Int J Theor Phys (2009) 48: 3383–3395
DOI 10.1007/s10773-009-0143-6

Quantum Algorithms Know in Advance 50% of the Solution They Will Find in the Future

Giuseppe Castagnoli

Received: 12 July 2009 / Accepted: 7 September 2009 / Published online: 22 September 2009

Abstract Quantum algorithms require less operations than classical algorithms. The exact reason of this has not been pinpointed until now. Our explanation is that quantum algorithms know in advance 50% of the solution of the problem they will find in the future. In fact they can be represented as the sum of all the possible histories of a respective "advanced information classical algorithm". This algorithm, given the advanced information (50% of the bits encoding the problem solution), performs the operations (oracle's queries) still required to identify the solution. Each history corresponds to a possible way of getting the advanced information and a possible result of computing the missing information. This explanation of the quantum speed up has an immediate practical consequence: the speed up comes from comparing two classical algorithms, with and without advanced information, with no physics involved. This simplification could open the way to a systematic exploration of the possibilities of speed up.

Keywords Quantum computation · Quantum algorithm · Quantum speed up · Quantum entanglement · Quantum measurement

1 Introduction

By integrating a set of notions developed in the series of articles [2–4], and [5], we provide a simple and self contained explanation of the quantum speed up.

This should answer an existing need: Gross et al. assert that the exact "reason" of the quantum speed up was never pinpointed, [9], (2009). Grover, with reference to a search in a database of size N, writes: "What is the reason that one would expect that a quantum mechanical scheme could accomplish the search in $\mathrm{O}(\sqrt{N})$ steps? It would be insightful to have a simple two line argument for this without having to describe the details of the search algorithm", [11], (2001).

G. Castagnoli (✉)
Pieve Ligure, Genoa, Italy
e-mail: giuseppe.castagnoli@gmail.com

The explanation set forth in this article, in two lines, is: quantum algorithms require a lower number of operations because they know in advance 50% of the information about the solution of the problem they will find in the future. The peculiar character of this explanation has to do with the non-sequential behavior of the wave function, already highlighted by Dolev and Elitzur in special interaction free measurement situations [8]. Here we show that this non-sequentiality is the crux of the quantum speed up.

In the sequel, quantum problem solving is seen as a game between two players: the oracle and the quantum algorithm. The oracle chooses a function out of a set of functions known to both players and gives to the second player the black box for its computation. The second player should find out a certain property of the function through function evaluation (oracle's query).

We show that a quantum algorithm: (i) requires the number of function evaluations of a classical algorithm that knows in advance 50% of the information about solution of the problem and correspondingly (ii) can be represented as the sum of all the possible histories of this classical algorithm—each history corresponds to a possible way of getting the advanced information and a possible result of computing the missing information. Thus the speed up comes from comparing two classical algorithms, with and without advanced information. This brings the characterization of the problems liable of being solved with a quantum speed up to an entirely classical framework.

2 Advanced Knowledge

We derive the *50% rule*—point (i) of the introduction—in a simple instance of Grover's data base search algorithm [10].

The oracle chooses a data base location—an n bit string $\mathbf{k} \equiv k_0, k_1, \ldots, k_{n-1} \in \{0,1\}^n$ (hides a ball in drawer number $\mathbf{k}$)—and gives to the second player the black box that computes the Kronecker function $\delta(\mathbf{k}, x)$. The second player has to find the value of $\mathbf{k}$ (the number of the drawer the ball is in) by computing $\delta(\mathbf{k}, x)$ for different values of x (by opening different drawers). We also say: "by evaluating the function $\delta(\mathbf{k}, x)$".

The key step of our approach is representing together the production of the problem on the part of the oracle and the production of the solution on the part of the algorithm. We ideally add to the usual quantum registers X (containing the argument of the function to query the black box with) and V (hosting the result of function evaluation, mod 2 added to its former content for logical reversibility) an auxiliary input register K containing $\mathbf{k}$, the data base location chosen by the oracle. The extended algorithm is: (0) prepare K in the (even weighted) superposition of all the values of $\mathbf{k}$, X in the superposition of all the values of x, and V in the antisymmetric state, (1) perform function evaluation and mod 2 add the result to the former content of V, and (2) apply the transformation U (see further below) to register X.

With $n = 2$, the initial state is:

$$\Psi_0 = \frac{1}{4\sqrt{2}}(|00\rangle_K + |01\rangle_K + |10\rangle_K + |11\rangle_K)(|00\rangle_X + |01\rangle_X + |10\rangle_X + |11\rangle_X)(|0\rangle_V - |1\rangle_V). \tag{1}$$

The superposition in K can indifferently be incoherent, in which case $|00\rangle_K$ should be replaced by $\mathrm{e}^{i\delta_{00}}\,|00\rangle_K$, with δ_{00} a random variable with uniform distribution in $[0, 2\pi]$, etc.

One function evaluation yields:

$$\Psi_1 = \frac{1}{4\sqrt{2}} \begin{bmatrix} |00\rangle_K(-|00\rangle_X + |01\rangle_X + |10\rangle_X + |11\rangle_X) \\ + |01\rangle_K(|00\rangle_X - |01\rangle_X + |10\rangle_X + |11\rangle_X) \\ + |10\rangle_K(|00\rangle_X + |01\rangle_X - |10\rangle_X + |11\rangle_X) \\ + |11\rangle_K(|00\rangle_X + |01\rangle_X + |10\rangle_X - |11\rangle_X) \end{bmatrix} (|0\rangle_V - |1\rangle_V), \tag{2}$$

namely four orthogonal states of K, each a value of **k**, correlated with four orthogonal states of X, which means that the information about the value of **k** has propagated to register X. A rotation of the measurement basis of X makes this information readable. One applies to X: (i) Hadamard transform, (ii) the transformation obtained by computing $\delta(0,x)$, and (iii) another time Hadamard transform (in the overall, the transformation U). This yields:

$$\Psi_2 = \frac{1}{2\sqrt{2}}(|00\rangle_K|00\rangle_X + |01\rangle_K|01\rangle_X + |10\rangle_K|10\rangle_X + |11\rangle_K|11\rangle_X)(|0\rangle_V - |1\rangle_V), \tag{3}$$

an entangled state where each value of **k** (in K) is correlated with the corresponding solution found by the second player: the same value of **k** in X.

We denote by $[K]$ and $[X]$ the contents of K and X. Measurement of $[K]$ and $[X]$ in (3) determines the moves of both players: the oracle's choice (the value of **k**) and the solution provided by the second player. Note that the state reduction induced by measuring $[K]$, backdated to before running the algorithm, yields the original Grover's algorithm.

We discuss the behavior of the state vector. We see the nondeterministic production of the contents of the two registers, due to measuring $[K]$ and $[X]$ in (3), as *mutual determination* between such contents, like between two polarizations measured in an entangled polarization state (mutual determination, or *mutual causality*, is between bits of information). The precise meaning of "mutual" is specified by the following use of the term (see also [2–4]):

We cannot say that reading the content of K (i.e. the outcome of measuring $[K]$) at the end of the algorithm causes the content of X (the outcome of measuring $[X]$), namely that choosing the drawer number (a value of **k**) to hide the ball in on the part of the oracle determines the drawer number the ball is found in by the second player—this is the classical perspective with no mutual determination.

For the same reason we cannot say that reading the content of X at the end of the algorithm causes the content of K, namely that reading the drawer number at the end of the algorithm, on the part of the second player, determines the drawer number chosen by the oracle, namely creates the ball in the drawer with that number.

In consonance with time-symmetric quantum theory, we assume that mutual causality is symmetrical [2]. Thus, the content of the two registers is determined by reading the first (second) bit of register K and the second (first) bit of register X. In this perspective, one bit of the data base location is created by the oracle (by the action of measuring either bit of K), the other bit by the action of reading, at the end of the algorithm and on the part of the second player, the other bit of the data base location in register X (i.e. by the action of measuring the other bit of X). It is important to notice that this other bit is the ball created in that bit. Thus, the second player (the quantum algorithm) has to search only the bit created by the oracle, which explains the speed up of Grover's algorithm for $n = 2$.

Mutual causality is, in a different perspective, *mutual knowledge*, or *advanced knowledge* (in the polarization example, the two outcomes of measurement would "know each other"). We should think of backdating to before running the algorithm the reduction induced by measuring $[K]$. To the second player (to the algorithm), this is indistinguishable from having a $[K]$ measured before running the algorithm—see (1) through (3)—thus to having a predetermined $\mathbf{k}$. Now the second player, by measuring $[X]$ at the end of the algorithm, does not "create" any bit of information, he just "finds" the two measurement outcomes created by the oracle. Mutual causality between the two bits becomes the second player knowing in advance, before running the algorithm, either one of the two bits that he will read at the end of the algorithm. In other words, the algorithm knows in advance, before running, 50% of the information about the solution it will produce at the end of the run.

Either form of mutual causality explains the structure of the quantum algorithm: the quantum algorithm can be represented as the sum of all the possible histories of a classical algorithm that, knowing in advance 50% of the information about the solution of the problem, performs the function evaluations still required to identify the solution. As clarified in the sequel, each history corresponds to a possible way of getting the advanced information and a possible result of computing the missing information.

We should make a specification. In Grover's algorithm, the outcomes of measuring $[K]$ and $[X]$ in (3) are identical. The advanced information is indifferently 50% of the content of either register. This is not always the case in quantum algorithms, therefore we define the advanced information, in relation to the measurement outcomes, in a more general way. Since the content of X is a function of the content of K (the solution is a function of the problem), the information contained in X is redundant. Thus, the information encoding the solution of the problem is all contained in register K, namely in the bit string $\mathbf{k}$. There is advanced knowledge of any half of $\mathbf{k}$.

In the following, for each one of the main quantum algorithms, we: (i) provide the extended representation, (ii) pinpoint the half $\mathbf{k}$'s representing advanced knowledge, (iii) check the 50% rule, (iv) show that the quantum algorithm can be represented as the sum of the histories of the related advanced information classical algorithm (each history, given a half $\mathbf{k}$, performs the function evaluations—if any—still required to identify the solution), and also (v) try to rebuild the quantum algorithm out of the advanced information classical algorithm using no a priori knowledge of the quantum algorithm. This part of the work contains a summary of [5]. However, in [5] we gave the 50% rule mostly as a pattern common to the main quantum algorithms. Now, algorithm by algorithm, we gear the 50% rule with the above explanation of the quantum speed up; moreover we further develop the methodology for building the quantum algorithm out of the advanced information classical algorithm.

3 Deutsch's Algorithm

The set of functions is $f_{\mathbf{k}}:\{0,1\}\rightarrow\{0,1\}$—table (4).

x	$f_{00}(x)$	$f_{01}(x)$	$f_{10}(x)$	$f_{11}(x)$
0	0	0	1	1
1	0	1	0	1

(4)

$\mathbf{k}\equiv k_0, k_1$ is the table of the function (the sequence of function values ordered for increasing values of the argument) and, clockwise rotated, the suffix of the function. The oracle chooses at random one of these functions and gives to the second player the black box that, given x, computes $f_{\mathbf{k}}(x)$. The problem is finding whether the function is "balanced" ($\mathbf{k}=01, 10$) or constant. This requires two function evaluations in the classical case, just one in the quantum case [6].

We ideally add to the usual registers a two qubit register K containing the oracle's choice $\mathbf{k}$. Now the black box, given $\mathbf{k}$ and x, computes $f(\mathbf{k},x)=f_{\mathbf{k}}(x)$.

The extended algorithm is: (0) prepare K in the superposition of all $\mathbf{k}$, X in the superposition of all x, and V in the antisymmetric state, (1) perform function evaluation, and (2) apply Hadamard to register X:

$$\Psi_0 = \frac{1}{4}(|00\rangle_K + |01\rangle_K + |10\rangle_K + |11\rangle_K)(|0\rangle_X + |1\rangle_X)(|0\rangle_V - |1\rangle_V), \tag{5}$$

$$\Psi_1 = \frac{1}{4}[(|00\rangle_K - |11\rangle_K)(|0\rangle_X + |1\rangle_X) + (|01\rangle_K - |10\rangle_K)(|0\rangle_X - |1\rangle_X)](|0\rangle_V - |1\rangle_V), \tag{6}$$

$$\Psi_2 = \frac{1}{2\sqrt{2}}[(|00\rangle_K - |11\rangle_K)|0\rangle_X + (|01\rangle_K - |10\rangle_K)|1\rangle_X](|0\rangle_V - |1\rangle_V). \tag{7}$$

Measuring $[K]$ and $[X]$ in (7) determines the moves of both players: the oracle's choice in register K and the solution found by the second player in register X: 0 if the function is constant, 1 if balanced. Backdating to before running the algorithm the reduction induced by measuring $[K]$ gives the original Deutsch's algorithm.

We check the 50% rule. The information acquired by measuring $[K]$ in (7) is the two bits read in K ($[X]$ is redundant). There is advanced knowledge of any one of these two bits. The quantum algorithm requires the number of function evaluations of a classical algorithm that knows in advance either $k_0 = f(\mathbf{k}, 0)$ or $k_1 = f(\mathbf{k}, 1)$. To identify the character of the function, this algorithm must acquire the other bit of information by computing, respectively, either $k_1 = f(\mathbf{k}, 1)$ or $k_0 = f(\mathbf{k}, 0)$. Thus the advanced information classical algorithm has to perform one function evaluation like the quantum algorithm, which verifies the 50% rule.

This rule shows that Deutsch's problem is solvable with a quantum speed up independently of our knowledge of the quantum algorithm. In fact the speed up comes from comparing two classical algorithms, with and without the advanced information. The same applies to all quantum algorithms.

We represent the quantum algorithm as the sum of the histories of the advanced information classical algorithm. Table (8) gives the combinations of advanced information and result of computing the missing information.

#	Advanced information	Result of function evaluation
1	$k_0 = 0$	$k_1 = f(\mathbf{k}, 1) = 0$
2	$k_0 = 0$	$k_1 = f(\mathbf{k}, 1) = 1$
3	$k_0 = 1$	$k_1 = f(\mathbf{k}, 1) = 0$
4	$k_0 = 1$	$k_1 = f(\mathbf{k}, 1) = 1$
5	$k_1 = 0$	$k_0 = f(\mathbf{k}, 0) = 0$
6	$k_1 = 0$	$k_0 = f(\mathbf{k}, 0) = 1$
7	$k_1 = 1$	$k_0 = f(\mathbf{k}, 0) = 0$
8	$k_1 = 1$	$k_0 = f(\mathbf{k}, 0) = 1$

(8)

Each classical computation history is represented in quantum notation as a sequence of sharp states:

- Row #1. Advanced information $k_0 = 0$. The classical algorithm should compute $k_1 = f(\mathbf{k}, 1)$ that, for this row, is $k_1 = 0$. The quantum representation of the oracle's choice is thus $|00\rangle_K$. To compute $f(\mathbf{k}, 1)$, the initial state of the input register X must be $|1\rangle_X$. Since the result is mod 2 added to the initial content of register V, we should split row #1 into: #1.1 with V initially in $|0\rangle_V$ and #1.2 with V initially in $|1\rangle_V$. The initial state

of #1.1 is $\Psi_0^{(1.1)} = |00\rangle_K |1\rangle_X |0\rangle_V$, that of #1.2 is $\Psi_0^{(1.2)} = -|00\rangle_K |1\rangle_X |1\rangle_V$. Computation histories have to be added and must be given an initial phase. We should set the initial phases in such a way that, in the superposition of all histories, we obtain the initial state of the quantum algorithm. This is what is needed to show that the quantum algorithm can be represented as a sum of the histories of the advanced information classical algorithm. The sum of the initial states of #1.1 and #1.2 is: $\Psi_0^{(1)} = \Psi_0^{(1.1)} + \Psi_0^{(1.2)} = |00\rangle_K |1\rangle_X (|0\rangle_V - |1\rangle_V)$. We normalize at the end. Function evaluation transforms $\Psi_0^{(1)}$ into itself: $\Psi_1^{(1)} = \Psi_0^{(1)}$(mod 2 adding $f(00, 1) = 0$ to the former content of V leaves this content unaltered).
- Row #5. Advanced information $k_1 = 0$, result of function evaluation $k_0 = 0$. The same rationale yields: $\Psi_0^{(5)} = |00\rangle_K |0\rangle_X (|0\rangle_V - |1\rangle_V)$, $\Psi_1^{(5)} = \Psi_0^{(5)}$.
- The sum of #1 and #5 yields the transformation of $|00\rangle_K (|0\rangle_X + |1\rangle_X)(|0\rangle_V - |1\rangle_V)$ into itself, i.e. the function evaluation stage of Deutsch's algorithm when K is in $|00\rangle_K$.
- Row #2. $\Psi_0^{(2)} = |01\rangle_K |1\rangle_X (|0\rangle_V - |1\rangle_V)$, $\Psi_1^{(2)} = -\Psi_0^{(2)}$ (mod 2 adding $f(01, 1) = 1$ to the former content of V swaps $|0\rangle_V$ and $|1\rangle_V$ – rotates by π the phase of the pair of histories).
- Row #7. $\Psi_0^{(7)} = |01\rangle_K |0\rangle_X (|0\rangle_V - |1\rangle_V)$, $\Psi_1^{(7)} = \Psi_0^{(7)}$ (mod 2 adding $f(01, 0) = 0$ to the former content of V leaves this content unaltered).
- The sum of #2 and #7 yields the transformation of $|01\rangle_K (|0\rangle_X + |1\rangle_X)(|0\rangle_V - |1\rangle_V)$ into $|01\rangle_K (|0\rangle_X - |1\rangle_X)(|0\rangle_V - |1\rangle_V)$, namely the function evaluation stage of Deutsch's algorithm when K is in $|01\rangle_K$.
- Proceeding in a similar way with the other histories, summing over, and normalizing yields the transformation of Ψ_0 (5) into Ψ_1 (6).

In hindsight, there is a shortcut. For each $|\mathbf{k}\rangle_K$, we perform function evaluation not only for the values of x required to identify the solution, also for the other values. I.e., we perform function evaluation for each product $|\mathbf{k}\rangle_K (|0\rangle_X + |1\rangle_X)(|0\rangle_V - |1\rangle_V)$; junk histories (for that $|\mathbf{k}\rangle_K$) do not harm, the important thing is performing function evaluation for the values of x required to identify the solution. This yields the transformation of Ψ_0 into Ψ_1. Conversely, by simply inspecting the form of Ψ_0 in (5), one can see that each $|\mathbf{k}\rangle_K (|0\rangle_X + |1\rangle_X)(|0\rangle_V - |1\rangle_V)$ is the initial state of a bunch of histories as from the shortcut. Quantum parallel computation is thus seen as the sum of the histories of a classical algorithm that, given the advanced information, computes the missing information required to identify the solution of the problem. The final rotation of the basis of register X serves to make the information about the oracle's choice—propagated to X with function evaluation—readable.

Summing up, Deutsch's algorithm can be represented as a sum of the histories of the related advanced information classical algorithm, with histories phases and final rotation of the X basis that reconstruct the quantum algorithm. In this way mutual causality explains the structure of the quantum algorithm.

Interestingly, we can also build the quantum algorithm out of the advanced information classical algorithm. In fact history initial phases and final rotation of the X basis are definable, independently of our a priori knowledge of the quantum algorithm, as follows.

Let us make a simplification: we still use a priori knowledge of the quantum algorithm to set the amplitudes of the initial superposition in X equal to one another. Instead, we take the generic initial state of V: $\alpha(|0\rangle_V + |1\rangle_V) + \beta(|0\rangle_V - |1\rangle_V)$. The initial amplitude of histories starting with V in $|0\rangle_V$ is thus $\alpha + \beta$, with V in $|1\rangle_V$ is $\alpha - \beta$. Under α, the computation performed by the advanced information classical algorithm gets lost in the quantum translation, since the overall factorizable initial state is transformed into itself. Under β, the transfer of information from classical to quantum algorithm is maximum and the entanglement between registers K and X is maximized (it is also maximal in the present case). We obtain the function evaluation stage of the quantum algorithm. One can readily check that the same holds also if we set the amplitudes in the initial state of register X free.

As for the rotation of the X basis, let us first discuss the form of Ψ_1, the state after function evaluation. With $\alpha = 0$ and $\beta = 1$, function evaluation creates a maximal entanglement, two orthogonal states of K, $|00\rangle_K - |11\rangle_K$ and $|01\rangle_K - |10\rangle_K$ are correlated with two orthogonal states of X. This means that register X contains the information that discriminates between $|00\rangle_K - |11\rangle_K$ and $|01\rangle_K - |10\rangle_K$ (or between $\mathrm{e}^{i\delta_{00}}|00\rangle_K - \mathrm{e}^{i\delta_{11}}|11\rangle_K$ and $\mathrm{e}^{i\delta_{01}}|01\rangle_K - \mathrm{e}^{i\delta_{10}}|10\rangle_K$ if the superposition in K is incoherent), namely between constant and balanced functions. Therefore we should rotate the basis of X in such a way that this information becomes readable: $|0\rangle_X + |1\rangle_X$ should go into $|0\rangle_X$, etc. We can define this rotation (independently of our a priori knowledge of the quantum algorithm) by setting the requirement that the entanglement between K and X (generated by function evaluation) becomes correlation between the outcomes of measuring $[K]$ and $[X]$.

Summing up, in the case of Deutsch's algorithm, the advanced information classical algorithm defines the quantum algorithm provided that history phases maximize the entanglement generated by function evaluation and final rotation of the X basis transforms this entanglement into correlation between measurement outcomes.

The fact that function evaluation generates the desired entanglement between the character of the function and the solution is not a necessity of course. It is Deutsch's problem that is designed around this fact. The problem could be different, for example distinguishing between f_{00} and f_{01} on the one side and f_{10} and f_{11} on the other. The algorithm should be correspondingly changed by left multiplying Hadamard on X by the appropriate permutation of the basis vectors of X. However, we will see that the problems addressed by the quantum algorithms are all designed to exploit the entanglement generated by function evaluation with the slightest use of other operations.

4 Deutsch&Jozsa's algorithm

The set of functions is the constant and balanced functions $f_{\mathbf{k}} : \{0,1\}^n \to \{0,1\}$; $\mathbf{k} \equiv k_0, k_1, \ldots, k_{2^n-1}$ is both table and suffix—table (9) for $n = 2$.

x	$f_{0000}(x)$	$f_{1111}(x)$	$f_{0011}(x)$	$f_{1100}(x)$	$f_{0101}(x)$	$f_{1010}(x)$	$f_{0110}(x)$	$f_{1001}(x)$
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

(9)

The oracle chooses a function. The problem is finding whether the function is balanced or constant through function evaluation—see [7].

The black box, given $\mathbf{k}$ and x, computes $f(\mathbf{k}, x) = f_{\mathbf{k}}(x)$. The 2^n qubit oracle's choice register K (just a conceptual reference) contains $\mathbf{k}$. The algorithm is: (0) prepare K in the superposition of all $\mathbf{k}$, X in the superposition of all x, and V in the antisymmetric state, (1) perform function evaluation, which changes the content of V from v to $v \oplus f(\mathbf{k}, x)$, and (2) apply Hadamard to register X:

$$\Psi_0 = \frac{1}{8}(|0000\rangle_K + |1111\rangle_K + |0011\rangle_K + |1100\rangle_K + \cdots)(|00\rangle_X + |01\rangle_X + |10\rangle_X + |11\rangle_X)(|0\rangle_V - |1\rangle_V). \tag{10}$$

$$\Psi_1 = \frac{1}{8}\left[\begin{array}{l}(|0000\rangle_K - |1111\rangle_K)(|00\rangle_X + |01\rangle_X + |10\rangle_X + |11\rangle_X)\\ \quad + (|0011\rangle_K - |1100\rangle_K)(|00\rangle_X + |01\rangle_X - |10\rangle_X - |11\rangle_X) + \cdots\end{array}\right](|0\rangle_V - |1\rangle_V). \tag{11}$$

$$\Psi_2 = \frac{1}{4}[(|0000\rangle_K - |1111\rangle_K)|00\rangle_X + (|0011\rangle_K - |1100\rangle_K)|10\rangle_X + \cdots](|0\rangle_V - |1\rangle_V). \tag{12}$$

Measuring $[K]$ and $[X]$ in (12) determines the oracle's choice and the solution found by the second player: all zeroes if constant, not so if balanced. Backdating the reduction on **k** gives the original algorithm.

We check the 50% rule. There is advanced knowledge of any half **k**. We distinguish between: (i) half tables that do not contain different values of the function and (ii) those that do. In case (i), the solution is identified by computing an extra row, i.e. by performing one function evaluation for any value of x outside the advanced information (if the value of the function is still the same, the function is constant, otherwise it is balanced). In case (ii), we know already that the function is balanced and no function evaluation is needed. Although some half **k** require one function evaluation and some none, the 50% rule is satisfied. In fact the half **k** that already specify the solution, thus require no function evaluation, contain 100% of the information about the solution and should not be taken into account.

The function evaluation stage of the quantum algorithm (the transformation of Ψ_0 into Ψ_1) is the sum of the histories of the advanced information classical algorithm as from the "shortcut" highlighted for Deutsch's algorithm.

We build the quantum algorithm out of the advanced information classical algorithm. As for the choice of the history initial phases, this is justified as in Deutsch's algorithm. As for the rotation of the X basis, we examine the outcome of function evaluation, namely Ψ_1 (11). There is maximal entanglement between K and X. Orthogonal states of K, discriminating between constant and balanced functions, are correlated with orthogonal states of X. The information whether the function is constant or balanced has propagated to register X. To read this information, we rotate the X basis in such a way that $(|0000\rangle_K - |1111\rangle_K)(|00\rangle_X + |01\rangle_X + |10\rangle_X + |11\rangle_X)$ goes into $(|0000\rangle_K - |1111\rangle_K)|00\rangle_X$, etc. This rotation of the basis of X is such that the entanglement between K and X becomes correlation between the outcomes of measuring $[K]$ and $[X]$. This is a constructive definition of Hadamard on X. This completes the derivation of the quantum algorithm from the advanced information classical algorithm.

5 Bernstein&Vazirani's Algorithm

The analysis of the previous section also applies to Bernstein&Vazirani's algorithm, see [1], which is a restriction of Deutsch&Jozsa's algorithm. The set of the constant and balanced functions is restricted to a proper subset thereof, namely to the set of the functions such that $f_{\mathbf{k}}(x) = a \cdot x$, with $a \cdot x = (\sum_{i\in\{0,1\}^n} a_i x_i) \bmod 2$. The problem is to find the "hidden string" a. The algorithm is Deutsch&Jozsa's algorithm with the superposition hosted in register K reduced to the new set of functions. Measuring $[K]$ and $[X]$ at the end of the algorithm yields a value of **k** and the corresponding value of a. The discussion is the same as in the former section.

6 Simon's Algorithm

The set of functions is $f_{\mathbf{k}}:\{0,1\}^n\to\{0,1\}^{n-1}$, such that $f_{\mathbf{k}}(x)=f_{\mathbf{k}}(y)$ iff $x=y$ or $x=y\oplus\mathbf{h}^{(\mathbf{k})}$; $\mathbf{h}^{(\mathbf{k})}\equiv h_0^{(\mathbf{k})},h_1^{(\mathbf{k})},\ldots,h_{n-1}^{(\mathbf{k})}$ (depending on $\mathbf{k}$) belongs to $\{0,1\}^n$ excluding the all zeroes string; $\oplus$ denotes bitwise mod 2 addition. Table (13) gives the set for $n=2$; $\mathbf{k}$ is both table and suffix. Since $\mathbf{h}^{(\mathbf{k})}\oplus\mathbf{h}^{(\mathbf{k})}=0$, each value of the function appears exactly twice in the table, thus 50% of the rows plus one surely identify $\mathbf{h}^{(\mathbf{k})}$.

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

(13)

The oracle chooses a function. The problem is finding the value of $\mathbf{h}^{(\mathbf{k})}$, "hidden" in the $f_{\mathbf{k}}(x)$ chosen by the oracle, through function evaluation, see [12]. The black box, given $\mathbf{k}$ and x, computes $f(\mathbf{k},x)=f_{\mathbf{k}}(x)$. The oracle's choice register K is $2^n(n-1)$ qubit. The algorithm is: (0) prepare K in the superposition of all $\mathbf{k}$, X in the superposition of all x, and V in the all zeroes string $|\mathbf{0}\rangle_V$, (1) perform function evaluation, which changes the content of V from $\mathbf{v}$ to $\mathbf{v}\oplus f(\mathbf{k},x)$, where $\oplus$ denotes bitwise mod 2 addition, and (2) apply Hadamard to X:

$$\Psi_0=\frac{1}{2\sqrt{6}}(|0011\rangle_K+|1100\rangle_K+|0101\rangle_K+|1010\rangle_K+\cdots)(|00\rangle_X+|01\rangle_X+|10\rangle_X+|11\rangle_X)|0\rangle_V. \tag{14}$$

$$\Psi_1=\frac{1}{2\sqrt{6}}\begin{bmatrix}(|0011\rangle_K+|1100\rangle_K)[(|00\rangle_X+|01\rangle_X)|0\rangle_V+(|10\rangle_X+|11\rangle_X)|1\rangle_V]\\ +(|0101\rangle_K+|1010\rangle_K)[(|00\rangle_X+|10\rangle_X)|0\rangle_V+(|01\rangle_X+|11\rangle_X)|1\rangle_V]+\cdots\end{bmatrix}. \tag{15}$$

$$\Psi_2=\frac{1}{2\sqrt{6}}\begin{bmatrix}(|0011\rangle_K+|1100\rangle_K)[(|00\rangle_X+|10\rangle_X)|0\rangle_V+(|00\rangle_X-|10\rangle_X)|1\rangle_V]\\ +(|0101\rangle_K+|1010\rangle_K)[(|00\rangle_X+|01\rangle_X)|0\rangle_V+(|00\rangle_X-|01\rangle_X)|1\rangle_V]+\cdots\end{bmatrix}. \tag{16}$$

In Ψ_2, for each value of the content of K and no matter the content of V, register X hosts even weighted superpositions of the 2^{n-1} strings $\mathbf{s}_j^{(\mathbf{k})}$ orthogonal to $\mathbf{h}^{(\mathbf{k})}$. By measuring $[K]$ and $[X]$ we obtain at random the oracle's choice $\mathbf{k}$ and one of the $\mathbf{s}_j^{(\mathbf{k})}$.

We leave K in its after-measurement state, thus fixing $\mathbf{k}$, and iterate the right part of the algorithm (preparation of registers X and V, function evaluation, and measurement of $[X]$) until obtaining $n-1$ different $\mathbf{s}_j^{(\mathbf{k})}$, which allows to find $\mathbf{h}^{(\mathbf{k})}$ by solving the system of $n-1$ mod 2 linear equations.

We check the 50% rule. We focus on the quantum part of Simon's problem, namely on the problem of generating at random a string $\mathbf{s}_j^{(\mathbf{k})}$ orthogonal to $\mathbf{h}^{(\mathbf{k})}$. Any $\mathbf{s}_j^{(\mathbf{k})}$ is a "solution" of this problem. This formulation of Simon's problem and the usual formulation (finding $\mathbf{h}^{(\mathbf{k})}$) are equivalent as far as an exponential speed up in the former implies an exponential speed up in the latter and vice-versa. The information acquired by measuring $[K]$ and $[X]$ in (16) is the information in K (the content of X—the string $\mathbf{s}_j^{(\mathbf{k})}$—is a function of the content of K). Measuring $[K]$ induces state reduction on $\mathbf{k}$. There is advanced knowledge of any half $\mathbf{k}$. If the half table does not contain a same value of the function twice, the solution is identified by performing one function evaluation for any value of x outside the advanced information. The new value of the function is necessarily a value already present in the advanced information, which identifies $\mathbf{h}^{(\mathbf{k})}$ thus all the $\mathbf{s}_j^{(\mathbf{k})}$. If the half table contains a same value of the function twice, this already identifies the solution and no function evaluation is needed. This verifies the 50% rule.

One can see that the same discussion applies to the Generalized Simon's algorithm, thus to the Hidden Subgroup Algorithms, like finding the period of a function (the quantum part of Shor's factorization algorithm), etc.

The function evaluation stage of the quantum algorithm is the sum of the histories of the advanced information classical algorithm—as from the shortcut highlighted for Deutsch's algorithm.

As for building the quantum algorithm out of the advanced information classical algorithm, things are more difficult. In fact Simon's algorithm is not optimal under the criteria of maximizing entanglement. We show this for $n = 2$. We still choose even amplitudes for the initial superposition in X, while letting V in the generic initial state $\alpha|0\rangle_V + \beta|1\rangle_V$. As readily checked, $\alpha = -\beta$ maximizes the entanglement between K and X after function evaluation. Hadamard on X transforms this entanglement into correlation between the outcomes of measuring $[K]$ and $[X]$. With V prepared in the antisymmetric state, state (16) becomes:

$$\frac{1}{2\sqrt{3}}[(|0011\rangle_K + |1100\rangle_K)|10\rangle_X + (|0101\rangle_K + |1010\rangle_K)|01\rangle_X + \cdots](|0\rangle_V - |1\rangle_V).$$

Swapping the basis vectors $|01\rangle_X$ and $|10\rangle_X$ yields the desired correlation between $\mathbf{k}$ and $\mathbf{h}^{(\mathbf{k})}$. In terms of number of oracle's queries, the optimal algorithm is more efficient than the known algorithm. In fact it yields the hidden string $\mathbf{h}^{(\mathbf{k})}$ (rather than a $\mathbf{s}_j^{(\mathbf{k})}$) with a single oracle's query, as expected since just one function evaluation identifies $\mathbf{h}^{(\mathbf{k})}$. The same holds for $n > 2$. We prepare each qubit of V in the antisymmetric state. The final rotation of the X basis becomes Hadamard followed by a permutation of the basis vectors—in general depending on $\mathbf{k}$. However, identifying and implementing this permutation might be costly. Simon's algorithm is a suboptimal solution in terms of number of oracle's queries, however the poly(n) cost of solving the system of $n-1$ linear equations does not frustrate the exponential speed up. This might not be the case for the "optimal" algorithm.

7 Grover's Algorithm

See Sect. 2 for $n = 2$. We check the 50% rule for $n = 2$. Mutual causality is between the two bits of the outcome of measuring $[K]$. The quantum algorithm requires the number of function evaluations of a classical algorithm that knows in advance either k_0 or k_1. To identify the missing bit, this algorithm should perform one function evaluation, as readily checked. This verifies the 50% rule. As for $n > 2$, a classical algorithm that knows in advance 50% of the n bits of the data base location, to identify the $n/2$ missing bits should perform $O(2^{n/2})$ function evaluations, against the $O(2^n)$ of a classical algorithm without advanced information. This verifies the 50% rule for $n > 2$.

As for representing the quantum algorithm as the sum of the histories of the advanced information classical algorithm, still for $n = 2$, the procedure is the same as in the former algorithms.

We build the quantum algorithm out of the advanced information classical algorithm—still for $n = 2$ to start with. As for the history initial phases, their choice is justified as in

Deutsch's algorithm. As for the final rotation of the X basis, we examine the outcome of function evaluation, i.e. Ψ_1 (2). Registers K and X are maximally entangled, orthogonal states of K, each corresponding to a value of **k**, are correlated with orthogonal states of X, which means that the value of **k** has propagated to register X. To read this value, i.e. to transform entanglement between K and X into correlation between the outputs of measuring $[K]$ and $[X]$, we should rotate the X basis in such a way that $-|00\rangle_X + |01\rangle_X + |10\rangle_X + |11\rangle_X$ (correlated with $|00\rangle_K$) goes into $|00\rangle_X$, etc. This is a constructive definition of the transformation U.

Generalizing to $n > 2$ is straightforward. Given the advanced knowledge of $n/2$ bits, in order to compute the missing $n/2$ bits we should perform function evaluation and rotation of the basis of X an $O(2^{\frac{n}{2}})$ times. Each time we obtain the superposition of an unentangled state of the form (1) and a maximally entangled state of the form (3). At each successive iteration, the amplitude of the latter state is amplified at the expense of the amplitude of the former, until it becomes about 1 in $O(2^{\frac{n}{2}})$ iterations. The final measurement of $[K]$ induces state reduction on **k**. There is advanced knowledge of any half **k**.

8 Engineering Quantum Algorithms

The 50% rule and the sum of the histories make up a tool for searching quantum speed ups. The rule, by comparing two classical algorithms (with and without advanced information), allows to identify the problems solvable with a quantum speed up in terms of number of oracle's queries. Once the problem is identified, the sum of the histories of the advanced information classical algorithm—with histories degrees of freedom set to maximize entanglement/correlation—should yield the speed up in terms of number of queries. This can also be an overall speed up, like in Deutsch's, Deutsch&Jozsa's, Bernstein&Vazirani's, and Grover's algorithms. However, from a practical standpoint, the speed up in terms of number of queries could be frustrated by the cost of setting the histories degrees of freedom. One might have to compromise between the two things, like in Simon's algorithm.

We exemplify this methodology by building from scratch a quantum algorithm (a variation of Deutsch's algorithm). The oracle chooses a function out of the set $f_{\mathbf{k}}$: $\{0, 1\} \to \{00, 01, 10\}$—table (17).

x	f_{0000}	f_{0001}	f_{0100}	f_{0101}	f_{0010}	f_{1000}	f_{1010}	f_{0110}	f_{1010}
0	00	00	01	01	00	10	10	01	10
1	00	01	00	01	10	00	10	10	10

(17)

The problem is whether the mod 2 addition of the four bits of **k** is 0 or 1. Given any half table, the advanced information classical algorithm, in order to find the solution, should perform function evaluation for the other value of x. It is easy to derive the quantum algorithm out of this classical algorithm. We need a 4 qubits oracle's choice register K, a 1 qubit oracle's query register X, and a 2 qubits register V for the result of function evaluation (bitwise mod 2 added to the former content of V for logical reversibility). Setting histories degrees of freedom to maximize entanglement/correlation yields the algorithm: (0) prepare K in the superposition of all **k**, X in the superposition of all x, each qubit of V in the antisymmetric state, (1) perform function evaluation, and (2) apply Hadamard on X. The final measurement of $[K]$ and $[X]$ yields the function chosen by the oracle in K and the solution in X.

Searching speed ups in terms of number of oracle's queries with the 50% rule is straightforward. We outline another example. Let the set of functions be $f_{\mathbf{k}}: \{0,1\}^2 \rightarrow \{0,1\}^2$ such that the sequence of function values for increasing values of the argument is a permutation of the values of the argument (there are 4! such functions). Given a black box that computes the function, the problem is to find some character of the permutation. A classical algorithm with advanced information, to identify the function requires one function evaluation, without advanced information it requires three function evaluations. Thus there is room for a speed up in terms of number of oracle's queries. Going to the quantum algorithm, register K is 8 qubits (the sequence of four fields of two qubits, each containing a value of the function), registers X is two qubits, enough to specify a four valued character of the function, and register V is 2 qubits. For example, we can prepare the two qubits V_0 and V_1 of register V in the antisymmetric state, so that function character, after function evaluation, is encoded in the phases of the superposition hosted in register X. The preparation is thus

$$\frac{1}{8\sqrt{6}}(|00\ 01\ 10\ 11\rangle_K + |00\ 01\ 11\ 10\rangle_K \ldots)(|00\rangle_X + |01\rangle_X + |10\rangle_X + |11\rangle_X)(|0\rangle_{V_0} - |1\rangle_{V_0}) \\ \times (|0\rangle_{V_1} - |1\rangle_{V_1}).$$

We can hope that function evaluation yields four orthogonal vectors of the Hilbert space of register K correlated with four orthogonal vectors of X. If this is the case, we should investigate which character of the function is discriminated by the four orthogonal vectors of X, and design around it the problem addressed by the quantum algorithm. Then we should identify the final rotation of the X basis that allows to read such a character. Clearly, all this is tentative, and one should play with things. For example, with a three valued character of the permutation, namely dividing the 24 functions in 3 partitions of 8, just one function evaluation (against three classically) followed by Hadamard on X and measurement of $[X]$, tells which partition the function belongs to.

With the 50% rule, one can figure out any number of these "potential speed up" situations. This should give a playground both for searching new quantum speed ups and for developing quantum algorithm engineering.

9 Conclusions

Summing up, the points in favour of the 50% rule are: (i) the rule is self evident, it is an immediate consequence of postulating the symmetry of mutual causality (like between two polarizations measured in an entangled polarization state) in a more complete representation of the quantum algorithms, (ii) the rule is verified by the main quantum algorithms, and (iii) the rule explains the structure of these algorithms—which are in fact representable as a sum of the histories of the respective advanced information classical algorithm.

The 50% rule and the sum of the histories picture make up a tool for the search of quantum speed ups, as discussed in the previous section. This should be interesting in a situation where finding a speed up was, so to speak, a matter of art and no further speed ups of practical interest have been discovered since 1996. Using this tool for a systematic search of the speed ups should also foster the quantum algorithms engineering outlined in this paper.

Because of its peculiar character, the present explanation of the speed up could have an epistemological and interdisciplinary interest. It questions the mechanistic vision of evolutions where causality propagates locally through an instantaneous present (the dt of

Schrödinger equation). Quantum algorithms are partly driven by their future outcome and, in some sense, "exist" (host mutual causality) in an extended present comprising preparation, unitary evolution, and measurement. Under the perspective of the 50% rule, quantum computation turns out to be the first formalized example of teleological evolution. This might shed light on quantum and teleological explanations of organic behavior.

Acknowledgements The author thanks for useful comments/suggestions his wife Ferdinanda Pavia and: Scott Aaronson, Pablo Arrighi, Vint Cerf, Artur Ekert, David Finkelstein, Shlomit Finkelstein, Hartmut Neven, and Charles Stromeyer.

References

1. Bernstein, E., Vazirani, U.: In: Proceedings of the 25th Annual ACM Symposium on the Theory of Computing, p. 11, 1993
2. Castagnoli, G., Finkelstein, D.: Theory of the quantum speed up. Proc. R. Soc. Lond. A **457**, 1799 (2001). arXiv:quant-ph/0010081v1
3. Castagnoli, G.: The mechanism of quantum computation. Int. J. Theor. Phys. **47**(8), 2181 (2008)
4. Castagnoli, G.: The quantum speed up as advanced cognition of the solution. Int. J. Theor. Phys. **48**(3), 857 (2009)
5. Castagnoli, G.: The 50% advanced information rule of the quantum algorithms. Int. J. Theor. Phys. (2009, to be published)
6. Deutsch, D.: Quantum theory, the Church-Turing principle, and the universal quantum computer. Proc. R. Soc. (Lond.) A **400**, 97 (1985)
7. Deutsch, D., Jozsa, R.: Rapid solution of problems by quantum computation. Proc. R. Soc. (Lond.) A **439**, 553 (1992)
8. Dolev, S., Elitzur, A.C.: Non-Sequential Behavior of the Wave Function. (2005). arXiv:quant-ph/0102109v1
9. Gross, D., Flammia, S.T., Eisert, X.: J. Phys. Rev. Lett. **102**(19) (2009)
10. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proc. 28th Ann. ACM Symp. Theory of Computing, 1996
11. Grover, L.K.: From Schrödinger equation to quantum search algorithm. (2001). arXiv:quant-ph/0109116
12. Simon, D.: On the power of quantum computation. In: Proc. 35th Ann. Symp. on Foundations of Comp. Sci., p. 116, 1994