Quantum Information Processing (2021) 20:214
https://doi.org/10.1007/s11128-021-03156-3

On a poset of quantum exact promise problems

Elías F. Combarro[1,2] · Sofia Vallecorsa[2] · Alberto Di Meglio[2] · Alejandro Piñera[3] · Ignacio Fernández Rúa[3]

Received: 6 January 2021 / Accepted: 14 June 2021

Abstract

Two of the most well-known quantum algorithms, those introduced by Deutsch–Jozsa and Bernstein–Vazirani, can solve promise problems with just one function query, showing an oracular separation with deterministic classical algorithms. In this work, we generalise those methods to study a family of quantum algorithms that can, with just one query, exactly solve promise problems stated over Boolean functions. We also show that these problems can be naturally ordered, inducing a partially ordered set of promise problems. We study the properties of such a poset, showing that the Deutsch–Jozsa and Bernstein–Vazirani problems are, in a certain sense, extremal problems in it, determining some of its automorphisms and proving that it is connected. We also prove that, for the problems in the poset, the corresponding classical query complexities can take any value between 1 and $2^{n-1} + 1$.

Keywords Quantum algorithms · Posets · Query complexity · Deutsch–Jozsa algorithm · Bernstein–Vazirani algorithm

✉ Elías F. Combarro
efernandezca@uniovi.es

Sofia Vallecorsa
Sofia.Vallecorsa@cern.ch

Alberto Di Meglio
Alberto.Di.Meglio@cern.ch

Alejandro Piñera
apnicolas@uniovi.es

Ignacio Fernández Rúa
rua@uniovi.es

[1] Computer Science Department, University of Oviedo, Oviedo, Spain

[2] CERN openlab, CERN, Geneva, Switzerland

[3] Mathematics Department, University of Oviedo, Oviedo, Spain

Published online: 23 June 2021

1 Introduction

Since the dawn of quantum computing [7,9,14], special attention has been paid to promise problems defined over Boolean functions. In fact, Deutsch's algorithm [7], which is arguably the first quantum algorithm, its generalisation in Deutsch–Jozsa algorithm [8] and the famous Bernstein–Vazirani [1] algorithm are all of them methods of this kind. The interest of these algorithms lies in the fact that they can exactly solve promise problems with just one function query (through the use of a quantum oracle that can be queried in superposition), whereas classical deterministic algorithms require a number of queries that grows (even exponentially, as in the case of the Deutsch–Jozsa problem) with the size of the function input in order to complete the same task.

Although the importance of these problems and algorithms is mainly theoretical, they prepared the ground, together with related methods such as the one used to solve Simon's problem [18], for later quantum computing breakthroughs, including the celebrated Grover algorithm [11] and the famous Shor's methods for factoring and for finding discrete logarithms [17].

Studying these and other quantum algorithms under a common framework is also interesting because it can lead to the discovery of new algorithms and to a better understanding of their properties. For instance, both Shor's and Simon's algorithms are particular cases of quantum solutions to the Hidden Subgroup Problem [13] and it has also been shown (see [6]) that both Grover's algorithm and a number of quantum walks [16,19,20] can be seen as instances of a family of methods called quantum abstract detecting systems, something that can help achieve better detection rates in some practical problems [4,5].

In this work, we introduce an extension of the concept of promise problem over Boolean functions and study in which cases they can be solved in the quantum setting with just one query by using the same scheme as in Deutsch–Jozsa and Bernstein–Vazirani algorithms. We show that these problems naturally form a partially ordered set and we study some of its properties, such as its maximal and minimal elements. We also prove that in the deterministic classical case, these problems may require any number of queries from 1 to $2^{n-1}+1$.

The rest of the paper is organised as follows. In Sect. 2, we define the type of promise problems that we will be studying in this work and show that the problems solved by Deutsch–Jozsa and Bernstein–Vazirani algorithms are particular cases of them. We also prove some useful results about these problems. In Sect. 3, we show that, by defining a natural binary relationship, the set of these problems can be partially ordered and we study some of the properties of the resulting poset, including its minimal and maximal elements, some of its automorphisms and showing that it is connected. Section 4 states and proves one of the main theorems of this work: That the range of complexities of deterministic classical algorithms for solving the promise problems considered in this paper is as diverse as possible. After that, in Sect. 5, we present numerical results of the execution on actual quantum computers of some the problems introduced in this paper. Finally, in Sect. 6, we raise some conclusions and present some ideas for future research.

2 Quantum exact promise problems

The problems that are solved with the Deutsch–Jozsa and Bernstein–Vazirani algorithms are promise problems defined over Boolean functions. In the first of these algorithms, we are given a Boolean function f which is either constant or balanced (that is, f takes value 0 on exactly half of inputs and value 1 on the other half) and we need to determine which of the two cases we are in; in the second, we are given a linear Boolean function f and we are asked to find the Boolean string s such that $f(x) = x \cdot s \bmod 2$.

In this paper, we study a generalisation of this kind of problems that is given in the following definition.

Definition 1 A promise problem over Boolean functions of n variables is given by a collection of $m \geq 2$ non-empty and pairwise disjoint subsets of functions $f : \{0,1\}^n \to \{0,1\}$. That is, a collection $\{F_i\}_{i=1}^m$ with $m \geq 2$ such that

- $F_i \neq \emptyset$ for $i = 1, \dots, m$
- $F_i \cap F_j = \emptyset$ if $i \neq j$
- $F_i \subseteq 2^{\{0,1\}^n}$ for $i = 1, \dots, m$

The problem we are asked to solve is, given (a black box for computing) $f \in F = \bigcup_{i=1}^m F_i$, determine the unique i such that $f \in F_i$.

When the size n is clear from the context, we will call $\{F_i\}_{i=1}^m$ simply a promise problem.

Notice that, for a fixed n, both Deutsch–Jozsa and Bernstein–Vazirani solve problems as those considered in Definition 1. In the case of Deutsch–Jozsa, we have $m = 2$, $F_1 = \{f : f \in 2^{\{0,1\}^n} \text{and } f \text{ is constant}\}$ and $F_2 = \{f : f \in 2^{\{0,1\}^n} \text{and } f \text{ is balanced}\}$. In the case of Bernstein–Vazirani, $m = 2^n$ and $F_i = \{f_{s_i}\}$, where $f_{s_i}(x) = x \cdot s_i \bmod 2$ and s_i is the string of length n that gives the binary expansion of i.

Of the possible promise problems over Boolean functions, we are especially interested in those that can be solved exactly with quantum algorithms that have access to a quantum oracle implementing the Boolean function f whose set F_i we need to determine. This is formalised in the following definition.

Definition 2 Consider $P = \{F_i\}_{i=1}^m$, a promise problem over Boolean functions of n variables and a quantum algorithm A that has access to an oracle O_f for a Boolean function f of n variables and gives a string $r \in \{0,1\}^n$ as output. We say that A is exact for P if there exists a partition $\{B_i\}_{i=1}^m$ of $\{0,1\}^n$ such that when f is taken from $\bigcup_{i=1}^m F_i$ we have

$$Pr(A \text{ outputs a string in } B_i | f \in F_i) = 1$$

That is, if O_f is a quantum oracle for a function f in F_i, then A will always return a string from B_i.

Fig. 1 Quantum circuit for the Deutsch–Jozsa and Bernstein–Vazirani algorithms

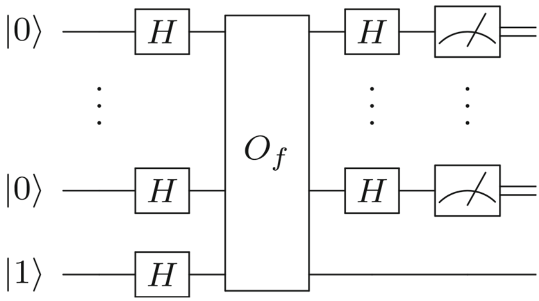
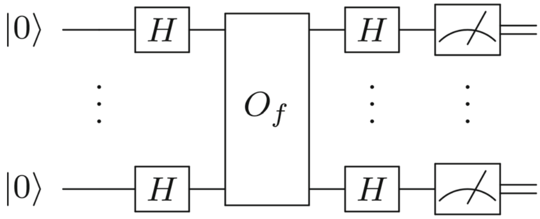


Fig. 2 Quantum circuit with phase oracle



Remark 1 The partition $\{B_i\}_{i=1}^m$ in Definition 2 is not necessarily unique. For instance, there may exist strings which are never output by the algorithm, so they can belong to any of the B_i's with no effect on its exactness.

If we have a quantum algorithm A that is exact for a promise problem P, then we can, obviously, solve P by using A once. Given f, we just need to run A with oracle O_f and determine which B_i the output belongs to. This is the case of the Deutsch–Jozsa and Bernstein–Vazirani algorithms, which use the circuit given in Fig. 1 to solve their promise problems. In the case of Deutsch–Jozsa, the partition is given by $B_1 = \{0^n\}$ and $B_2 = \{0, 1\}^n \backslash \{0^n\}$. If the algorithm returns 0^n, we know that f is constant; otherwise, the function is balanced. For the Bernstein–Vazirani algorithm, the partition is clearly $\{B_i\}_{i=0}^{2^n-1}$ with $B_i = \{s_i\}$ (and, s_i, as above, the binary expansion of i). In fact, both algorithms use just one query to the oracle O_f, which leads us to our main definition.

Definition 3 A promise problem P over Boolean functions of n variables is said to be 1-quantum exact (or, simply, 1-qe) if the algorithm that uses one query to oracle following the circuit of Fig. 1 is exact for P.

It is a well-known fact that the ancillary lowermost qubit in the circuit of Fig. 1 is only used to facilitate a phase kickback when the Boolean function f implemented by the oracle returns 1. Thus, such an ancillary qubit can be removed if we replace the usual Boolean oracle with a phase oracle (see, for instance, [10]). That is, instead of considering O_f a unitary such that $O_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$ we define O_f by $O_f|x\rangle = (-1)^{f(x)}|x\rangle$ (which is clearly unitary). Then, the circuit with phase oracle is the one depicted in Fig. 2.

We will consider all our oracles to be phase oracles from now on, something that will greatly simplify our definitions and proofs. We redefine, accordingly, 1-qe algorithms in terms of the quantum circuit of Fig. 2. We will also identify Boolean functions with their phase oracles so, for instance, the constant function $f = 0$ will be identified with the identity I and the constant function $f = 1$ will be identified with $-I$.

Our goal in the rest of this section is to prove some necessary and sufficient conditions that promise problems need to verify in order to be 1-qe. We start by fixing some notation and by stating and proving a useful lemma about the representation of phase oracles.

Definition 4 For a fixed non-negative integer n,s we denote by Z_i, with $1 \leq i \leq n$, the tensor product of matrices

$$\overbrace{I \otimes \ldots \otimes I}^{i-1} \otimes Z \otimes \overbrace{I \otimes \ldots \otimes I}^{n-i}$$

where I is 2×2 identity matrix and Z is the Pauli matrix $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$. For a binary string t of length n, we denote

$$Z^t = \prod_{i=1}^{n} Z_i^{t_i}$$

where t_i is the ith bit in t.

The following, very useful lemma follows from the results in [3]. For the sake of completeness, we provide here a different, self-contained proof.

Lemma 1 *If O_f is the phase oracle of a Boolean function f of n variables, then*

$$O_f = \sum_{t \in \{0,1\}^n} a_t Z^t$$

with $a_t = \frac{1}{2^n} Tr(Z^t O_f)$.

In particular, if f is a linear form given by $f(x) = x \cdot s \bmod 2$ for some $s \in \{0, 1\}^n$, then $O_f = Z^s$.

Proof It is clear that O_f can be expressed as

$$O_f = I - 2 \sum_{f(x)=1} |x\rangle\langle x|$$

We also know (see [15], for instance) that any $2^n \times 2^n$ matrix can be written as a linear combination of tensor products of n Pauli matrices. Thus, $O_f = \sum b_P P$ where the matrices P are tensor products of Paulis. If Q is also a tensor product of Paulis, we have

$$Tr(Q \cdot O_f) = \sum b_P Tr(Q \cdot P) = b_Q Tr(Q^2) = b_Q Tr(I) = 2^n b_Q$$

because the product of two different tensor products of Paulis is traceless and the square of a tensor product of Paulis is the identity.

Now, suppose P contains at least one X or one Y. Then

$$b_P = 2^{-n}Tr(P \cdot O_f) = Tr(P) - 2 \sum_{f(x)=1} Tr(P|x\rangle\langle x|) = 0 - 2 \sum_{f(x)=1} \langle x|P|x\rangle = 0$$

since $\langle 0|X|0\rangle = \langle 0|Y|0\rangle = \langle 1|X|1\rangle = \langle 1|Y|1\rangle = 0$.

Thus, it holds that if $b_P \neq 0$ then P is of the form Z^t for some binary string t.

Finally, if $f(x) = x \cdot s \bmod 2$, since for every $x \in \{0,1\}^n$ we have $O_f|x\rangle = (-1)^{f(x)}|x\rangle = (-1)^{x \cdot s \mod 2}|x\rangle = Z^s|x\rangle$, we get $O_f = Z^s$. □

Now, we prove a couple of lemmas that relate the expression for phase oracles that we have found in Lemma 1 with the probabilities of measuring certain outcomes with the circuit of Fig. 2.

Lemma 2 *If we use the phase oracle O_f of a Boolean function of n variables in the circuit of Fig. 2, then the probability of measuring s is*

$$\left(\frac{Tr(Z^s O_f)}{2^n}\right)^2$$

In particular, if $f(x) = x \cdot s$ mod 2 for some $t \in \{0,1\}^n$, then the probability of measuring s is 1 if $s = t$ and 0 otherwise.

Proof The probability of measuring s is given by

$$Pr(s) = |\langle s H^{\otimes n} O_f H^{\otimes n} 0\rangle|^2$$

But, from Lemma 1 we get

$$H^{\otimes n} O_f H^{\otimes n} = \sum_{t \in \{0,1\}^n} a_t H^{\otimes n} Z^t H^{\otimes n} = \sum_{t \in \{0,1\}^n} a_t X^t$$

and, consequently, it holds that

$$Pr(s) = \left| \langle s| \sum_{t \in \{0,1\}^n} a_t X^t |0\rangle \right|^2 = \left| \sum_{t \in \{0,1\}^n} a_t \langle s|X^t|0\rangle \right|^2 = |a_s|^2$$

because $\langle s|X^t|0\rangle = \langle s|t\rangle = \delta_{s,t}$. The result now follows from the expression of a_t given in Lemma 1 and the fact that both Z^s and O_f are real operators. □

Lemma 3 *Consider f a Boolean function of n variables and define $S = \{s \in \{0,1\}^n : f(s) = 1\}$. For a fixed binary string $t \neq 0$, define also $T_0 = \{s \in S : s \cdot t = 0\}$ and $T_1 = S \backslash T_0$. Then*

a) $Tr(I \cdot O_f) = 0 \iff |S| = 2^{n-1}$ *(* $\iff$ *f is balanced)*
b) For $t \neq 0$, $Tr(Z^t \cdot O_f) = 0 \iff |T_0| = |T_1| = \frac{|S|}{2}$

Proof To prove the first equivalence, we just need to notice that $O_f = I - 2\sum_{s \in S} |s\rangle\langle s|$ and thus

$$Tr(I \cdot O_f) = Tr(I) - 2\sum_{s \in S} Tr(|s\rangle\langle s|) = 2^n - 2|S|$$

For the second equivalence, notice that if $t \neq 0$ then

$$\begin{aligned} Tr(Z^t \cdot O_f) &= Tr(Z^t) - 2\sum_{s \in S} Tr(Z^t|s\rangle\langle s|) = 0 - 2\sum_{s \in S}\langle s Z^t s\rangle \\ &= -2\sum_{s \in S}\langle s(-1)^{t \cdot s} s\rangle = -2\sum_{s \in S}(-1)^{t \cdot s} = -2\left(\sum_{s \in T_0} 1 - \sum_{s \in T_1} 1\right) \\ &= -2(|T_0| - |T_1|). \end{aligned}$$

□

As a consequence of these lemmas, we have the following necessary condition for a Boolean function to be part of a 1-qe promise problem.

Proposition 1 *If f is a Boolean function of n variables that is in a 1-qe promise problem, then the size of* $S = \{s \in \{0, 1\}^n : f(s) = 1\}$ *must be even.*

Proof Suppose that $\{F_i\}_{i=1}^m$ is 1-qe promise problem and that $f \in F_i$. Then, there exists a partition $\{B_i\}_{i=1}^m$ of $\{0, 1\}^n$ such that when we use O_f in the circuit of Fig. 2 we always obtain as a result a string in B_i. It is clear from Definition 2 that every B_i is non-empty, and since $m \geq 2$, there exists a string $s \in \{0, 1\}^n$ such that $s \notin B_i$. Then, the probability of obtaining s when using O_f in the circuit of Fig. 2 is 0 and it follows from Lemma 2 that $Tr(Z^s O_f) = 0$. This implies, on virtue of Lemma 3, that $|S|$ must be even. □

The condition in Proposition 1 is not sufficient for a function to be part of a 1-qe problem, as the next example shows.

Example 1 The Boolean function f on 4 variables that takes value 1 on 0110, 0111, 1001, 1011, 1101, 1110 and 0 on the rest of strings is not part of any 1-qe problem. Indeed, it can be readily seen that $Tr(Z^t O_f) \neq 0$ for every string $t \in \{0, 1\}^4$.

However, we can give some sufficient conditions for Boolean functions to be included in 1-qe problems. We state and prove them in the following proposition.

Proposition 2 *Consider f a Boolean function on n variables and* $S = \{s \in \{0, 1\}^n : f(s) = 1\}$. *Then, if any of the following conditions holds, there exists a 1-qe promise problem that includes f:*

1. $|S| = 0$
2. $|S| = 2^n$
3. $|S| = 2^{n-1}$
4. $|S| = 2$
5. $|S| = 2^n - 2$
6. $|S| = 4$
7. $|S| = 2^n - 4$
8. $n = 1$
9. $|S|$ *is even and* $n = 2$
10. $|S|$ *is even and* $n = 3$

Proof 1. $f(s) = 0$ for every string s, so f is part of Deutsch–Jozsa.

2. $f(s) = 1$ for every string s, so f is part of Deutsch–Jozsa.
3. f is balanced, so f is part of Deutsch–Jozsa.
4. Let s_1 and s_2 be the two strings on which f takes value 1. They differ on at least a bit, say in position i. If we consider the string t which is 0 on every bit but the ith, then Z^t verifies condition b) in Lemma 3 and we have $Tr(Z^t O_f) = 0$. Thus, the promise problem with $F_1 = \{f\}$ and $F_2 = \{g\}$, where g is the linear Boolean function $g(s) = t \cdot s \bmod 2$, is 1-qe (with $B_1 = \{0, 1\}^n \backslash \{t\}$ and $B_2 = \{t\}$).
5. Consider $g(s) = f(s) \oplus 1$ and notice that $O_g = -O_f$ and that f verifies condition 4.
6. Let s_1, s_2, s_3 and s_4 be the four strings on which f takes value 1. If there exists a position in which two of them are 0 and the other two are 1, we can proceed as in 4. In other case, for each position in which the strings differ, there are exactly 3 which are equal and the other one is different. Consider a position i in which those three strings differ from the fourth. We can suppose, without loss of generality, that s_1 is 0 on that position while the other three strings are 1. (If not, we can reorder the strings, and the case in which the value of the bit is 1 is analogous.) Now, there must be another position j in which s_2, s_3 and s_4 differ. Again without loss of generality, we can suppose that s_2 is different from s_3 and s_4 on that position. Then, s_1 must be equal to s_3 and s_4 in position j. We consider the string t that is 1 on positions i and j and 0 everywhere else. Then, if we define $h(s) = t \cdot s \bmod 2$ we have $h(s_1) = h(s_2) \neq h(s_3) = h(s_4)$ and we can apply case b in Lemma 3.
7. As in 5, consider $g(s) = f(s) \oplus 1$ to reduce to case 6.
8. All Boolean functions on 1 variable are included in Deutsch's problem.
9. We have $2^n = 4$ and thus $|S|$ is either 0, 2 or 4 and we are in one of the previous cases.
10. We have $2^n = 8$ and thus $|S|$ is either 0, 2, 4, 6 or 8 and we are in one of the previous cases. □

Observe that because of Proposition 2, Example 1 is minimal in the number of variables and size of S.

To close this section, we give a characterisation of 1-qe promise problems. To do that, we first need to define a notion that will also prove useful in the rest of this paper.

Definition 5 Given f a Boolean function of n variables, we denote

$$T(f) := \{s \in \{0,1\}^n : Tr(Z^s O_f) \neq 0\}$$

and if F is a set of Boolean functions we define

$$T(F) := \cup_{f \in F} T(f).$$

Theorem 1 *Consider, $P = \{F_i\}_{i=1}^m$, a promise problem on Boolean functions of n variables. Then P is 1-qe if and only if $T(F_i) \cap T(F_j) = \emptyset$ whenever $i \neq j$.*

Proof $\Leftarrow$ We can define

- $B_i = T(F_i)$, for $i = 1, \ldots, m-1$
- $B_m = \{0,1\}^n \setminus \cup_{i=1}^{m-1} B_i$

Lemma 2 shows that this partition makes the problem 1-qe.

$\Rightarrow$ Since P is 1-qe, there exists a partition $\{B_i\}_{i=1}^m$ of $\{0,1\}^n$ such that when we take $f \in F_i$ and we use the circuit of Fig. 2 with O_f we always obtain a string from B_i. Lemma 2 then implies that $T(F_i) \subseteq B_i$ and the result follows because the sets B_i are pairwise disjoint. □

3 A poset of quantum exact promise problems

In this section, we define a natural order relationship among promise problems and we show that Deutsch–Jozsa and Bernstein–Varizirani are maximal and minimal elements in the poset of 1-qe problems for a fixed number of variables. Then, we study some of the automorphisms of the poset. We also introduce some notions that allow us to prove that it is connected.

We begin by defining the order relationship.

Definition 6 Let $\{F_i\}_{i=1}^{m_1}$ and $\{G_j\}_{j=1}^{m_2}$ be promise problems on n variables. We say that $\{F_i\} \leq \{G_j\}$ if for each i there exists j such that $F_i \subseteq G_j$.

Obviously, $\leq$ is reflexive and transitive. It is also antisymmetric, for suppose $\{F_i\} \leq \{G_j\}$ and $\{G_j\} \leq \{F_i\}$. Then, for every F_k we know that there exists G_l such that $F_k \subseteq G_l$. Also, there exists F_i such that $G_l \subseteq F_i$, which implies $F_k \subseteq F_i$. But all the sets in $\{F_i\}$ are pairwise disjoint, so this implies $F_k = F_i$ and, hence, $F_k = G_l$. Therefore, $\{F_i\} = \{G_j\}$.

As an example of this order relationship, notice, for instance, that for fixed n, Bernstein–Vazirani $\leq$ Deutsch–Jozsa, because all nonzero linear functions are balanced.

Now, we define some additional notions that will help us proving properties of the poset of 1-qe problems.

Definition 7 Consider a 1-qe problem $\{F_i\}_{i=1}^m$. We say that the problem is complete if for every $f \notin \cup_{i=1}^m F_i$ there exist $i \neq j$ such that $T(f) \cap T(F_i) \neq \emptyset \neq T(f) \cap T(F_j)$.

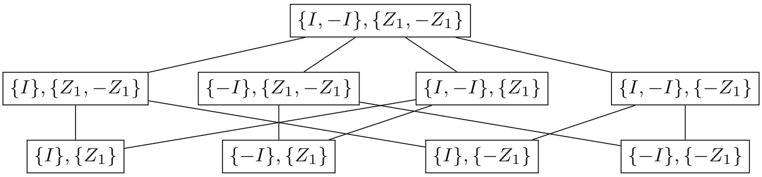


Fig. 3 Hasse diagram of the poset of 1-qe problems for $n = 1$

The intuitive meaning of this definition is that if a problem is complete, then you cannot extend it, to form another 1-qe problem, by including a new Boolean function in any of its sets (for it already includes all the Boolean functions that can be exactly distinguished with those outputs).

A related notion is given in the following definition.

Definition 8 Consider a promise problem $\{F_i\}_{i=1}^m$ on n variables. We say that it is exhaustive if for every $s \in \{0,1\}^n$ there exists i such that $s \in T(F_i)$.

Notice that a complete problem is also exhaustive. In fact, suppose that $\{F_i\}_{i=1}^m$ is not exhaustive. Then, there exists $s \in \{0,1\}^n$ such that $s \notin T(F_i)$ for every $i = 1, \ldots, m$. Then, we can consider $f = Z^s$ and we have $T(f) = \{s\}$ and, thus, $T(f) \cap T(F_i) = \emptyset$ for every i and the problem cannot be complete. However, the converse is not true. For instance, both Deutsch–Jozsa and Bernstein–Vazirani are exhaustive, but only Deutsch–Jozsa is complete (notice that $-I$ is not in Bernstein–Vazirani but $T(-I) = 0^n$, which only intersects one of the $T(F_i)$ of the problem).

In the following, we are interested in studying the structure of the poset of 1-qe problems for a fixed number of variables n. We will focus first on its minimal and maximal elements, because it is clear that in general, there are no minimum or maximum elements. For instance, it is easy to see that $\{\{I\},\{Z_1\}\}$ and $\{\{-I\},\{Z_1\}\}$ are incomparable and there cannot be any element that is strictly smaller than any of them. For $n = 1$, the poset (see Fig. 3 for its structure) possesses a maximum element, which is Deutsch–Jozsa because it contains all possible Boolean functions and they cannot be partitioned in a different way because $T(I) = T(-I) = \{0\}$ and $T(Z_1) = T(-Z_1) = \{1\}$. However, for $n > 1$ there is no maximal element. To see it, we first prove the following proposition.

Proposition 3 *For $n \geq 1$, Deutsch–Jozsa is a maximal element of the poset of 1-qe problems (and also a maximum element if $n = 1$).*

Proof It follows easily from noticing that Deutsch–Jozsa only has two elements and it is complete. □

If we now consider $n > 1$ and define

$$F_1 = \{f \in 2^{\{0,1\}^n} : T(f) \subseteq \{(0,\ldots,0,0),(0,\ldots,0,1)\}\}$$

and

$$F_2 = \{f \in 2^{\{0,1\}^n} : T(f) \subseteq \{0,1\}^n \backslash \{(0,\ldots,0,0),(0,\ldots,0,1)\}\}$$

then it is clear that $\{F_1, F_2\}$ and Deutsch–Jozsa are incomparable and that $\{F_1, F_2\}$ is also maximal (it is complete and has only two elements).

The number of maximal elements can be determined with the following proposition.

Proposition 4 *The poset of 1-qe problems on n variables has* $2^{2^n-1} - 1$ *maximal elements.*

Proof Clearly, any maximal problem must have exactly two sets of functions. (Otherwise, we can obtain join two of the sets to obtain a new problem which is above the original one.) For any $\emptyset \subsetneq B \subsetneq \{0, 1\}^n$, we can consider

$$F_1 = \{T(f) \subseteq B\}$$

and

$$F_2 = \{T(f) \subseteq \{0, 1\}^n \backslash B\}$$

It is easy to see that $\{F_1, F_2\}$ is maximal and that any maximal problem must be of this form. There are $2^{2^n} - 2$ choices for B. However, choosing B and $\{0, 1\}^n \backslash B$ leads to the same problem. Thus, the total number of possibilities is $2^{2^n-1} - 1$. □

Notice, also, that every maximal element is necessarily complete and thus exhaustive.

Now, we turn our attention to minimal elements. It is easy to see that if f_1 and f_2 are Boolean functions such that $T(f_1) \cap T(f_2) = \emptyset$ then $\{\{f_1\}, \{f_2\}\}$ is minimal in the poset of 1-qe problems. A situation a little bit more interesting appears when we restrict ourselves to exhaustive problems. This is addressed in the next proposition.

Proposition 5 *There are at least* 2^{2^n} *minimal elements in the poset of 1-qe exhaustive problems on n variables and Bernstein–Vazirani is one of them.*

Proof For $s \in \{0, 1\}^n$, we define F_s as either $\{Z^s\}$ or $\{-Z^s\}$. It is clear that, then, $\{F_s\}$ is exhaustive and that there is no other exhaustive problem G such that $G < \{F_s\}$. The number of such problems is exactly 2^{2^n} and when choose $F_s = \{Z^s\}$ for all s we obtain the Bernstein–Vazirani problem. □

We now consider two natural transformations of 1-qe problems that, in fact, induce automorphisms of the poset (that is, bijective transformations that preserve the order of the elements). We start with the simplest one, which simply takes each function f in a problem to its complement (i.e. the function that is 0 when f is 1 and 1 when f is 0).

Theorem 2 *Consider the transformation* $\mathcal{I}$ *that takes* $\{F_i\}_{i=1}^m$ *to* $\{F_i'\}_{i=1}^m$ *given by*

$$F_i' = \{f \oplus 1 : f \in F_i\}$$

Then, $\mathcal{I}$ *is an automorphism of the poset of 1-qe problems and Deutsch–Jozsa is one of its fixed points.*

Proof Notice that if the oracle associated with f is O_f, then the oracle associated with $f \oplus 1$ is $-O_f$. As a consequence, $\{F_i'\}_{i=1}^m$ is a 1-qe problem and if $F \leq G$, then $\mathcal{I}(F) \leq \mathcal{I}(G)$. Also, $\mathcal{I}$ is bijective because it is its own inverse, so it is an automorphism. Finally, it is clear that $\mathcal{I}$ takes Deutsch–Jozsa to itself, because the complement of a balanced function is balanced and the complement of the constantly 0 function is the constantly 1 function. □

The other transformation of 1-qe problems depends on the choice of a string $s \in \{0,1\}^n$ and it is given in the following theorem.

Theorem 3 *Given $s \in \{0,1\}^n$, consider the transformation $\mathcal{L}_s$ that takes $\{F_i\}_{i=1}^m$ to $\{F_i'\}_{i=1}^m$ given by*

$$F_i' = \{f \oplus l_s : f \in F_i\}$$

where l_s is the linear form $l_s(x) = x \cdot s \bmod 2$. Then, $\mathcal{L}_s$ is an automorphism of the poset of 1-qe problems and Bernstein–Vazirani is one of its fixed points.

Proof The proof is similar to the one of the previous theorem. We only need to note that if the oracle associated with f is O_f, then the oracle associated with $f \oplus l_s$ is $O_f Z^s$. From this, it follows easily that $\{F_i'\}_{i=1}^m$ is 1-qe and that $\mathcal{L}_s$ preserves the order. Moreover, $\mathcal{L}_s$ is its own inverse and, hence, an automorphism. Since the xor of linear Boolean functions is, again, a Boolean linear function, it also follows that $\mathcal{L}_s$ fixes the Bernstein–Vazirani problem. □

It is interesting to note that if we apply $\mathcal{L}_s$ to the Deutsch–Jozsa problem, then we recover the generalised Deutsch–Jozsa problems introduced in Section 4 of [3].

Another property of these automorphisms is that they preserve the properties of being exhaustive and complete.

Proposition 6 *If F is a 1-qe complete (resp. exhaustive) problem, then $\mathcal{L}_s(F)$, for all $s \in \{0,1\}^n$ and $\mathcal{I}(F)$ are complete (resp. exhaustive).*

Proof Suppose that $F = \{F_i\}_{i=1}^m$ is complete. Consider $\mathcal{I}(F) = \{F_i'\}_{i=1}^m$. Then, $T(F_i') = T(F_i)$ and, hence, $\mathcal{I}(F)$ is also complete. The same argument shows that if f is exhaustive, then $\mathcal{I}(F)$ is exhaustive.

Now, fix $s \in \{0,1\}^n$ and consider $\mathcal{L}_s(F) = \{F_i''\}_{i=1}^m$. It is easy to see that $T(F_i'') = T(F_i) \oplus s$, from which, again, it follows that $\mathcal{L}_s(F)$ is complete if F is complete and exhaustive if F is exhaustive. □

To complete this section, we are now going to prove that the poset of 1-qe problems is connected.

Theorem 4 *For each fixed n, the poset of 1-qe problems on n variables is connected.*

Proof It is enough to prove that, given $F = \{F_i\}_{i=1}^m$, it is connected to Bernstein–Vazirani. Define G by

$$G_i = \{f \in 2^{\{0,1\}^n} : T(f) \subseteq T(F_i)\}$$

for $i = 1, \dots, m-1$ and

$$G_m = \{f \in 2^{\{0,1\}^n} : T(f) \subseteq \{0,1\}^m \setminus \cup_{i=1}^{m-1} T(F_i)\}.$$

Clearly, $F \leq G$ and G is complete and, thus, exhaustive. Then, in particular, every Z^i is in some set of G and, thus, Bernstein–Vazirani is below G and, consequently, connected to F. □

4 Complexity with classical deterministic algorithms

This section shows that the classical query complexity of the 1-qe problems we have been studying can have a wide range of values. In fact, in the two theorems of this section we give constructions to obtain 1-qe problems with particular classical query complexities.

Theorem 5 *Fix $n \geq 1$ and consider k such that $1 \leq k \leq 2^{n-1}+1$. Then, there exists a 1-qe problem that, with a deterministic classical algorithm, requires exactly k queries to be solved.*

Proof Consider A a set of exactly k binary strings of length n. Consider, also, B the set of Boolean functions f such that f is balanced and there exists $s \in A$ such that $f(s) = 1$.

Clearly, $\{\{I\}, B\}$ is a 1-qe problem, because if $f \in B$, then f is balanced and thus the probability of obtaining 0 when using the circuit of Fig. 2 is 0, while for I that probability is 1.

Obviously, the problem can be solved with k classical queries because, given f, it is enough to check $f(s)$ for every $s \in A$. If $f(s) = 1$ for some s, then $f \in B$. Otherwise, $f = I$ (remember that we are using phase oracles and, hence, I is the Boolean function that is identically 0).

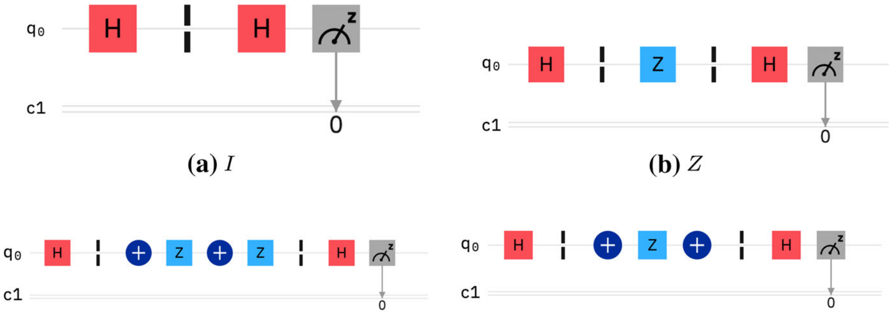
However, $k-1$ classical queries are not sufficient to solve the problem. In fact, for any set C of $k-1$ strings there exists $f \in B$ such that $f(s) = 0$ for every $s \in C$. To see this, notice that $k-1 \leq 2^{n-1}$ and then we can define f such that

1. $f(s) = 0$ for every $s \in C$
2. $f(s) = 1$ for every $s \in A \backslash C$
3. f is balanced

In order to prove this, we need to consider two possible cases. If $A \cap C = \emptyset$ then necessarily $k-1 \leq 2^{n-1}-1$, because if we had $k-1 = 2^{n-1}$ then we would have $|C| = 2^{n-1}$ and $|A| = 2^{n-1}+1$ and then these two sets would not be disjoint. Then, f is 0 on $k-1$ strings, 1 on k strings and since $k + (k-1) = 2k-1 \leq 2^n - 1$ we have enough "unset" strings to make f balanced.

On the other hand, if $A \cap C \neq \emptyset$ then $|C| = k-1 \leq 2^{n-1}$ and $|A \backslash C| \leq k-1 \leq 2^{n-1}$ so, again, we have enough free strings to make f balanced. □

The construction used in the proof of the previous theorem can be extended to obtain uniform families of 1-qe problems with given classical query complexities, as the following corollary shows.



(a) I **(b)** Z

(c) $-I$ **(d)** $-Z$

Fig. 4 Original circuits with phase oracles

Corollary 1 *If* $g : \mathbb{N} \to \mathbb{N}$ *is a computable function such that* $1 \leq g(n) \leq 2^{n-1} + 1$ *for each* n*, then there exists a uniform family of 1-qe problems* F_n *such that the classical query complexity of* F_n *is exactly* $g(n)$*.*

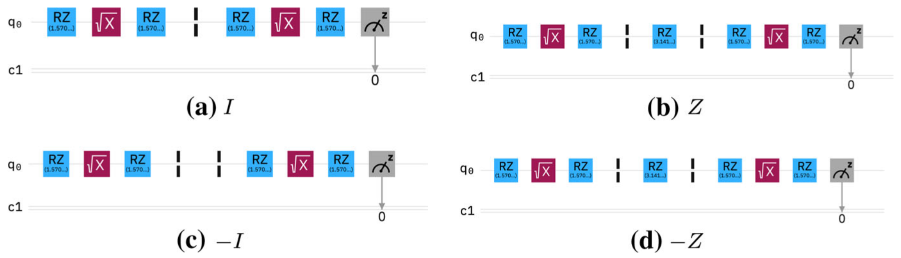
Proof We can simply use the construction in the proof of Theorem 5 with $A = A_n =$ {The first g(n) strings in $\{0, 1\}^n$} to construct F_n. Then, the classical query complexity of F_n is exactly $g(n)$. □

5 Experiments on quantum hardware

In this section, we present some experiments that we have conducted on actual quantum computers with the type of problems that we have introduced in this paper. Although, as we have shown, 1-qe problems can be solved exactly with just one quantum query, this presupposes the existence of fault-tolerant quantum computers, while the devices that are available today are still subject to noise and gate and readout errors.

To test the possibility of solving 1-qe problems on current quantum computers, we have designed phase oracles for the four Boolean functions involved in the 1-qe problems of Fig. 3 and implemented them on one of the quantum devices accessible through IBM Quantum [12]. Namely, we have used *ibmq_armonk*, an IBM Quantum Canary Processor of one qubit. The implementations of these phase oracles were then integrated in the circuit of Fig. 2 to obtain the circuits shown in Fig. 4. We have included barriers on both sides of the oracle to prevent cancellation or simplifications of the Hadamard gates with the gates of the oracle (which must be treated as a black box). This leads to the transpiled circuits shown in Fig. 5, which are the ones actually executed on the quantum computer.

Notice that the IBM Quantum transpiler detects the presence of a physically irrelevant global phase in the circuits for $-I$ and $-Z$ and generates exactly the same transpiled circuits for I and $-I$ and for Z and $-Z$. For this reason, we only need to actually run the circuits of Fig. 5a, b. We executed these two circuits on both the simulator and on the *ibmq_armonk* quantum processor, where we run each circuit 75 times with 8192 shots (or samples) each time. These are the maximum numbers allowed at IBM Quantum. The simulator always obtained the correct results, as expected from



(a) I

(b) Z

(c) $-I$

(d) $-Z$

Fig. 5 Transpiled circuits with phase oracles

Table 1 Results on the actual quantum computer

Circuit	Probability of 0	Probability of 1
$I/-I$	0.9345 ± 0.0029	0.0655 ± 0.0029
$Z/-Z$	0.0945 ± 0.0030	0.9054 ± 0.030

our mathematical proofs. The average and standard deviations of the results on the actual quantum computer are shown in Table 1.

The average probability of success for solving any of the 1-qe problems of Fig. 3 would, then, be the average of the probability of measuring 0 with either I or $-I$ and of the probability of measuring 1 with either Z or $-Z$, giving a final result of 0.91995. This is strictly less than the probability 1 predicted by our mathematical results (and obtained with the simulator) and can be explained by the gate and readout errors and the noise present in actual quantum hardware. Moreover, the values obtained here are consistent with the ones reported in [2] for similarly simple circuits executed on the same quantum processor. In fact, in that work, it was also observed that a readout error of obtaining 0 when the correct value was 1 is usually higher than a readout error of obtaining 1 when 0 is the correct value.

6 Conclusions and future work

In this paper, we have introduced a generalisation of promise problems such as Deutsch–Jozsa or Bernstein–Vazirani and we have shown that all of them can be solved with just one oracle query in the quantum circuit model. We have also studied necessary and sufficient conditions for Boolean functions on n variables to be part of such promise problems.

Then, we have defined a natural order relationship among these problems and we have proved that the Deutsch–Jozsa and Bernstein–Vazirani problems are, under some conditions, maximal and minimal elements in the poset of these promise problems. We have also studied some of the automorphisms of the poset and shown that it is always connected.

Finally, we have proved that, although in the quantum setting one oracle query is enough to solve these promise problems, if we only use classical resources then the query complexity can vary from 1 to $2^{n-1}+1$, taking all the values in between, and we have also presented numerical results of the execution of some 1-qe problems on actual quantum hardware.

There are some open questions that we would like to explore in future works. An interesting problem is to give a more explicit characterisation of the Boolean functions that can take part in 1-qe problems. Also, we are interested in determining all the automorphisms of the poset of such problems. Finally, we would like to extend the study done in this paper to other quantum schemes (with different quantum circuits) that also allow to solve some promise problems with a small number of oracle queries.

Acknowledgements This work was supported in part by the Ministry of Economy, Industry and Competitiveness from Spain/FEDER under Grants TIN2017-87600-P, Grant MTM-2017- 83506-C2-2-P, by the Regional Ministry of the Principality of Asturias under Grants FC-GRUPIN-IDI/2018/000226 and Grant FC-GRUPIN- IDI/2018/000193, and by Campus de Excelencia Internacional of the University of Oviedo in collaboration with Banco de Santander, as part of the "ayudas económicas de movilidad de excelencia para docentes e investigadores de la Universidad de Oviedo". MINECO (Grant No. MINECO-16-TEC2015-67387-C4-3-R), MICINN (Grant No. RTI2018-098085-B-C44).

References

1. Bernstein, E., Vazirani, U.: Quantum complexity theory. SIAM J. Comput. **26**(5), 1411–1473 (1997). https://doi.org/10.1137/S0097539796300921
2. Combarro, E.F., Carminati, F., Vallecorsa, S., Ranilla, J., Rúa, I.F.: On protocols for increasing the uniformity of random bits generated with noisy quantum computers. J. Supercomput. (2021) **in press**
3. Combarro, E.F., Piñera Nicolás, A., Ranilla, J., Rúa, I.F.: An explanation of the Bernstein–Vazirani and Deustch–Josza algorithms with the quantum stabilizer formalism. Comput. Math. Methods e1120 (2020)
4. Combarro, E.F., Ranilla, J., Rúa, I.F.: A quantum algorithm for the commutativity of finite dimensional algebras. IEEE Access **7**, 45554–45562 (2019)
5. Combarro, E.F., Ranilla, J., Rúa, I.F.: Quantum walks for the determination of commutativity of finite dimensional algebras. J. Comput. Appl. Math. **354**, 496–506 (2019)
6. Combarro, E.F., Ranilla, J., Rúa, I.F.: Quantum abstract detecting systems. Quantum Inf. Process. **19**, 258 (2020)
7. Deutsch, D.: Quantum theory, the Church–Turing principle and the universal quantum computer. Proc. R. Soc. Lond. Ser. A **400**, 97–117 (1985)
8. Deutsch, D., Jozsa, R.: Rapid solution of problems by quantum computation. Proc. R. Soc. Lond. A Math. Phys. Eng. Sci. **439**(1907), 553–558 (1992)
9. Feynman, R.: Simulating physics with computers. Int. J. Theor. Phys. **21**(6), 467–488 (1982)
10. Figgatt, C., Maslov, D., Landsman, K.A., et al.: Complete 3-Qubit Grover search on a programmable quantum computer. Nat. Commun. **8**, 1918 (2017)
11. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing, STOC '96, pp. 212–219. ACM, New York, NY, USA (1996)
12. IBM Quantum (2021). https://quantum-computing.ibm.com/

13. Kitaev, A.Y.: Quantum measurements and the abelian stabilizer problem. Electron. Colloq. Comput. Complex.: ECCC **3** (1995)
14. Manin, Y.: Vychislimoe i nevychislimoe. Sov. Radio, pp. 13–15 (1980)
15. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information: 10th Anniversary. Cambridge University Press, Cambridge (2011)
16. Santos, R.A.M.: Szegedy's quantum walk with queries. Quantum Inf. Process. **15**(11), 4461–4475 (2016)
17. Shor, P.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings of FOCS, pp. 124–134 (1994)
18. Simon, D.R.: On the power of quantum computation. In: Proceedings of the 35th Annual Symposium on Foundations of Computer Science, SFCS '94, pp. 116–123. IEEE Computer Society, USA (1994). https://doi.org/10.1109/SFCS.1994.365701
19. Szegedy, M.: Quantum speed-up of Markov chain based algorithms. In: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, FOCS '04, pp. 32–41. IEEE Computer Society, Washington, DC, USA (2004)
20. Wong, T.: Faster search by lackadaisical quantum walk. Quantum Inf. Process. **17**(68) (2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.