in state (12), Alice projects ρ_B on $\frac{1}{\sqrt{2}}(e^{i\varphi_0}|00\rangle_B + e^{i\varphi_1}|01\rangle_B)$. This projection goes back unaltered to the beginning of the algorithm.

State (15) says that, “after” back evolved projection, Alice knows in advance that the value of \mathbf{b} is either $\mathbf{b} = 00$ or $\mathbf{b} = 01$, namely that $b_0 = 0$. Correspondingly, the entropy representing Alice’s initial ignorance of the solution has decreased from two to one bit. How Alice utilizes this gain in information to achieve a speed-up is explained in the next section.

¹We should keep in mind that Alice’s measurement contributes to the random selection of a value of \mathbf{b} , then transformed into the value chosen by Bob by the unitary transformation U_B . Since this latter should be considered fixed from the standpoint of quantum correlation, we can say that Alice contributes to Bob’s choice.

C. Utilizing the back evolved half projections

By measuring \hat{A} Alice projects state (12) on the solution eigenstate. Let us go exhaustively through all the possible ways of halving this projection on the solution. Until now we have considered the binary observable \hat{A}_0 , whose measurement tells whether $\mathbf{b} \in \{00,01\}$ or $\mathbf{b} \in \{10,11\}$, and \hat{A}_1 , whose measurement tells whether $\mathbf{b} \in \{00,10\}$ or $\mathbf{b} \in \{01,11\}$. There is a third binary observable, say \hat{A}_+ , whose measurement tells whether $\mathbf{b} \in \{00,11\}$ or $\mathbf{b} \in \{01,10\}$. Measuring any pair of these three observables projects the output state (12) on the solution. Measuring any single observable halves the projection on the solution.

In the overall, there are six halved projections, on $\{00,01\}$, $\{10,11\}$, \dots , and $\{01,10\}$ —all the ways of taking a pair of elements of four. Each halved projection (actually, on an incoherent superposition of two values of \mathbf{b}) goes back unaltered to the beginning of the quantum algorithm, where it halves the entropy of Alice's state of knowledge of the value of \mathbf{b} , originating eight classical computation histories, as follows.

Let us start with the projection on $\mathbf{b} \in \{00,01\}$. In other words, Alice knows in advance that $\mathbf{b} \in \{00,01\}$. To identify the missing bit, she should compute $\delta(\mathbf{b}, \mathbf{a})$ for either $\mathbf{a} = 00$ or $\mathbf{a} = 01$. We assume that she does it for $\mathbf{a} = 00$; we are pinpointing one of the possible combinations. If the outcome of the computation is $\delta = 1$, this means that $\mathbf{b} = 00$. This originates two classical computation histories (represented as sequences of sharp quantum states), depending on the initial state of register V . History 1: initial state $e^{i\varphi_0}|00\rangle_B|00\rangle_A|0\rangle_V$, the state after the computation $e^{i\varphi_0}|00\rangle_B|00\rangle_A|1\rangle_V$. History 2: initial state $e^{i\varphi_0}|00\rangle_B|00\rangle_A|1\rangle_V$, the state after the computation $e^{i\varphi_0}|00\rangle_B|00\rangle_A|0\rangle_V$. If the outcome of the computation is $\delta = 0$, this means that $\mathbf{b} = 01$. This originates the other two histories. History 3: initial state $e^{i\varphi_1}|01\rangle_B|00\rangle_A|0\rangle_V$, the state after the computation $e^{i\varphi_1}|01\rangle_B|00\rangle_A|0\rangle_V$. History 4: initial state $e^{i\varphi_1}|01\rangle_B|00\rangle_A|1\rangle_V$, the state after the computation $e^{i\varphi_1}|01\rangle_B|00\rangle_A|1\rangle_V$. If she computes $\delta(\mathbf{b}, \mathbf{a})$ for $\mathbf{a} = 01$ instead, this originates the other four histories, and so on.

If we sum together all the different histories (some histories are originated more than once), each with a suitable phase, and normalize, we obtain the function evaluation stage of the quantum algorithm, namely the transformation of state (10) into (11).

This answers the question of how Alice knows of the *advanced information*—the information conveyed back by the back-evolved half projections on the solution. We are at the level of elementary logical operations, where “knowing” means “doing.” Alice knows of the advanced information by acting like she knew it, namely by computing on the basis of it the missing information. It should be noted that Alice could also ignore (do without) the advanced information, which simply means a quantum algorithm with no speed-up, isomorphic with a classical algorithm that starts from complete ignorance of the value of \mathbf{b} . An algorithm that yields a speed-up is isomorphic with a classical algorithm that starts from the back-evolved half projections on the solution.

The 50% rule only says that the quantum algorithm can be broken down into a superposition of classical computation histories that start from the advanced information; the history phases and the rotation of the basis of register A (i.e.,

U_A) after the computation of $\delta(\mathbf{b}, \mathbf{a})$ are what is needed for reconstructing the quantum algorithm. However, in Refs. [3,4], we have shown that the quantum algorithm can be synthesized from the advanced information classical algorithm (from the classical computation histories in quantum notation) through an optimization procedure. We should choose history phases and rotation of the basis of A in such a way that they maximize (i) entanglement between registers A and B after the computation of $\delta(\mathbf{b}, \mathbf{a})$ and (ii) the information about the solution readable in A at the end of the algorithm.

D. Quantum search for $n > 2$

Registers B and A are n qubits each. Register V is one qubit. Given the advanced knowledge of $n/2$ of the bits of the value of \mathbf{b} selected by Bob, in order to compute the missing $n/2$ bits, Alice should compute $\delta(\mathbf{b}, \mathbf{a})$ for all the values of \mathbf{a} in quantum superposition and apply to register A the appropriate unitary transformation U_A an $O(2^{n/2})$ times; each time U_A maximizes the entanglement between registers B and A . Eventually we obtain (approximately):

$$\frac{1}{2^{(n+1)/2}} \left(\sum_{\mathbf{c}=0}^{2^n-1} e^{i\varphi_{\mathbf{c}}} |\mathbf{c}\rangle_B |\mathbf{c}\rangle_A \right) (|0\rangle_V - |1\rangle_V). \quad (16)$$

Measuring either \hat{A} or \hat{B} , or both, projects (16) on the solution eigenstate. According to the rationale of the previous sections, we should halve the final projection on the solution in all possible ways; for example, by measuring $\hat{A}_0, \dots, \hat{A}_{\frac{n}{2}-1}$. Let \mathcal{I} be the information acquired by reading the solution at the end of the algorithm. Evidently, the considerations of the previous sections apply also here: Back evolving a half projection to the beginning of the quantum algorithm, makes available at the input of the computation the corresponding 50% of \mathcal{I} .

The fact that, for large n , the optimal number of times is $\frac{\pi}{4} 2^{n/2}$, not $2^{n/2}$, does not imply that Grover's algorithm outperforms the 50% rule. In fact this optimal number is associated with a nonzero probability— $O(1/2^n)$ —that the algorithm delivers a wrong solution. One should look for the possible cases where Grover's algorithm yields the solution with certainty, like in the case $n = 2$.

III. CHECKING THE 50% RULE ON OTHER QUANTUM ALGORITHMS

Until now we have discussed the 50% rule on Grover's algorithm. It is therefore important to check that the rule holds for the very diverse quantum algorithms that yield an exponential speed-up. In many of these algorithms, there is a set of functions $f_{\mathbf{b}} : \{0,1\}^n \rightarrow \{0,1\}^m$ known to both Alice and Bob. Bob selects a value of \mathbf{b} and Alice should find a character of the function $f_{\mathbf{b}}$ by computing $f_{\mathbf{b}}(\mathbf{a})$ for various values of \mathbf{a} . Since the problems addressed by such algorithms are structured, identifying the advanced information and sharing out the projection on the solution between Alice and Bob requires some care. With respect to the similar section of Ref. [4], the present one provides various clarifications.

A. Deutsch and Jozsa’s algorithm

In Deutsch and Jozsa’s [7] algorithm, the set of functions known to both Bob and Alice is all the constant and “balanced” functions (with an even number of zeros and ones) $f_{\mathbf{b}}$:

\mathbf{a}	$f_{0000}(\mathbf{a})$	$f_{1111}(\mathbf{a})$	$f_{0011}(\mathbf{a})$	$f_{1100}(\mathbf{a})$	$f_{0101}(\mathbf{a})$	$f_{1010}(\mathbf{a})$	$f_{0110}(\mathbf{a})$	$f_{1001}(\mathbf{a})$
00	0	1	0	1	0	1	0	1
01	0	1	0	1	1	0	1	0
10	0	1	1	0	0	1	1	0
11	0	1	1	0	1	0	0	1

Alice should find whether the function selected by Bob is balanced or constant by computing $f_{\mathbf{b}}(\mathbf{a}) = f(\mathbf{b}, \mathbf{a})$. In the classical case this requires, in the worst case, a number of computations of $f(\mathbf{b}, \mathbf{a})$ exponential in n ; in the quantum case one computation.

The initial state of the algorithm relativized to Alice is:

$$\frac{1}{2\sqrt{2}}(\rho_0 e^{i\varphi_0} |0000\rangle_B + \rho_1 e^{i\varphi_1} |1111\rangle_B + \rho_2 e^{i\varphi_2} |0011\rangle_B + \rho_3 e^{i\varphi_3} |1100\rangle_B + \dots)(|00\rangle_A + |01\rangle_A + |10\rangle_A + |11\rangle_A)(|0\rangle_V - |1\rangle_V). \tag{18}$$

The coefficients $\rho_i > 0$, such that $\sum \rho_i^2 = 1$, account for a nonflat probability distribution of Bob’s selection. Things will be simpler if we assume that the ρ_i are the same for dual values of \mathbf{b} , like 0000 and 1111. Modulo 2 adding the result of the computation of $f(\mathbf{b}, \mathbf{a})$ to the content of V and performing the Hadamard transform on register A yields the entangled state:

$$\frac{1}{\sqrt{2}}[(\rho_0 e^{i\varphi_0} |0000\rangle_B - \rho_1 e^{i\varphi_1} |1111\rangle_B)|00\rangle_A + (\rho_2 e^{i\varphi_2} |0011\rangle_B - \rho_3 e^{i\varphi_3} |1100\rangle_B)|10\rangle_A + \dots] \times (|0\rangle_V - |1\rangle_V). \tag{19}$$

Measuring \hat{B} and \hat{A} in (19) yields Bob’s selection of a value of \mathbf{b} and the solution found by Alice: all zeros if the function is constant and not so if it is balanced.

We check that the quantum algorithm requires the number of function evaluations of a classical algorithm that knows in advance 50% of \mathcal{I} ; we call \mathcal{I} the information acquired by reading the solution at the end of the algorithm. Since the solution is a function of \mathbf{b} , we can define the advanced information as any 50% of the information about the solution contained in \mathbf{b} , namely in the table of $f_{\mathbf{b}}(\mathbf{a})$. If $f_{\mathbf{b}}(\mathbf{a})$ is constant, for reasons of symmetry, the advanced information is any 50% of the table of the function; see Array (17). If the function is balanced, still for reasons of symmetry, it is any 50% of the table that does not contain different values of the function; for each balanced function there are two such half tables. In fact, the half tables that contain different values of the function already indicate that the function is balanced and thus contain 100% of \mathcal{I} . For the good half tables, that do not contain different values of the function, the solution (whether the function is constant or balanced) is always identified by computing $f_{\mathbf{b}}(\mathbf{a})$ for only one value of \mathbf{a} (any one) outside

$\{0,1\}^n \rightarrow \{0,1\}$. Array (17) gives this set for $n = 2$. The string $\mathbf{b} \equiv b_0, b_1, \dots, b_{2^n-1}$ is both the suffix and the table of the function—the sequence of function values for increasing values of the argument.

the half table. Thus, both the quantum algorithm and the advanced information classical algorithm require just one function evaluation.

We should note that the present definition of advanced information:

(i) Could be applied as well to Grover’s algorithm, where it becomes any 50% of the table of $\delta(\mathbf{b}, \mathbf{a})$ (for a given value of \mathbf{b}) that does not contain the value $\delta(\mathbf{b}, \mathbf{a}) = 1$. All the results of Sec. II would remain unaltered.

(ii) Identifies a back-evolved half projection on the solution. In fact (disregarding the sign of the random-phase factors that is irrelevant) the reduced density operator of register B in the random-phase representation is:

$$\rho_B = \rho_0 e^{i\varphi_0} |0000\rangle_B + \rho_1 e^{i\varphi_1} |1111\rangle_B + \rho_2 e^{i\varphi_2} |0011\rangle_B + \rho_3 e^{i\varphi_3} |1100\rangle_B + \dots \tag{20}$$

throughout the unitary part of the quantum algorithm. We assume that the advanced information (a good half table) is $f(\mathbf{b}, 00) = 0$ and $f(\mathbf{b}, 01) = 0$. This means that the function selected by Bob is either $f_{0000}(\mathbf{a})$ or $f_{0011}(\mathbf{a})$; see Array (17). This corresponds to projecting ρ_B on $\rho'_B = (\rho_0 e^{i\varphi_0} |0000\rangle_B + \rho_2 e^{i\varphi_2} |0011\rangle_B)$, up to normalization; this outcome goes back unaltered to the beginning of the quantum algorithm, where it becomes Alice’s advanced knowledge of the solution. We should note that Alice, by measuring \hat{A}_1 in state (19) and finding $a_1 = 0$, projects ρ_B not on ρ'_B but on:

$$\rho_B = \rho_0 e^{i\varphi_0} |0000\rangle_B + \rho_1 e^{i\varphi_1} |1111\rangle_B + \rho_2 e^{i\varphi_2} |0011\rangle_B + \rho_3 e^{i\varphi_3} |1100\rangle_B, \tag{21}$$

up to normalization. To project (21) on ρ'_B , Bob should measure a single \hat{B}_i , e.g., \hat{B}_0 , thus finding in present assumptions $b_0 = 0$. This latter projection, although performed by Bob, can be added to Alice’s advanced knowledge of the solution. In fact it selects between dual values of \mathbf{b} , which does not disclose to Alice any information about the solution and does not affect the entropy of the reduced density operator of register A in state (19).

This time, sharing out the projection on the solution between Alice and Bob would be more complex, because of the asymmetry between the two actions. However, we can bypass this difficulty. It suffices to note that, with all the $\rho_i > 0$, state (19) is certainly entangled. Thus, in the present criteria, there is in any case a nonzero contribution to the determination of the value of \mathbf{b} on the part of both Alice and Bob. This is

enough to see that the advanced information available to Alice cannot exceed 50% of \mathcal{I} . In fact, increasing it over 50% would mean increasing any good half table by one row, which would project the output state (19) on the solution, leaving to Bob nothing to project. Summing up, we have ascertained that Alice’s advanced information is back-evolved projection and that it is (and cannot exceed) 50% of \mathcal{I} .

Now we go to the history superposition picture. Let us assume that the advanced information is $f(\mathbf{b},00) = 0$ and $f(\mathbf{b},01) = 0$. Alice can find the value of \mathbf{b} (thus the character of the function), by performing function evaluation for either $\mathbf{a} = 10$ or $\mathbf{a} = 11$. We assume that she does it for $\mathbf{a} = 10$. If the result of the computation is 0, this means that $\mathbf{b} = 0000$. This originates two classical computation histories in quantum notation: history 1: initial state $\rho_0 e^{i\varphi_0} |0000\rangle_B |10\rangle_A |0\rangle_V$, the state after the computation $\rho_0 e^{i\varphi_0} |0000\rangle_B |10\rangle_A |0\rangle_V$; history 2: initial state $\rho_0 e^{i\varphi_0} |0000\rangle_B |10\rangle_A |1\rangle_V$, the state after the computation $\rho_0 e^{i\varphi_0} |0000\rangle_B |10\rangle_A |1\rangle_V$. If the result of the computation is 1, this means that $\mathbf{b} = 0011$. This originates two histories: history 3: initial state $\rho_2 e^{i\varphi_2} |0011\rangle_B |10\rangle_A |0\rangle_V$, the state after the computation $\rho_2 e^{i\varphi_2} |0011\rangle_B |10\rangle_A |1\rangle_V$; history 4: initial state $\rho_2 e^{i\varphi_2} |0011\rangle_B |10\rangle_A |1\rangle_V$, the state after the computation

$\rho_2 e^{i\varphi_2} |0011\rangle_B |10\rangle_A |0\rangle_V$. If she performs function evaluation for $\mathbf{a} = 11$ instead, this originates other four histories, and so on. If we sum together all the different histories, each with a suitable phase, and normalize, we obtain the function evaluation stage of the quantum algorithm.

To obtain the quantum algorithm, we should choose history phases and the final unitary transformation applied to register A in such a way that the information about the solution readable in that register at the end of the algorithm is maximized.

B. Simon’s and the hidden subgroup algorithms

In Simon’s [8] algorithm, the set of functions is all the $f_{\mathbf{b}} : \{0,1\}^n \rightarrow \{0,1\}^{n-1}$ such that $f_{\mathbf{b}}(\mathbf{a}) = f_{\mathbf{b}}(\mathbf{c})$ if and only if $\mathbf{a} = \mathbf{c}$ or $\mathbf{a} = \mathbf{c} \oplus \mathbf{h}^{(\mathbf{b})}$; \oplus denotes bitwise modulo 2 addition; the bit string $\mathbf{h}^{(\mathbf{b})} \equiv h_0^{(\mathbf{b})}, h_1^{(\mathbf{b})}, \dots, h_{n-1}^{(\mathbf{b})}$, depending on \mathbf{b} and belonging to $\{0,1\}^n$ excluded the all zeros string, is a sort of period of the function. Array (22) gives the set of functions for $n = 2$. The bit string \mathbf{b} is both the suffix and the table of the function. Since $\mathbf{h}^{(\mathbf{b})} \oplus \mathbf{h}^{(\mathbf{b})} = 0$, each value of the function appears exactly twice in the table, thus 50% of the rows plus one surely identify $\mathbf{h}^{(\mathbf{b})}$.

	$\mathbf{h}^{(0011)} = 01$	$\mathbf{h}^{(1100)} = 01$	$\mathbf{h}^{(0101)} = 10$	$\mathbf{h}^{(1010)} = 10$	$\mathbf{h}^{(0110)} = 11$	$\mathbf{h}^{(1001)} = 11$	
\mathbf{a}	$f_{0011}(\mathbf{a})$	$f_{1100}(\mathbf{a})$	$f_{0101}(\mathbf{a})$	$f_{1010}(\mathbf{a})$	$f_{0110}(\mathbf{a})$	$f_{1001}(\mathbf{a})$	
00	0	1	0	1	0	1	(22)
01	0	1	1	0	1	0	
10	1	0	0	1	1	0	
11	1	0	1	0	0	1	

Bob selects a value of \mathbf{b} . Alice’s problem is finding the value of $\mathbf{h}^{(\mathbf{b})}$, “hidden” in $f_{\mathbf{b}}(\mathbf{a})$, by computing $f_{\mathbf{b}}(\mathbf{a}) = f(\mathbf{b},\mathbf{a})$ for different values of \mathbf{a} . In present knowledge, a classical algorithm requires a number of computations of $f(\mathbf{b},\mathbf{a})$ exponential in n . The quantum algorithm solves the hard part of this problem, namely finding a string $\mathbf{s}_j^{(\mathbf{b})}$ orthogonal² to $\mathbf{h}^{(\mathbf{b})}$, with one computation of $f(\mathbf{b},\mathbf{a})$. There are 2^{n-1} such strings. Running the quantum algorithm yields one of these strings at random (see further below). The quantum algorithm is iterated until finding $n - 1$ different strings. This allows us to find $\mathbf{h}^{(\mathbf{b})}$ by solving a system of modulo 2 linear equations. Register B is now $2^n (n - 1)$ -qubit, given that \mathbf{b} is the sequence of 2^n fields each on $n - 1$ bits.

The initial state of the algorithm relativized to Alice, with register V prepared in the all zeros string (just one zero for $n = 2$), is:

$$\frac{1}{2}(\rho_0 e^{i\varphi_0} |0011\rangle_B + \rho_1 e^{i\varphi_1} |1100\rangle_B + \rho_2 e^{i\varphi_2} |0101\rangle_B + \rho_3 e^{i\varphi_3} |1010\rangle_B + \dots)(|00\rangle_A + |01\rangle_A + |10\rangle_A + |11\rangle_A)|0\rangle_V. \tag{23}$$

Computing $f(\mathbf{b},\mathbf{a})$, which changes the content of V from zero to the outcome of the computation, and performing the Hadamard transform on register A yields:

$$\frac{1}{2}\{(\rho_0 e^{i\varphi_0} |0011\rangle_B + \rho_1 e^{i\varphi_1} |1100\rangle_B)[(|00\rangle_A + |10\rangle_A)|0\rangle_V + (|00\rangle_A - |10\rangle_A)|1\rangle_V] + (\rho_2 e^{i\varphi_2} |0101\rangle_B + \rho_3 e^{i\varphi_3} |1010\rangle_B)[(|00\rangle_A + |01\rangle_A)|0\rangle_V + (|00\rangle_A - |01\rangle_A)|1\rangle_V] + \dots\}, \tag{24}$$

where, for each value of \mathbf{b} , register A (no matter the content of V) hosts even weighted superpositions of the 2^{n-1} strings $\mathbf{s}_j^{(\mathbf{b})}$ orthogonal to $\mathbf{h}^{(\mathbf{b})}$. By measuring \hat{A} and \hat{B} in state (24), we obtain at random Bob’s selection of \mathbf{b} and one of the $\mathbf{s}_j^{(\mathbf{b})}$.

We leave B in its after-measurement state, thus fixing the value of \mathbf{b} , and iterate the “right part” of the algorithm (preparation of registers A and V , computation of $f(\mathbf{b},\mathbf{a})$, and measurement of \hat{A}) until obtaining $n - 1$ different $\mathbf{s}_j^{(\mathbf{b})}$.

We check that the quantum algorithm requires the number of function evaluations of a classical algorithm that knows in advance 50% of \mathcal{I} . Any $\mathbf{s}_j^{(\mathbf{b})}$ is a solution of the problem addressed by the quantum part of Simon’s algorithm. The advanced information is any 50% of the information about the solution contained in \mathbf{b} . For reasons of symmetry, this is any 50% of the table of the function that does not contain the same value of the function twice. In fact, the half tables that contain a same value twice already specify the value of $\mathbf{h}^{(\mathbf{b})}$

²The modulo 2 addition of the bits of the bitwise product of the two strings should be zero.

and thus the value of any $s_j^{(b)}$. For the half tables that do not contain the same value of the function twice, the solution is always identified by computing $f(\mathbf{b}, \mathbf{a})$ for only one value of \mathbf{a} (any one) outside the half table. The new value of the function is necessarily a value already present in the half table, which identifies $\mathbf{h}^{(b)}$ and thus all the $s_j^{(b)}$. Thus, both the quantum algorithm and the advanced information classical algorithm require just one function evaluation. As in Sec. III A, the above defined advanced information is back evolved projection on the solution and cannot exceed 50% of \mathcal{I} .

Now we go to the history superposition picture. For example, let us assume that the advanced information is $f(\mathbf{b}, 00) = 0$ and $f(\mathbf{b}, 11) = 1$, namely the first and last row of either $f_{0011}(\mathbf{a})$ or $f_{0101}(\mathbf{a})$; see Array (22). To find which is the case, Alice should perform a function evaluation for either $\mathbf{a} = 01$ or $\mathbf{a} = 10$. We assume she does it for $\mathbf{a} = 01$. If the result of the computation is 0, this means that $\mathbf{b} = 0011$. This originates two classical computation histories in quantum notation: history 1: initial state $\rho_0 e^{i\varphi_0} |0011\rangle_B |01\rangle_A |0\rangle_V$, the state after the computation $\rho_0 e^{i\varphi_0} |0011\rangle_B |01\rangle_A |0\rangle_V$; history 2: initial state $\rho_0 e^{i\varphi_0} |0011\rangle_B |01\rangle_A |1\rangle_V$, the state after the computation $\rho_0 e^{i\varphi_0} |0011\rangle_B |01\rangle_A |1\rangle_V$. If the result of the computation is 1, this means that $\mathbf{b} = 0101$. This originates two histories: history 3: initial state $\rho_2 e^{i\varphi_2} |0101\rangle_B |01\rangle_A |0\rangle_V$, the state after the computation $\rho_2 e^{i\varphi_2} |0101\rangle_B |01\rangle_A |1\rangle_V$; history 4: initial state $\rho_2 e^{i\varphi_2} |0101\rangle_B |01\rangle_A |1\rangle_V$, the state after the computation $\rho_2 e^{i\varphi_2} |0101\rangle_B |01\rangle_A |0\rangle_V$. If she performs function evaluation for $\mathbf{a} = 10$ instead, this originates other four histories, and so on. If we sum together all the different histories, each with a suitable phase, and normalize, we obtain the function evaluation stage of the quantum algorithm. To obtain the quantum algorithm, we should choose history phases and the final unitary transformation applied to register A in such a way that the information about the solution readable in that register at the end of the algorithm is maximized.

The 50% rule also applies to the generalized Simon's problem and to the hidden subgroup problem. In fact the corresponding algorithms are essentially the same as the algorithm that solves Simon's problem. In the hidden subgroup problem, the set of functions $f_{\mathbf{b}} : G \rightarrow W$ map a group G to some finite set W with the property that there exists some subgroup $S \leq G$ such that for any $\mathbf{a}, \mathbf{c} \in G$, $f_{\mathbf{b}}(\mathbf{a}) = f_{\mathbf{b}}(\mathbf{c})$ if and only if $\mathbf{a} + S = \mathbf{c} + S$. The problem is to find the hidden subgroup S by computing $f_{\mathbf{b}}(\mathbf{a})$ for various values of \mathbf{a} . Now, a large variety of problems solvable with a quantum speed-up can be reformulated in terms of the hidden subgroup problem [9]. Among these we find Deutsch's problem, finding orders, finding the period of a function (thus the problem solved by the quantum part of Shor's factorization algorithm), discrete logarithms in any group, hidden linear functions, self-shift equivalent polynomials, the Abelian stabilizer problem, and the graph automorphism problem.

IV. APPLYING THE 50% RULE TO THE SEARCH OF QUANTUM SPEED-UPS

In hindsight, the quantum algorithms examined are skillfully designed around the 50% rule. In unstructured database search, the advanced knowledge of 50% of the solution yields

a quadratic speed-up, given that the number of function evaluations goes from $O(2^n)$ to $O(2^{n/2})$. Thus, the possibility of a quadratic speed-up is established by the 50% rule; one does not need to know Grover's algorithm. Similarly, in the structured algorithms that yield an exponential speed-up, the problem is chosen in such a way that, if one knows in advance 50% of the rows of the table of the function, computing $f_{\mathbf{b}}(\mathbf{a})$ for a single value of \mathbf{a} outside the half table yields the solution. Thus, the possibility of an exponential speed-up is established by the 50% rule before knowing the quantum algorithm.

One way of searching for new quantum speed-ups is thus looking for problems solvable with a single computation of $f_{\mathbf{b}}(\mathbf{a})$ once that 50% of the rows of the table of the function are known. We provide an example; see also Ref. [4]. The set of functions is the 4! functions $f_{\mathbf{b}} : \{0, 1\}^2 \rightarrow \{0, 1\}^2$ such that the sequence of function values is a permutation of the values of the argument.

\mathbf{a}	$f_{00011110}(\mathbf{a})$	$f_{00110110}(\mathbf{a})$	$f_{00011011}(\mathbf{a})$	\dots
00	00	00	00	\dots
01	01	11	01	\dots
10	11	01	10	\dots
11	10	10	11	\dots

The string \mathbf{b} is both the suffix and the table of the function—the sequence of function values for increasing values of the argument. We have chosen this set because, if we know 50% of the rows of one table, we can identify the corresponding value of \mathbf{b} with a single computation of $f_{\mathbf{b}}(\mathbf{a})$. Without advanced information, three computations of $f_{\mathbf{b}}(\mathbf{a})$ are required. Thus there is room for a speed-up. We build a quantum algorithm over this possibility. Register B is eight qubits, register A is two qubits, and register V is two qubits, denoted V_0 and V_1 . The result of the computation of $f_{\mathbf{b}}(\mathbf{a}) = f(\mathbf{b}, \mathbf{a})$ is bitwise modulo 2 added to the former content of V . The initial state is:

$$\frac{1}{8\sqrt{6}}(e^{i\varphi_0} |00011110\rangle_B + e^{i\varphi_1} |00110110\rangle_B + e^{i\varphi_2} |00011011\rangle_B \dots)(|00\rangle_A + |01\rangle_A + |10\rangle_A + |11\rangle_A)(|0\rangle_{V_0} - |1\rangle_{V_0})(|0\rangle_{V_1} - |1\rangle_{V_1}).$$

Computing $f(\mathbf{b}, \mathbf{a})$, then performing the Hadamard transform on register A , yields

$$\frac{1}{4\sqrt{6}}[(e^{i\varphi_0} |00011110\rangle_B + \dots)|01\rangle_A + (e^{i\varphi_1} |00110110\rangle_B + \dots)|10\rangle_A + (e^{i\varphi_2} |00011011\rangle_B + \dots)|11\rangle_A] \times (|0\rangle_{V_0} - |1\rangle_{V_0})(|0\rangle_{V_1} - |1\rangle_{V_1}),$$

an entangled state where three orthogonal states of B (each a superposition of eight values of \mathbf{b} , corresponding to a partition of the set of 24 functions) are correlated with, respectively, $|01\rangle_A$, $|10\rangle_A$, and $|11\rangle_A$. Measuring \hat{A} in the above state tells which of the three partitions the function belongs to. In the case of a classical algorithm, identifying the partition requires three computations of $f(\mathbf{b}, \mathbf{a})$, as readily checked. There is thus a quantum speed-up.

With the 50% rule, one can figure out any number of these speed-ups in terms of number of function evaluations. Thus,

this rule provides a playground for studying the engineering of quantum algorithms.

V. CONCLUSION

Summarizing, moving from classical to quantum problem solving, the classical problem-solution correlation becomes quantum. There is quantum correlation between the selection of an eigenvalue of \hat{B} on the part of Bob and that of an eigenvalue of \hat{A} —the solution—on the part of Alice. The random selection of an eigenvalue of \hat{B} is required to set register B in a known eigenstate and then put it in the desired eigenstate by means of a permutation of the basis vectors of B . From the standpoint of correlation, which is defined on repetitions of the same quantum experiment, this permutation should be considered fixed: the fact that Bob can change it to always obtain the desired value of \mathbf{b} belongs to a different film. Because of quantum correlation, all is like Alice contributed to selecting 50% of the information that specifies the problem. As the solution is a function of the problem, this becomes Alice knowing in advance 50% of the information that specifies the solution.

The fact that the quantum speed-up comes from comparing two classical algorithms, with and without advanced information, has a practical interest. It allows us to characterize the problems solvable with a quantum speed-up in an entirely computer science framework, with no physics involved—an important simplification. It should also allow us to identify new quantum speed-ups, as exemplified in Sec. IV.

The fact that quantum algorithms are faster because they know in advance 50% of the solution they will themselves produce in the future has an obvious interest from the standpoint of the philosophy of quantum mechanics. Future work should aim to check the 50% rule for all quantum algorithms found so far, to possibly demonstrate it in a more general way, for example, for the generic quantum computational network or quantum Turing machine, and to explore the quantum speed-ups achievable on the basis of it.

ACKNOWLEDGMENTS

The author thanks David Finkelstein for useful discussions and David Deutsch, Tom Toffoli, and Lev Vaidman for useful comments.

-
- [1] D. Gross, S. T. Flammia, and J. Eisert, *Phys. Rev. Lett.* **102**, 190501 (2009).
 - [2] G. Castagnoli and D. Finkelstein, *Proc. Roy. Soc. London A* **457**, 1799 (2001).
 - [3] G. Castagnoli, *Int. J. Theor. Phys.* **48**, 2412 (2009).
 - [4] G. Castagnoli, *Int. J. Theor. Phys.* **48**, 3383 (2009).
 - [5] L. K. Grover, in *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing, Philadelphia, PA, May 22–24, 1996* (ACM Press, New York, 1996), p. 212.
 - [6] C. Rovelli, *Int. J. Theor. Phys.* **35**, 1637 (1996).
 - [7] D. Deutsch and R. Jozsa, *Proc. R. Soc. London A* **439**, 553 (1992).
 - [8] D. R. Simon, in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (IEEE Computer Society Press, Los Alamitos, CA, 1994), p. 116.
 - [9] P. Kaye, R. Laflamme, and M. Mosca, *An Introduction to Quantum Computing* (Oxford University Press, Oxford, 2007).