On the Probabilistic Query Complexity of Transitively Symmetric Problems

Pascal Koiran, Vincent Nesme, Natacha Portier

On the Probabilistic Query Complexity of Transitively Symmetric Problems *

Pascal Koiran, Vincent Nesme and Natacha Portier
Laboratoire de l'Informatique du Parallélisme†
Ecole Normale Supérieure de Lyon

19th December 2006

Abstract

We obtain optimal lower bounds on the nonadaptive probabilistic query complexity of a class of problems defined by a rather weak symmetry condition. In fact, for each problem in this class, given a number T of queries we compute *exactly* the performance (i.e., the probability of success on the worst instance) of the best nonadaptive probabilistic algorithm that makes T queries. We show that this optimal performance is given by a minimax formula involving certain probability distributions. Moreover, we identify two classes of problems for which adaptivity does not help.

We illustrate these results on a few natural examples, including unordered search, Simon's problem, distinguishing one-to-one functions from two-to-one functions, and hidden translation. For these last three examples (which are of particular interest in quantum computing), the recent theorems of Aaronson, of Laplante and Magniez, and of Bar-Yossef, Kumar and Sivakumar on the probabilistic complexity of black-box problems do not yield any nonconstant lower bound.

1 Introduction

There has been in the past few years a surge of interest for lower bounds in the black-box model, motivated in particular by the study of quantum algorithms. Indeed, since quantum circuit lower bounds seem very difficult to obtain, most of the known quantum lower bounds have been derived in the black-box setting ([11], which shows how to simulate classically certain families of constant-depth quantum circuits, may be considered an exception). Two methods proved particularly successful: the polynomial method and the adversary method. We will not give exhaustive references here, and will just point out [7] and [2] for the polynomial

*Rapport de recherche du LIP RR2006–27

†UMR CNRS - ENS Lyon - UCB Lyon - INRIA 5668.
Email addresses: `Firstname.Lastname@ens-lyon.fr`

method as well as [4] and [24] for the adversary method. There was recently some unexpected feedback from quantum to probabilistic complexity: inspired by quantum adversary lower bounds, Aaronson [1] and Laplante and Magniez [19] obtained new lower bounds on probabilistic query complexity. Applications to sorting, ordered search [19], local search [1] and Sperner problems [13] were given. Earlier probabilistic query lower bounds were often obtained by ad-hoc arguments. As pointed out in [1], with a general method one can more easily "focus on what is unique about a problem, and ignore what is common among many problems". Like their quantum ancestors, the lower bounds of [1, 19] are very general (they apply to any black-box problem) and nevertheless give optimal results for some natural problems. They unfortunately suffer from the same drawback as their ancestors, namely, they cannot yield any nonconstant lower bound for promise problems such that every positive instance is "far away" from every negative instance. Note that the polynomial method does not suffer from this drawback [2, 15, 16, 18].

The contribution of this paper is twofold. First, we identify a class of problems, dubbed "transitively symmetric problems", for which optimal lower bounds on the *nonadaptive* probabilistic query complexity can be obtained. Our lower bound method is close in spirit to the adversary method. More precisely, for each problem in this class, given a number T of queries we compute *exactly* the performance (i.e., the probability of success on the worst instance) of the best nonadaptive probabilistic algorithm that makes T queries. We show that this optimal performance is given by a minimax formula involving certain probability distributions. A precise definition of the class of transitively symmetric problems is given at the beginning of section 3. The idea is that the automorphism group of the problem must act transitively on the set of positive instances as well as on the set of negative instances. The elements of the automorphism group act by permutations on the domain of the black-box function and by permutations on its range. For instance, when arbitrary permutations on the domain but no permutation of the range (except the identity) are allowed the usual notion of symmetric function is recovered. A lower bound for approximating such functions can be found in [6]. By contrast, in some of the examples studied in section 3, permutations on the domain come from a strict subgroup of the symmetric group but arbitrary permutations of function values are allowed. According to [6], "an important open problem is to find tight lower bounds for the query complexity of non-symmetric functions". In this paper we make a step in this direction since we work with a weaker notion of symmetry than the usual one.

The restriction to nonadaptive algorithms is of course rather severe. Our second contribution is the identification of two classes of problems for which adaptivity does not help. The first one is the class of problems that are symmetric in the usual sense of this word, i.e., are invariant under arbitrary permutations on the domain of the black-box. This observation was already implicit in [6]. The second class is the class of *collision problems* that are invariant under arbitrary permutations on the range of the black-box.

We illustrate these result on several natural problems: search in an unordered list, distinguishing 1-to-1 functions from 2-to-1 functions, Simon's problem, and the hidden translation problem. Namely, we show that adaptivity does not help for any of these problems, and we give optimal lower bounds on their probabilistic query complexity. The first two problems are symmetric in the usual sense, but the last three are not. The results of [1] and [19] yield a nonconstant lower bound only for the first problem. In the quantum setting, no nonconstant lower bound is known for the hidden translation problem. Note that symmetry considerations are essential in recent applications of the polynomial method [2, 15, 16, 18]. Our paper may be seen as a systematic attempt to incorporate such considerations into the probabilistic adversary method. Symmetry considerations also play an important role in a recent quantum version of the adversary method [3].

Organization of the paper. The probabilistic query model is defined in section 2. Transitively symmetric problems are defined at the beginning of section 3, and their probabilistic query complexity is computed in Theorem 1. We explain in section 4 why the restriction to nonadaptive algorithms in Theorem 1 is essential, and we present the two classes of problems for which adaptivity does not help. Several examples are discussed in section 5. Finally, the relations between Theorem 1, the variation distance and the results of [6] are discussed in section 6. In particular, we show that the methods of [6], based on the block sensitivity and on the Hellinger distance, do not yield any nonconstant lower bound on a problem which is as symmetric as one might wish: the 1-to-1 versus 2-to-1 problem. These methods also do not yield any nonconstant lower bound for Simon's problem and Hidden Translation, which are subproblems of the 1-to-1 versus 2-to-1 problem.

2 The probabilistic query model

We define *black-box problems* to be partial functions $\mathscr{P}$ from $[M]^{[N]}$ to $[L]$, where N, M and L are positive integers and $[n]$ stands for $\{0; 1; \ldots; n-1\}$. In the sequel $L = 2$ and we call X the set of function $f \in [M]^{[N]}$ such that $\mathscr{P}(f) = 1$ and Y the set of function $g \in [M]^{[N]}$ such that $\mathscr{P}(g) = 0$.

Next we define our model of a probabilistic algorithm.

Such an algorithm is defined by the following data: a probability distribution on the space $\Omega = \{0,1\}^t$ of internal random bits, a function h_1 from Ω to $[N]$, h_2 from $\Omega \times [M]$ to $[N]$, ..., h_T from $\Omega \times [M]^{T-1}$ to $[N]$ and finally a function O from Ω $[M]^T$ to $\{0,1\}$. By definition, T is the query complexity of A. On a black box function f, the algorithm works as follows:

- Choose randomly $\omega \in \Omega$.
- Compute $i_1 = h_1(\omega)$; query i_1 and set $j_1 = f(i_1)$.
- Compute $i_2 = h_2(\omega, j_1)$; query i_2 and set $j_2 = f(i_2)$.
- ...
- Compute $i_s = h_s(\omega, j_1, \ldots, j_{s-1})$; query i_s and set $j_s = f(i_s)$.
- Output $O(\omega, j_1, \ldots, j_s)$.

As there is clearly no need for an algorithm to perform the same query twice, we will always assume, unless explicitly stated otherwise, that the queries are distinct. Formally, this means that we assume $h_k(\omega, j_1, \ldots, j_{k-1})$ is never equal to $h_1(\omega)$, $h_2(\omega, j_1)$, ... or $h_{k-1}(\omega, j_1, \ldots, j_{k-2})$.

An algorithm A solving $\mathscr{P}$ queries a black-box function to decide whether it belongs to X or to Y. The algorithm succeeds on a black-box $f \in X$ if it decides that $f \in X$, that is, if its output is equal to 1. Similarly, the algorithm succeeds on a black box $g \in Y$ if it decides that $g \in Y$, that is, if its output is equal to 0. If A is a probabilistic algorithm its success probability ε is its worst case success probability, that is the minimum over all $f \in X \cup Y$ of the success probability of A with black-box f. By contrast, the average success probability of an algorithm is relative to a given probability distribution on the set $X \cup Y$.

We say that an algorithm A is *nonadaptive* if the functions $h_2, \ldots, h_s$ do not depend on $j_1, \ldots, j_s$. A nonadaptive algorithm can thus be informally described in a simpler way: first choose the queries to be made, then perform them, at last decide to accept or reject the black-box based on the answers to the queries and on the values of your random bits.

Let us look at an example: the unordered search problem $\mathscr{P}_{\text{search}}$. Let N be an integer and $M = 2$. The set X only contains the constant zero function z. Let f_i be the function such that $f_i(i) = 1$ and $f_i(j) = 0$ if $j \neq i$. We set $Y = \{f_i | i \in [N]\}$. Consider the following simple-minded algorithm: choose $i \in [N]$ uniformly at random, query i and set $j = f(i)$. Decide that $f \in X$ if $j = 0$ and that $f \in Y$ if $j = 1$. The query complexity of this algorithm is 1. Its success probability for the function $z \in X$ is 1 and for a function $f \in Y$ is $1/N$, so its success probability is $1/N$. We can do better with another 1-query probabilistic algorithm: choose $i \in [N]$ uniformly at random, query i and set $j = f(i)$. With probability $(N-1)/(2N-1)$ decide that $f \in Y$; with probability $N/(2N-1)$ decide that $f \in X$ if $j = 0$ and that $f \in Y$ if $j = 1$. The success probability for the function $z \in X$ is $N/(2N-1)$. The success probability for a function $f \in Y$ is $(N-1)/(2N-1) + 1/(2N-1)$, that is $N/(2N-1)$. So the success probability of our algorithm is $N/(2N-1)$, which is much better than $1/N$. Can we do better? What happens if we allow more queries? We will be able to answer these questions thanks to the results of sections 3 and 4.

3 Nonadaptive query complexity of transitively symmetric problems

3.1 Statement of the theorem

As explained in the introduction, our theorem applies to black-box problem which are transitively symmetric. Here is a precise definition. Let $\mathfrak{S}_N$ and $\mathfrak{S}_M$ be the permutations group of respectively $[N]$ and $[M]$. We consider the group $\mathfrak{S}_N \times \mathfrak{S}_M$ endowed with the product $(\sigma', \tau')(\sigma, \tau) = (\sigma' \circ \sigma, \tau' \circ \tau)$.

Definition 1 *An* automorphism *of a black-box problem $\mathscr{P}$ is an element (σ, τ) of $\mathfrak{S}_N \times \mathfrak{S}_M$ under which $\mathscr{P}$ is invariant, i.e.*

(i) For every $f \in X$, $\tau \circ f \circ \sigma^{-1} \in X$.

(ii) For every $g \in Y$, $\tau \circ g \circ \sigma^{-1} \in Y$.

The automorphisms of $\mathscr{P}$ form a subgroup of $\mathfrak{S}_N$ $\mathfrak{S}_M$, which will be noted $\mathrm{Aut}\,(\mathscr{P})$.

Definition 2 *A subgroup G of $\mathfrak{S}_N \times \mathfrak{S}_M$* acts transitively *on a black-box problem $\mathscr{P}$ if:*

(i) $G \leq \mathrm{Aut}(\mathscr{P})$.

(ii) For every $(f, g) \in X^2 \cup Y^2$ there exists $(\sigma, \tau) \in G$ such that $g = \tau \circ f \circ \sigma^{-1}$.

We say that a black-box problem $\mathscr{P}$ is transitively symmetric *if $Aut(\mathscr{P})$ acts transitively on $\mathscr{P}$.*

For example, $\mathfrak{S}_N \times \{\mathrm{Id}\}$ acts transitively on $\mathscr{P}_{\mathrm{search}}$. This fact was used in the design of the two algorithms of section 2 since we chose $i \in [N]$ uniformly at random.

Let $\mathscr{P}$ be a black-box problem. Let I be a list of queries and B a set of possible answers. If the length of I is T then B is a subset of $[M]^T$. We define $P_I^X(B)$ as the proportion of functions f in X satisfying the condition $f(I) \in B$. Likewise, we define $P_I^Y(B)$ to be the proportion of functions g in Y satisfying the condition $g(I) \in B$. We can now state our main theorem.

Theorem 1 *Let $\mathscr{P}$ be a transitively symmetric black-box problem. The success probability of the best nonadaptive algorithm for $\mathscr{P}$ of query complexity T is equal to:*

$$\gamma = \min_{0 \leq p \leq 1} \max_{I \in [N]^T, B \subseteq [M]^T} pP_I^X(B) + (1-p)(1 - P_I^Y(B)).$$

In this formula for γ the maximum is taken over all lists of queries of length T. In particular, the same query may occur several times in I. It should come as no surprise that we can restrict our attention to lists of T *distinct* queries I as soon as $T \leq N$ (which is of course the case of interest). Indeed, suppose that query $i \in [N]$ appears at least twice in the list. We replace the second query i by an element $i' \in [N]$ which does not appear in the list I. This yields a new list I' of T queries. Consider now a set B of list of answers J. We are looking for a set B' of lists of answers such that $f(I) \in B$ iff $f(I') \in B'$, and thus $P_I^X(B) = P_{I'}^X(B')$ and $P_I^Y(B) = P_{I'}^Y(B')$. In a list $J \in B$, consider the two answers j_1 and j_2 to the query i. If $j_1 \neq j_2$ then no function f satisfies $f(I) = J$, and so suppose there is no such J in B. If $j_1 = j_2$ then replacing j_2 by all elements of $[M]$ yields a set B'_J of M lists of answers such that $f(I) = J$ iff $f(I') \in B'_J$. We then have the expected property for $B' = \cup_{J \in B} B'_J$.

This remark will be used later in this section in the study of several specific examples.

The proof of Theorem 1 is naturally divided into an upper bound proof (that is, the presentation of an efficient algorithm) and a lower bound proof. The upper bound is established in section 3.2 and the lower bound in section 3.3. Note that the lower bound is actually an upper bound on the success probability γ, and vice versa.

3.2 Proof of the upper bound

Given an algorithm A, we define the symmetrized algorithm $\tilde{A}$ as follows: $\tilde{A}$ simulates A on function $\tau f \sigma^{-1}$ where (σ, τ) is a random permutation uniformly distributed in $\mathrm{Aut}(\mathscr{P})$. For this purpose replace query i on $\tau f \sigma^{-1}$ by query $\sigma^{-1}(i)$ on f and then apply τ. More formally:

Definition 3 *Let $\mathscr{P}$ be a problem, and A an algorithm for $\mathscr{P}$. $\tilde{A}$, the* symmetrization *of A, is defined as follows:*

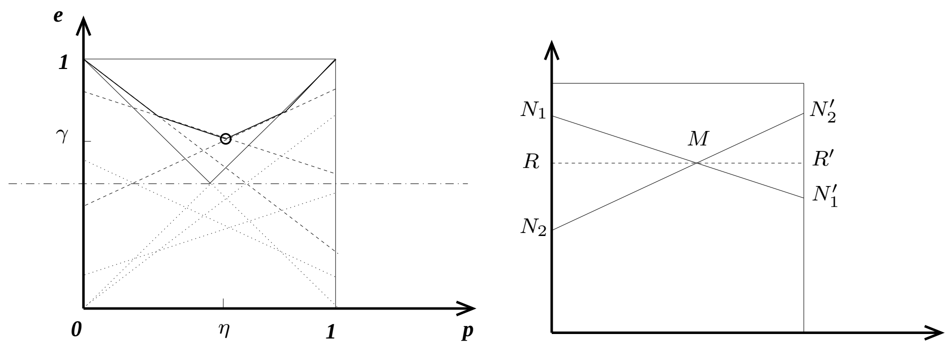

Figure 1: The lines $\Delta_{I,B}$ (left) and the minimum of F as intersection of two lines.

- *Choose randomly* $\omega \in \Omega$ *and* $(\sigma, \tau) \in \mathrm{Aut}(\mathscr{P})$.
- *Compute* $i_1 = h_1(\omega)$. *Query* $\sigma^{-1}(i_1)$ *and set* $j_1 = \tau f \sigma^{-1}(i_1)$.
- *Compute* $i_2 = h_2(\omega, j_1)$. *Query* $\sigma^{-1}(i_2)$ *and set* $j_2 = \tau f \sigma^{-1}(i_2)$.
- ...
- *Compute* $i_s = h_s(\omega, j_1, \ldots, j_{s-1})$. *Query* $\sigma^{-1}(i_s)$ *and set* $j_s = \tau f \sigma^{-1}(i_s)$.
- *Output* $O(\omega, j_1, \ldots, j_s)$.

Note that the success probability of $\tilde{A}$ (as defined in section 2) is at least equal to that of A.

Proposition 1 *Let $\mathscr{P}$ be a transitively symmetric black-box problem. There exists a nonadaptive algorithm for $\mathscr{P}$ of query complexity T and success probability γ, where*

$$\gamma = \min_{0 \le p \le 1} \max_{I \in [N]^T, B \subseteq [M]^T} \; pP_I^X(B) + (1-p)(1 - P_I^Y(B))$$

is defined as in Theorem 1.

Proof. For every list of queries I and set of answers B we have a line $\Delta_{I,B}$ of equation: $e = pP_I^X(B) + (1-p)(1 - P_I^Y(B))$. So $e = F(p) = \max_{I,A} \; pP_I^X(A) + (1-p)(1 - P_I^Y(A))$ is the equation of a continuous function F, piecewise affine and convex (see figure 1). By definition, its minimum value between 0 and 1 is γ. Lets η be a real number such that $\gamma = F(\eta)$. We first prove that $\gamma \ge 1/2$. For $B = \emptyset$ and any I, the equation of $\Delta_{I,B}$ is $e = 1 - p$. When B is equal to the set of all possible answers, that is, $B = [M]^T$, the equation of $\Delta_{I,B}$ is $e = p$. Thus F is everywhere greater or equal to the function $\max(p, 1-p)$, and its minimum γ greater or equal to $\min_{0 \le p \le 1} \max\{p, 1-p\} = 1/2$. A consequence of the inequality $F(p) \ge \max(p, 1-p)$ is that there are two list of queries I_1 and I_2 and two set of answers B_1 and B_2

such that $M = (\eta, \gamma)$ is the intersection of Δ_{I_1,B_1} and Δ_{I_2,B_2}. We now compute the coordinates of this intersection point. This is not strictly necessary for the proof, but it helps to understand what the algorithm really is. If we use the notations $P_1^X = P_{I_1}^X(B_1)$, $P_2^X = P_{I_2}^X(B_2)$, $P_1^Y = P_{I_1}^Y(B_1)$ and $P_2^Y = P_{I_2}^Y(B_2)$ we have

$$\gamma = \eta P_1^X + (1-\eta)(1-P_1^Y) = \eta P_2^X + (1-\eta)(1-P_2^Y)$$

so that:

$$\eta = \frac{P_1^Y - P_2^Y}{P_1^X - P_2^X + P_1^Y - P_2^Y}, \quad \gamma = \frac{P_2^X P_1^Y - P_1^X P_2^Y + P_1^X - P_2^X}{P_1^X - P_2^X + P_1^Y - P_2^Y}.$$

Consider now the points $N_1(0, 1-P_1^Y)$, $N_2(0, 1-P_2^Y)$, $R(0,\gamma)$, $N_1'(1, P_1^X)$, $N_2'(1, P_2)$ and $R'(1,\gamma)$ of Figure 1. There exists ζ between 0 and 1 such that R is the barycenter of N_1 and N_2 with respective weights ζ and $1-\zeta$. The triangles $(N_1 M N_2)$ and $(N_1' M N_2')$ are homothetic, so that R' is the barycenter of N_1' and N_2' with the same weights. It follows that

$$\gamma = \zeta P_1^X + (1-\zeta)P_2^X = \zeta(1-P_1^Y) + (1-\zeta)(1-P_2^Y),$$

hence

$$\zeta = \frac{1 - P_2^X - P_2^Y}{P_1^X - P_2^X + P_1^Y - P_2^Y}.$$

We can now define our probabilistic algorithm. Let A be the following algorithm:

Run either subroutine (i) or subroutine (ii), the first one with probability ζ and the second one with probability $1-\zeta$:

(i) Query I_1. If $f(I_1) \in B_1$ decide that $f \in X$; otherwise, decide that $f \in Y$.

(ii) Query I_2. If $f(I_2) \in B_2$ decide that $f \in X$; otherwise, decide that $f \in Y$.

Now we consider $\tilde{A}$. $\tilde{A}$ is clearly nonadaptive, as well as A. What is its success probability? By definition, the probability of success for $f \in X$ by querying I_1 is $P_1^X = P_{I_1}^X(B_1)$ and is $P_2^X = P_{I_2}^X(B_2)$ by querying I_2. Thus the success probability for any black-box function in X is $\zeta P_1^X + (1-\zeta)P_2^X = \gamma$. Similarly the success probability for any function in Y is $\zeta(1-P_1^Y) + (1-\zeta)(1-P_2^Y) = \gamma$. We have shown that the success probability of the algorithm is equal to γ. □

3.3 Proof of the lower bound

We now turn our attention to the lower bound. Our proof is somewhat closer to the probabilistic lower bound of [1] than to [19] since we use a version of Yao's minimax principle [25] (see Lemma 1 below).

Lemma 1 (Yao's principle for nonadaptive algorithms) *Assume that the success probability of a nonadaptive algorithm A of query complexity T on any function $f \in X \cup Y$ is at least ε. Fix some arbitrary probability distribution on $X \cup Y$. Then there is a deterministic algorithm of query complexity T with average success probability at least ε on this probability distribution. Moreover, the queries I made by this algorithm are independent of the input black-box function $f \in X \cup Y$.*

Proof. Let $p : X \cup Y \to [0;1]$ be a probability distribution. Consider an array where column indices are the functions of $X \cup Y$. The width of a column indexed by f is its probability in $X \cup Y$. Row indices are of the form (ω, I), where I is a sequence of (distinct) queries of length T and $\omega \in \Omega$. The height of the row (ω, I) is the probability that the random bits are equal to ω and A queries I. The array is therefore of area 1. The cell at row (ω, I) and column f contains $O(\omega, f(I))$ if $f \in X$ and $1 - O(\omega, f(I))$ if $f \in Y$ — let us recall that O is the output function of the algorithm, defined in section 2. In other words, the cell contains a 1 if the algorithm's output is correct, and a 0 otherwise.

The success probability of algorithm A is at least ε for every function in $X \cup Y$ and so in each column the proportion (according to cell area) of 1's is at least ε. The sum of the areas of all cells in the array which contain a 1 must therefore be at least ε. Thus there must be a row (I, ω) such that the proportion of 1 in the row is at least ε. Let $B = \{J | O(\omega, J) = 1\}$. Our deterministic algorithm is the following: make the queries I on f, decide that $f \in X$ if $f(I) \in B$, and decide that $f \in Y$ otherwise. $\square$

We can now complete the lower bound proof. Let A be a probabilistic algorithm for $\mathscr{P}$, and ε its success probability. For $p \in [0,1]$, consider the following distribution on the set $X \cup Y$: $f \in X$ gets probability $p/|X|$ and $g \in Y$ gets probability $(1-p)/|Y|$. By Lemma 1 there is a tuple of queries I and a set of answers B (both depending on p) such that the deterministic algorithm querying I and deciding that $f \in X$ iff $f(I) \in B$ has success probability at least ε. By definition of the deterministic algorithm, its success probability is equal to $pP_I^X(B) + (1-p)(1 - P_I^Y(B))$. Thus $\varepsilon \leq \max_{I,B}\ pP_I^X(B) + (1-p)(1 - P_I^Y(B))$. Since this inequality holds true for every $p \in [0,1]$, we have

$$\varepsilon \leq \gamma = \min_{0 \leq p \leq 1} \max_{I,B}\ pP_I^X(B) + (1-p)(1 - P_I^Y(B)).$$

An alternative proof of Proposition 1 can be derived from a converse of Lemma 1. Details will be given in a future version of this paper.

4 When adaptivity does not help

In light of the results of section 3, it is natural to ask whether the restriction to nonadaptive algorithms is a severe one. This seems to be a hard problem in general. For instance, a famous conjecture, still open to this day, states that that any nontrivial monotone graph property is elusive (i.e., any deterministic algorithm checking this property must in the worst case query all of the $\binom{n}{2}$ entries of the adjacency matrix of the input graph). This conjecture is attributed to Karp by Rosenberg [21]. The nonadaptive deterministic query complexity of any nontrivial graph property is equal to $\binom{n}{2}$. Karp's conjecture therefore amounts to the statement that adaptivity does not help for deterministically checking monotone graph properties.

A related (nonmonotone) example, the recognition of a scorpion graph, shows that strong hypotheses are needed in Theorem 1. A graph G of order n is called a *scorpion graph* if it contains a vertex b of degree $n-2$, the only vertex not adjacent to b being of degree 1 and linked with a vertex u, itself of degree 2. There exists a deterministic algorithm recognizing scorpion graphs of size n, using at most $6n$

queries. For a proof of this result and a nice picture of what a scorpion graph looks like, see [9], chapter VIII, theorem 1.5. Now, fix a minimal scorpion graph G on n vertices, and an edge e of G. Let G' be the graph obtained from G by deleting e. Let X be the set of graphs on n vertices which are isomorphic to G, and Y the set of graphs which are isomorphic to G'. The problem of distinguishing graphs of X from graphs of Y is transitively symmetric in the sense of Definition 2. Its nonadaptive deterministic query complexity is equal to $\binom{n}{2}$, but as explained above it can be solved by a deterministic algorithm of query complexity at most $6n$ (in fact, there is a significantly simpler linear time algorithm for this promise problem than for the total problem of scorpion graph recognition). Distinguishing star graphs from star graphs deprived from an edge provides yet another example of a transitively symmetric problem of query complexity $O(n)$ but of nonadaptive query complexity $\binom{n}{2}$[1]. These observations show that adaptivity can help for some transitively symmetric problems, and that the nonadaptivity hypothesis therefore can't be removed from Theorem 1. Note that the extent to which adaptivity can help has also been studied in the related framework of *graph property testing*, where one must accept (in the one-sided error case) all graphs that satisfy a given property, and reject with high probability all graphs that are "far" from satisfying this property [8, 14].

In this section we identify two simple classes of problems for which adaptivity does not help: the class of problems that are invariant under an arbitrary permutation on the domain of the black-box function, and the class of collision problems that are invariant under an arbitrary permutation on the range of the black-box function.

4.1 Invariance under permutations on the domain

We have already defined, for an algorithm A, its symmetrized version $\tilde{A}$. In this subsection we give a symmetry condition on $\mathscr{P}$ which implies that $\tilde{A}$ is nonadaptive, whatever A.

We denote by $P_A(f, I)$ the probability that algorithm A makes queries I when f is the black-box function. It is an immediate consequence of the definition that an algorithm A of query complexity T can be written in a nonadaptive way if and only if, for every I of size at most T, $P_A(f, I)$ does not depend on f.

Fact 1 *Suppose that there exists $H \leq \mathfrak{S}_{[N]}$ such that $H \times \{0\} \leq Aut(\mathscr{P})$ and H acts k-transitively on $[N]$. Then for every $f \in X \cup Y$ and every I of size at most k, $P_{\tilde{A}}(f, I) = \frac{1}{N(N-1)\cdots(N-|I|+1)}$.*

Recall that H acts k-transitively on $[N]$ if given two tuples $(a_1, \ldots, a_k)$ and $(b_1, \ldots, b_k)$, each made up of distinct elements of $[N]$, there exists $\sigma \in H$ such that $\sigma(a_i) = b_i$ for all $i = 1, \ldots, k$.

Proof. Let us fix the value of the random bits ω, and thus assume for now that A is a deterministic algorithm. As explained in section 2, we also assume that A never makes the same query twice.

[1] A star graph is a graph with $n-1$ edges which connect $n-1$ "external" vertices to one "central" vertex. The proof of the $O(n)$ upper bound on the query complexity of this problem is simple and left to the reader.

The first query is always the same, i_1. The actual first query made by $\tilde{A}$ is $\sigma^{-1}(i_1)$, where (σ, τ) is chosen at random in $\mathrm{Aut}(\mathscr{P})$. If $k \geq 1$, $\mathrm{Aut}(\mathscr{P})$ act transitively on $[N]$, so $\sigma^{-1}(i_1)$ is uniformly distributed on $[N]$, so that $P_{\tilde{A}}(f, I) = \frac{1}{N}$ when $|I| = 1$.

After the first query, A won't ask for i_1 a second time, so we can see the remaining part of the algorithm as an auxiliary algorithm that works on $\mathscr{P}$ as if the functions were only defined on $[N] \setminus \{i_1\}$. To put it in another way, it would not change anything if, before the second query, we were to replace (σ, τ) by a new random (σ', τ) such that $\sigma'(i_1) = \sigma(i_1)$. So, actually, each query is chosen at random among the set of queries not yet made. This proves the result for deterministic algorithms. An arbitrary randomized algorithm just consists of randomly picking a deterministic algorithm, so the result remains true in the general case. □

Unfortunately, it is a well known fact — see for instance [10], section 1.12 — that if a permutation group acts 7-transitively, then it acts $n-2$-transitively. This lemma will then only be useful when H happens to be the full symmetric group $\mathfrak{S}_{[N]}$ or the alternating group $\mathfrak{A}_{[N]}$. For instance, we have the following corollary for $\mathfrak{S}_{[N]}$ (it also follows immediately from Lemma 9 of [6]).

Corollary 1 *Let $\mathscr{P}$ be a problem such that $\mathfrak{S}_{[N]} \times \{0\} \leq \mathrm{Aut}(\mathscr{P})$. Then nonadaptive algorithms for $\mathscr{P}$ are no weaker than adaptive algorithms.*

Proof. Let A be a probabilistic algorithm for $\mathscr{P}$. Algorithm $\tilde{A}$ is nonadaptive by Fact 1, and its success probability is at least equal to that of A. □

It could seem at first sight that the hypotheses for Fact 1 are too strong, that it is enough to suppose that $\mathrm{Aut}(\mathscr{P})$ acts k-transitively on $[N]$ — the action being defined by $(\sigma, \tau).i = \sigma(i)$. However, this may not be the case if there is some kind of entanglement between the σ's and the τ's. For instance, consider the following problem: $N = M$, X contains only one function, the identity, and Y contains all transpositions. The groups of automorphisms of this problem is the diagonal group, $\{(\sigma, \sigma)/\sigma \in \mathfrak{S}_{[N]}\}$. Now let A be the following algorithm (we give only the beginning of the sequence of queries, as for now this is the only relevant thing):

- Query $i = 1, 2, 3, \ldots$ until finding i such that $f(i) \neq i$.
- When that happens, let the following query be $f(i)$.

This is what $\tilde{A}$ looks like:

- Query random distinct i's until finding i such that $f(i) \neq i$.
- When that happens, let the following query be $f(i)$.

The conclusion of Fact 1 is thus false in this case. For instance, let f be the (12) transposition; then $P_{\tilde{A}}(f, (1, 3)) = 0$.

4.2 Invariance under permutations on the range

In this subsection we show that adaptivity does not help for a certain class of collision problems.

Definition 4 *A problem $\mathscr{P}$ is a* collision problem *if every function in X, but no function in Y, is one-to-one — or vice-versa.*

The idea is that a collision problem is solved by answering the question: "is the black-box function one-to-one?". Like in section 4.1 our argument relies on the symmetrized algorithm $\tilde{A}$, but in contrast with that section we will have to modify $\tilde{A}$. Namely, whenever $\tilde{A}$ finds a collision, we will fool it by answering its queries with distinct elements randomly known from the range $[M]$ of the black-box function.

Fact 2 *If $\mathscr{P}$ is a collision problem such that $\{0\} \times \mathfrak{S}_{[M]} \leq \mathrm{Aut}(\mathscr{P})$, then non-adaptive algorithms for $\mathscr{P}$ are no weaker than adaptive algorithms.*

Proof. Let $\mathscr{P}$ be such a problem, let A be an algorithm for $\mathscr{P}$ of query complexity T and let $\tilde{A}$ be the symmetrized algorithm. Assume for instance that X contains only one-to-one functions; Y therefore contains none of them. Since $\mathrm{Aut}(\mathscr{P})$ contains $\{0\} \times \mathfrak{S}_{[M]}$, X actually contains *all* one-to-one functions from $[N]$ to $[M]$.

Suppose that $f \in X$. Let us denote by J the sequence of answers to the queries of $\tilde{A}$ when the black-box function is f. Since $\{0\} \times \mathfrak{S}_{[M]} \leq \mathrm{Aut}(\mathscr{P})$, in $\tilde{A}$, after each query $\sigma^{-1}(i)$, $\tau f \sigma^{-1}(i)$ is uniformly distributed among the elements of $[M]$ which are not results of previous queries. This means that J is uniformly distributed among the sequences of size T of distinct elements of $[M]$.

We now define an algorithm A' by modifying $\tilde{A}$ in the following way: while a collision is not found, we apply $\tilde{A}$ respectfully. But, as soon as a collision is found we lure $\tilde{A}$ into thinking that there is no collision until after the last query, when we take control of the output of $\tilde{A}$ in its place and declare that the black-box function has a collision. To make $\tilde{A}$ believe that it deals with a one-to-one function, we just answer its queries with random distinct elements of $[M]$ which are not the result of previous queries. This ensures that until the last query, $\tilde{A}$ cannot tell the difference between the functions of $X \cup Y$. Indeed, whatever f, the answers to the queries of $\tilde{A}$ are distinct elements drawn uniformly at random from $[M]$. In particular, $P_{A'}(f, I)$ is independent of f, so that A' can be written in a nonadaptive way. Moreover, the success probability of A' is at least equal to that of $\tilde{A}$, and the success probability of $\tilde{A}$ is at least equal to that of A. $\square$

5 Examples

5.1 Unordered Search

Let us first return to the problem $\mathscr{P}_{\text{search}}$ from section 2. As noticed earlier, this problem is transitively symmetric; moreover, its symmetry group is $\mathfrak{S}_{[N]} \times \{0\}$, so we can use Fact 1 and Theorem 1 to study it. First let us take a closer look at algorithms of query complexity 1. For every pair (I, B) there is a line $\Delta_{I,B}$ of equation $e = pP_I^X(B) + (1-p)(1 - P_I^Y(B))$ in the (p, e) plane. In fact, for any query we obtain the same four lines, depicted in Figure 2:

- $B = \emptyset$: $e = 1 - p$.

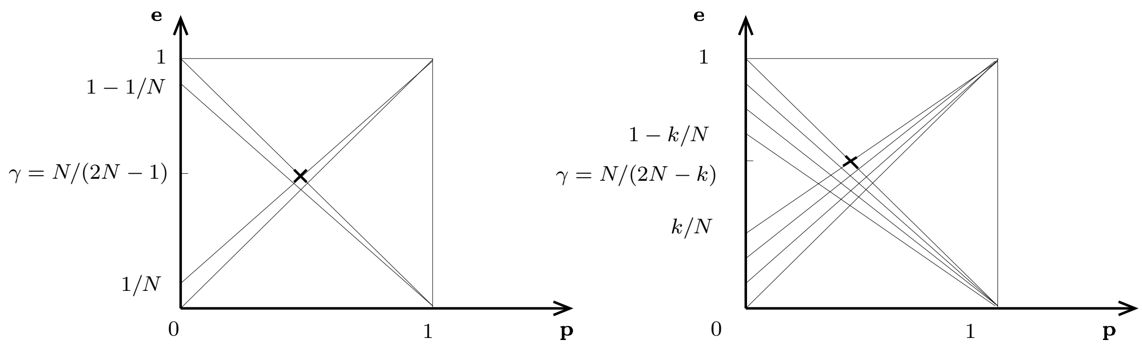

Figure 2: one query and k queries for the search problem

- $B = \{0\}$: $e = p(1-1/N) + 1/N$.
- $B = \{1\}$: $e = (1-p)(1-1/N)$.
- $B = \{0;1\}$: $e = p$.

Note the symmetry of this figure relatively the line of equation $e = 1/2$: the lines $\Delta_{I,B}$ and $\Delta_{I,\mathcal{P}([N])\setminus B}$ are symmetric.

We find that $\gamma = N/(2N-1)$ so the second algorithm from section 2 was optimal.

Suppose now that we allow k queries ($k \leq N$). By Theorem 1, the best success probability that can be achieved is $\gamma = N/(2N-k)$.

Usually, we are given a success probability $\varepsilon > 1/2$ and we want to compute the minimal number k of queries needed to have a probabilistic algorithm with k queries and success probability ε. In our example this value can be computed exactly:

$$k = \left\lceil N\left(\frac{2\varepsilon - 1}{\varepsilon}\right)\right\rceil$$

5.2 Simon's problem

For Simon's problem, $N = M = 2^n$. However, it is not so convenient to see the black-box functions as mere functions from $[N]$ to $[M]$. Instead, we will look at them as functions from the additive group $(\mathbb{Z}/2\mathbb{Z})^n$ to itself. We then define Y as the set of one-to-one functions on $(\mathbb{Z}/2\mathbb{Z})^n$, and X as the set of functions f such that there exists $s_f \in (\mathbb{Z}/2\mathbb{Z})^n \setminus \{0\}$ such that for all $x, y \in (\mathbb{Z}/2\mathbb{Z})^n$, $f(x) = f(y)$ if and only if $x = y + s_f$. Such an s_f is unique, hence the notation. This problem is transitively symmetric: if H is the group of all linear automorphisms of $(\mathbb{Z}/2\mathbb{Z})^n$ and H' the group of all permutations of $(\mathbb{Z}/2\mathbb{Z})^n$, then $H \times H'$ is suitable. Moreover, since it is a collision problem and its symmetry group contains $\{0\} \times \mathfrak{S}_{[N]}$, we need only consider nonadaptive algorithms by Fact 2.

Simon's problem has received a great deal of attention in the quantum computing literature because it was one of the first problems for which an exponential speedup over classical computation was exhibited. To demonstrate this speedup, Simon [22, 23] gave an efficient quantum algorithm for his problem and proved an $\Omega\left(\sqrt{2^n}\right)$ lower bound in a probabilistic model of computation (his result is stated in a slightly different manner, but the $\Omega\left(\sqrt{2^n}\right)$ lower bound does follow for algorithms with a bounded probability of error). However, Simon's lower bound was established only for the search version of his problem, that, is for the problem of

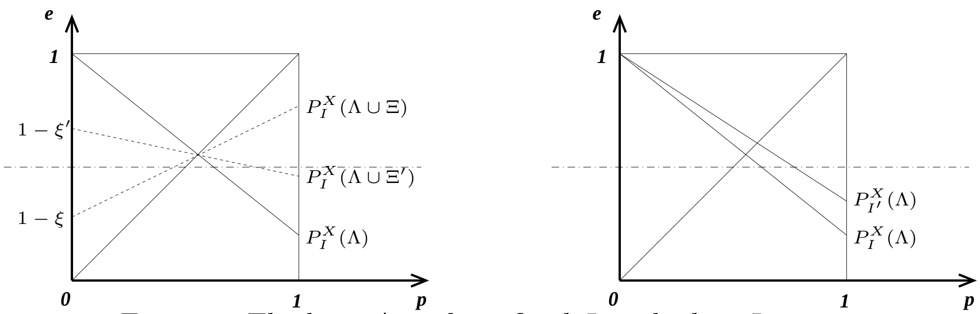

Figure 3: The lines $\Delta_{I,B}$ for a fixed I, and when I varies

finding a collision (that is, two distinct elements with the same image by f) in the case $f \in X$. The decision problem is a priori easier, and the corresponding lower bound more difficult. A proof sketch of a probabilistic lower bound for this decision problem can be found in the lecture notes of a recent course on quantum computing[2].

Let I be a sequence of T distinct queries, and J a sequence of T answers. Of course, if J takes the same value twice or more, then $P_I^Y(J)$ is 0. So, when trying to evaluate γ, we only need to consider those B that include all the non-injective sequences, the set of which we will call Λ. The point is that when two queries are given the same answer, this is no time to quibble: the black-box function is in X for sure.

Let us compute $P_I^X(\Lambda)$. A function $f \in X$ is one-to-one on I if and only if for all $x, y \in I$, $s_f \neq x - y$. When picking a random function f in X with uniform probability, s_f is uniformly distributed among $(\mathbb{Z}/2\mathbb{Z})^n \setminus \{0\}$; so $P_I^X(\Lambda) = \frac{|I-I|-1}{2^n-1}$, where $I - I$ denotes the Minkowski difference of I and itself, i.e. $I - I = \{x - y / x, y \in I\}$.

Let I be fixed. Note that, for a fixed I, neither $P_I^X(J)$ nor $P_I^Y(J)$ depend on J when J is a sequence of distinct elements (for instance, $P_I^Y(J)$ is equal to $\frac{(N-T)!}{N!}$ since $N!$ is the total number of one-to-one functions and $(N-T)!$ the number of the those taking the values J on I). When B contains only one-to-one J's, $P_I^X(B)$ and $P_I^Y(B)$ are therefore linear functions of the size of B. Taking into account the fact that we only need to consider those B that contain Λ, this remark implies that (for our fixed set I of T queries) all the relevant lines $\Delta_{I,B}$ go through the same point. As shown in Figure 3, this point is in particular at the intersection of the lines corresponding to $B = \Lambda$ and $B = [M]^T$ (in that picture, Ξ and Ξ' are sets of one-to-one functions respectively containing a proportion ξ and ξ' of the set of all one-to-one functions).

From these observations we obtain that, for a fixed I,

$$\min_{0 \leq p \leq 1} \max_A p P_I^X(A) + (1-p)\left(1 - P_I^Y(A)\right) = \frac{1}{2 - P_I^X(\Lambda)} = \frac{1}{2 - \frac{|I-I|-1}{2^n-1}}$$

It should also be clear from Figure 3 that, since the minmaxes all take place on the line of equation $y = x$, we can switch $\min\limits_{0 \leq p \leq 1}$ and $\max\limits_I$ in the definition of γ so

[2] `http://www.cwi.nl/themes/ins4/qc2005/yaoprin.pdf`

that

$$\gamma = \max_I \frac{1}{2 - \frac{|I-I|-1}{2^n-1}}$$

The best algorithm using T queries then consists in choosing an I of size T maximizing $|I - I|$, querying f on I and then applying one of the following two subroutines, the first one with probability $1-\gamma$ and the second one with probability γ:

- Discard the answers and claim that f is in X.
- Claim that f is in X if and only if the same answer has been returned twice.

If we want the algorithm to be successful with probability at least ε for every black-box function, we need to find an I such that $|I - I| \geq (2^n - 1)\left(2 - \frac{1}{\varepsilon}\right) + 1$. Computing exactly the size of the smallest such set seems to be a difficult combinatorial question. However, it is easily proven that for a fixed ε, the size of this set (and therefore the query complexity of our optimal algorithm) is $\Theta\left(\sqrt{2^n}\right)$.

5.3 One-to-one versus two-to-one functions

For this problem we have $N = M = 2K$. While Y is still the set of one-to-one functions, X is now the set of all two-to-one functions, so that there is no longer any algebraic structure. Quantum lower bounds for this are established in [2, 5, 17]. This problem is transitively acted upon by $\mathfrak{S}_{2K} \times \mathfrak{S}_{2K}$. And once more, according to Fact 2, we need only consider nonadaptive algorithms.

As for Simon's problem, the sequences J containing at least twice the same element play a distinctive role, since knowing that f takes twice the same value is a mean — and the only mean, provided that T is small enough — to know for sure that f is in X; so we will likewise name Λ the set of those sequences.

Again, let us compute $P_I^X(\Lambda)$. This is the probability that a random two-to-one function f be not injective on a fixed set I. Dually, this is also the probability for the restriction of a fixed two-to-one function f on a random set I of a given size to be non injective. There are $\binom{2K}{T}$ sets of size T. Now, to count the number of sets I such that the restriction of f on I is one-to-one, consider the domain of f divided into a partition of two-element sets on which f is constant. First, choose T of those sets: there are $\binom{K}{T}$ possibilities. Then for each two-element set choose which element you keep: there are 2^T possibilities. Hence $P_I^X(\Lambda) = 1 - 2^{|I|}\frac{\binom{K}{|I|}}{\binom{2K}{|I|}}$.

In the same way as for Simon's problem, for a fixed I neither $P_I^X(J)$ nor $P_I^Y(J)$ depend on J when J is made up of distinct elements. It follows that

$$\gamma = \max_{|I|=T} \frac{1}{2 - P_I^X(\Lambda)} = \frac{1}{1 + 2^T \frac{\binom{K}{T}}{\binom{2K}{T}}}.$$

From this formula it can be inferred that, in order for an algorithm to solve this problem with bounded probability of error, the optimal number of queries is $\Theta(\sqrt{N})$.

5.4 Hidden Translation

This problem is studied from a quantum point of view in [12]. We set $M = N = 2K$ and Y is once more the set of one-to-one functions. In a similar manner as for Simon's problem, we will look at our functions as functions from the set $\{0;1\} \times \mathbb{Z}/K\mathbb{Z}$ to itself. We define Y as the set of one-to-one functions, and X as the set of functions f that are one-to-one on $\{0\} \times \mathbb{Z}/K\mathbb{Z}$ and such that there exists an element $s_f \in \mathbb{Z}/K\mathbb{Z}$ such that for all $x \in \mathbb{Z}/K\mathbb{Z}$ we have $f(1,x) = f(0, x+s_f)$. This s_f is the eponymous ***hidden translation*** of f, since on one half on its domain f is deduced from its values on the other half by a translation of parameter s_f. Let us denote by $\sigma_0, \dots, \sigma_{k-1}$ the elements of the additive group $\mathbb{Z}/K\mathbb{Z}$. We can make H act on $\{0;1\} \times \mathbb{Z}/K\mathbb{Z}$ as follows: $\sigma_i(0,j) = (0, i+j)$ and $\sigma_i(1,j) = (1,j)$. Our problem is transitively acted upon by $H \times \mathfrak{S}_{2K}$ and, by Fact 2, nonadaptive algorithms are optimal.

As expected, we can still define Λ in the same way as in the previous two examples, since there is no one-to-one function in X. For $I \subseteq \{0;1\} \times \mathbb{Z}/K\mathbb{Z}$, let us define two subsets of $\mathbb{Z}/K\mathbb{Z}$, I_0 and I_1, such that $I = (\{0\} \times I_0) \cup (\{1\} \times I_1)$.

A function $f \in X$ is non-injective on I if and only if there are $x \in I_0$ and $y \in I_1$ such that $f(x) = f(y+s_f)$. But, when f is uniformly distributed on X, s_f is uniformly distributed in $\mathbb{Z}/K\mathbb{Z}$; hence $P_I^X(\Lambda) = \frac{|I_1 - I_0|}{K}$. Once again, for a fixed I, neither $P_I^X(J)$ nor $P_I^Y(J)$ depends on J when J is made up of distinct elements. So, for a fixed number T of queries, we have

$$\gamma = \max_{|I|=T} \frac{1}{2 - P_I^X(\Lambda)} = \max_{|I|=T} \frac{1}{2 - \frac{|I_1 - I_0|}{K}} = \frac{1}{2 - \frac{\max_{|I|=T} |I_1 - I_0|}{K}}.$$

We again encounter a nontrivial combinatorial problem: given T, maximizing $A - B$ for $A, B \subseteq \mathbb{Z}/K\mathbb{Z}$, under the condition that $|A| + |B| = T$. Nevertheless, it is easy to prove from this expression for γ that the optimal number of queries is $\Theta(\sqrt{N})$.

5.5 Completely symmetric problems

For a black box $f : [N] \to [M]$, let $n_{f,i}$ denote the number of x's such that $f(x) = i$. A completely symmetric problem is a black-box problem $\mathscr{P}$ such that $\mathrm{Aut}(\mathscr{P}) \supseteq \mathfrak{S}_N \times \{\mathrm{Id}\}$, i.e., such that $\mathscr{P}(f)$ depends only on the numbers $n_{f,i}$. As pointed out already in the introduction, this is the usual notion of a symmetric problem. According to Fact 2, for completely symmetric problems, nonadaptive algorithms are optimal. However, as soon as $M \geq 2$ and $N \geq 4$, a problem which is total and completely symmetric cannot be transitively symmetric[3]. These problems can nevertheless be made to fit our framework through a reduction. Indeed, let $\mathscr{P}$ be a problem which is total and completely symmetric, and suppose moreover that $\mathscr{P}$ is nontrivial in the sense that it has at least one positive instance and one negative instance. Then there exist black boxes f and g and j_1, j_2 in $[M]$ such that $\mathscr{P}(f) \neq \mathscr{P}(g)$, $n_{g,j_1} = n_{f,j_1} + 1$, $n_{g,j_2} = n_{f,j_2} - 1$ and $n_{g,i} = n_{f,i}$ for all $i \neq j_1, j_2$.

---

[3] This follows from the fact that for $M = 2$, an automorphism (σ, τ) either preserves $n_{f,0}$ (in the case $\tau = \mathrm{Id}$) or replaces it by $N - n_{f,0}$. For $N \geq 4$ there are at least 5 possible values for $n_{f,0}$, so one of the two classes must contain black boxes f with 3 different values for $n_{f,0}$.

To find a lower bound for $\mathscr{P}$, we can restrict $\mathscr{P}$ to the easier problem $\mathscr{P}'$ equal to $\mathscr{P}$ except that it is defined only for the functions h such that $n_{h,i} = n_{f,i}$ for all i, or $n_{h,i} = n_{g,i}$ for all i. This is a generalization of the unordered search problem studied in Section 5.1 (take $M = 2$, $j_1 = 0$, $j_2 = 1$, $n_{f,j_1} = N-1$, $n_{f,j_2} = 1$, $n_{g,j_1} = N$, $n_{g,j_2} = 0$). The promise problem $\mathscr{P}'$ is transitively symmetric, and could therefore be handled with Theorem 1. It is however more convenient to make use of the Hellinger distance (see Section 6.3). If $r = n_{f,j_1}$ and $s = n_{f,j_2}$, the parameter $h_0(\mathscr{P}')$ defined in that section is equal to

$$h = \sqrt{\frac{1}{N}\left(r + s - \sqrt{r(r+1)} - \sqrt{s(s-1)}\right)}.$$

Since $h \leq \sqrt{\frac{1}{N}}$, an $\Omega(N)$ lower bound follows from equation (1) of Section 6.3.

6 Connection to the variation distance, to block sensitivity and to the Hellinger distance

6.1 The variation distance

In this section we define for every black-box problem $\mathscr{P}$ a new parameter α which is closely related to the parameter γ of Theorem 1. Its definition is somewhat simpler than that of γ, and is based on the variation distance (a notion which was also used in [6]).

Let $\mathscr{P}$ be a black-box problem from $[M]^{[N]}$ to $\{0,1\}$. Recall that if T is an integer, $I \in [N]^T$ a list of queries and $B \subseteq [M]^T$ a set of possible answers, we denote by $P_I^X(B)$ the proportion of functions $f \in X$ satisfying the condition $f(I) \in B$; and that we denote by $P_I^Y(B)$ the proportion of functions $g \in Y$ satisfying the condition $g(I) \in B$.

Let α_I be the variation distance between the probability distributions P_I^X and P_I^Y. It is by definition equal to the supremum over B of $|P_I^X(B) - P_I^Y(B)|$. It is a well known (and elementary) fact that the variation distance is equal to one half of the L_1-distance. We therefore have:

$$\alpha_I = \max_{B\subseteq[M]^T} |P_I^X(B) - P_I^Y(B)| = \frac{1}{2} d_{L_1}(P_I^X, P_I^Y) = \frac{1}{2} \sum_{J\in[M]^T} |P_I^X(J) - P_I^Y(J)|.$$

The parameter α is defined as the maximum of the α_I's:

$$\alpha = \max_{I\in[N]^T} \alpha_I = \max_{I\in[N]^T, B\subseteq[M]^T} |P_I^X(B) - P_I^Y(B)|.$$

We now show that this new parameter provides a reasonably good approximation of γ.

Proposition 2 *For every black-box problem $\mathscr{P}$ we have $\frac{1}{2-\alpha} \leq \gamma \leq \frac{1}{2} + \frac{\alpha}{2}$.*

Proof. Let us consider the convex function

$$\begin{array}{rcl} \mathscr{E} : [0;1] & \to & \left[\frac{1}{2};1\right] \\ p & \mapsto & \max_{I\in[N]^T, B\subseteq[M]^T} \; pP_I^X(B) + (1-p)(1 - P_I^Y(B)) \end{array}.$$

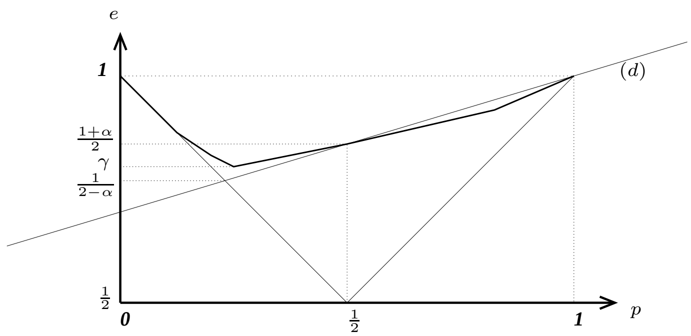

Figure 4: γ and the variation distance

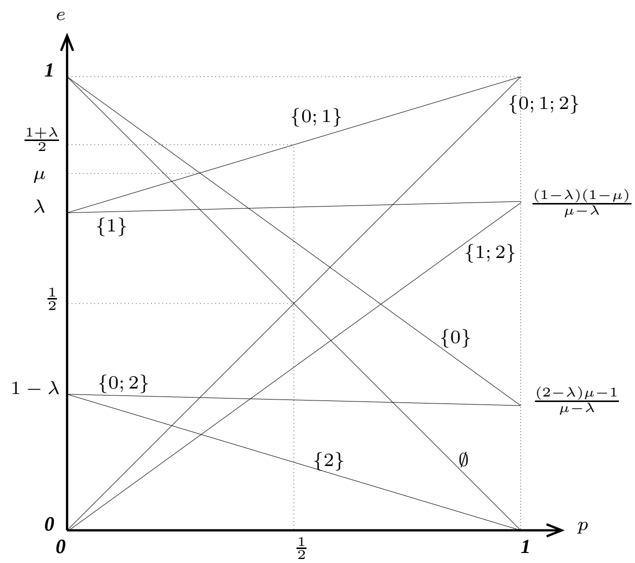

Figure 5: Optimality of Proposition 2

Almost by definition, $\mathscr{E}\left(\frac{1}{2}\right) = \frac{1+\alpha}{2}$. Hence $\gamma \leq \frac{1}{2} + \frac{\alpha}{2}$ since $\gamma = \min \mathscr{E}$. Suppose that this minimum of $\mathscr{E}$ is attained for $p \in \left[0; \frac{1}{2}\right]$ — the other case is of course quite similar. Taking $B = \emptyset$ shows that $\mathscr{E}(p) \geq 1 - p$. Furthermore, the convexity of $\mathscr{E}$ gives another relation, as can be seen on Figure 4. Namely, for $p \in \left[0; \frac{1}{2}\right]$, the graph of $\mathscr{E}$ must be above the line (d) passing through the points $\left(\frac{1}{2}; \frac{1+\alpha}{2}\right)$ and $(1;1)$. The equation of (d) being $e = (1-\alpha)p + \alpha$, the two minorations of $\mathscr{E}$ on $\left[0; \frac{1}{2}\right]$ imply that $\gamma \geq \frac{1}{2-\alpha}$. $\square$

These inequalities are optimal. Indeed, given $\lambda \in [0;1) \cap \mathbb{Q}$ and $\mu \in \left[\frac{1}{2}; 1\right) \cap \mathbb{Q}$ such that $\frac{1}{2-\lambda} \leq \mu \leq \frac{1+\lambda}{2}$, one can consider the following problem. Take $M = 3$ and N such that λN and $\frac{(2-\lambda)\mu-1}{\mu-\lambda} N$ are both integral. Put in X all the functions which take $\frac{(2-\lambda)\mu-1}{\mu-\lambda} N$ times the value 0 and, which never take the value 2. Put in Y all the functions which take the value 2 exactly λN times, and which never take the value 0. A careful study of this problem for $T = 1$ (i.e., for algorithms making only one query) shows that $\alpha = \lambda$ and $\gamma = \mu$. Here is a quick justification. First, note that the constraints on λ and μ ensure that $0 \leq \frac{(2-\lambda)\mu-1}{\mu-\lambda} \leq 1$. Then, it is a bit tedious but straightforward to check that the graph of $\mathscr{E}$ is made of two line segments: the first one corresponds to the question "is $f(i)$ equal to 0?", and the second one to the question "is $f(i)$ in $\{0;1\}$?". The minimum of the convex function is achieved at the intersection of these two line segments, and its value there is indeed equal to μ. For $p = \frac{1}{2}$ the higher line is the one asking if $f(i)$ is in $\{0;1\}$, and its value there is λ. Probably the best way to understand that is to stare at Figure 5 for a sufficiently long time. For each $B \subseteq \{0;1;2\}$, the line of equation $e = pP^X_{\{i\}}(B) + (1-p)P^Y_{\{i\}}(B)$ is drawn and tagged with B, so that there are eight lines corresponding to the eight subsets of $\{0;1;2\}$.

6.2 Block sensitivity

In the next two subsections we will discuss the relations between our result and the results of [6]. In particular, we show that the methods of that paper do not yield any nonconstant lower bound for the problem of section 5.3 (distinguishing one-to-one functions from two-to-one functions).

The notion of block sensitivity was defined by Nisan [20] for Boolean functions and generalized in [6] for general finite domains and range.

First we give the definitions of "approximation" and "disjoint inputs", which are simplified versions of the definitions in [6]. Let $\mathscr{P}$ be a function from a subset Z of $[M]^{[N]}$ to $[L]$, where N, M and L are positive integers. By definition, the black-box functions in Z are said to respect the promise.

Definition 5 *An approximation for $\mathscr{P}$ is a function $C : Z \to 2^{[L]}$. Two black-box functions $f, g \in Z$ are said to be C-disjoint if $C(f) \cap C(g) = \emptyset$.*

The trivial approximation function is $C(f) = \{\mathscr{P}(f)\}$ for every f and more generally, we choose for every $f \in Z$ a set $C(f)$ that contains $\mathscr{P}(f)$.

Now we adapt the definition of block sensitivity to a partial function. In the following definition we denote by $f^{(I \leftarrow Q)}$ the black-box function obtained from f by changing the images $f(I)$ of the elements in $I \subseteq [N]$ to the values $Q \in [M]^{|I|}$. The function $f^{(I \leftarrow Q)}$ is not necessarily respecting the promise.

Definition 6 *$\mathscr{P}$ is C-sensitive to a subset of variables $I \subseteq [N]$ on function $f \in Z$ if there exists $Q \in [M]^{|I|}$ such that f and $f^{(I \leftarrow Q)}$ are C-disjoint and $f^{(I \leftarrow Q)} \in Z$. The C-block sensitivity of $\mathscr{P}$ on f, $bs_C(\mathscr{P}, f)$, is the maximum number t of pairwise disjoint subsets $I_1, \ldots, I_t \subseteq [N]$, such that $\mathscr{P}$ is C-sensitive to each of them on f. The C-block sensitivity of $\mathscr{P}$, $bs_C(\mathscr{P})$, is the maximum of $bs_C(\mathscr{P}, f)$ over all $f \in Z$.*

The block sensitivity gives a lower bound for the query complexity.

Definition 7 *An algorithm is said to (C, ε)-approximate $\mathscr{P}$ if for every black-box $f \in Z$ the probability that the output of the algorithm belongs to $C(f)$ is at least ε. The worst-case query complexity of the algorithm is then the maximum number of queries of the algorithm on a black-box function $f \in Z$. The (C, ε) worst-case query complexity of $\mathscr{P}$ is the minimum worst-case query complexity of an algorithm (C, ε)-approximating $\mathscr{P}$; we denote it $S^w_{C,\varepsilon}(\mathscr{P})$.*

If C is the trivial approximation then $S^w_{C,\varepsilon}(\mathscr{P})$ is the query complexity of the best probabilistic algorithm solving $\mathscr{P}$ with probability of success at least ε. The next theorem, as the above definitions, is relative to some approximation C.

Theorem 2 *For every $1 \geq \varepsilon \geq 1/2$, $\mathscr{P} : Z \subseteq [M]^{[N]} \mapsto [L]$ and approximation C of $\mathscr{P}$ we have:*

$$S^w_{C,\varepsilon}(\mathscr{P}) \geq (2\varepsilon - 1)\ bs_C(\mathscr{P}).$$

In [6] this theorem seems to be stated for total functions only (i.e., for the case $Z = [M]^{[N]}$), probably because the examples that the authors have in mind (such as approximating the median, the mean, and higher statistical moments) are total. In fact, this theorem is still true for partial functions, and can be proven with essentially the same proof as in the total case. Here we prove it by extending $\mathscr{P}$. Let $\mathscr{P}'$ be any total extension of $\mathscr{P}$. We define an approximation C' for $\mathscr{P}'$: if $f \in Z$ then $C'(f) = C(f)$ and if $f \notin Z$ then $C'(f) = 2^{[L]}$.

The first remark is that if $f \notin Z$ and $g \in [M]^{[N]}$ then f and g are not C'-disjoint. Thus $bs_{C'}(\mathscr{P}') = bs_C(\mathscr{P})$. Moreover an algorithm (C, ε)-approximates $\mathscr{P}$ iff it (C', ε)-approximates $\mathscr{P}'$. So $S^w_{C,\varepsilon}(\mathscr{P}) = S^w_{C',\varepsilon}(\mathscr{P}')$ and the theorem is proven.

For the following applications we work with the trivial approximation function, we take $L = 2$ and as in the remainder of the paper we call X the set of function $f \in [M]^{[N]}$ such that $\mathscr{P}(f) = 1$ and Y the set of function $g \in [M]^{[N]}$ such that $\mathscr{P}(g) = 0$. Therefore, $Z = X \cup Y$.

First we consider the problem $\mathscr{P}_{\text{search}}$ from section 2. The block-sensitivity of the constant zero function is N and the block-sensitivity of the others functions are 1. Thus the block sensitivity of $\mathscr{P}_{\text{search}}$ is N and we get the lower bound $N(2\varepsilon - 1)$ which is equal up to a factor ε to the one of section 5.1.

For the one-to-one versus two-to-one problem (see section 5.3), Simon's problem and the hidden translation problem which are subproblems of the one-to-one versus two-to-one problem, the block-sensitivity is 2 and so we obtain a constant lower bound. The lower bounds of section 5 are of course much higher.

6.3 The Hellinger distance

Suppose that problem $\mathscr{P}$ (a function from $[M]^{[N]}$ to $\{0; 1\}$) is *completely symmetric* in the sense that $\mathfrak{S}_N \times \{\mathrm{Id}\} \leq \mathrm{Aut}\,(\mathscr{P})$. As a consequence, $\mathscr{P}(f)$ depends only on the number of $x \in [N]$ such that $f(x) = y$, for each y in $[M]$. Then the theorem 8 of [6] gives a lower bound on $S^w_C(\mathscr{P}, \varepsilon)$. Recall that the Hellinger distance between two distributions P and Q on a finite set Ω is defined as follows:

$$h(P, Q) = \sqrt{1 - \sum_{\omega \in \Omega} \sqrt{P(\omega) Q(\omega)}}.$$

Each function f in $[M]^{[N]}$ induces a probability distribution P_f on $[M]$. Now, $h_C(\mathscr{P})$ for an approximation C of $\mathscr{P}$ is defined as the minimum of the Hellinger distances between P_f and P_g for f and g being C-disjoint. According to the theorem 8 of [6], the following lower bound holds provided that $\varepsilon > \frac{3}{4}$, $h_C(\mathscr{P}) \leq 1/2$, and $S^w_C(\mathscr{P}, \varepsilon) \leq N/4$:

$$S^w_{C,\varepsilon}(\mathscr{P}) \geq \frac{1}{4h^2_C(\mathscr{P})} \ln \frac{1}{4(1-\varepsilon)+O\left(\frac{1}{N}\right)}. \tag{1}$$

The parameter $h_C(\mathscr{P})$ is defined only in terms of the P_f's, and thus does not seem to take into account the possibly complex effects that can happen when the behaviors of black boxes in X and Y are compared on *several* inputs. This peculiarity may explain some of the limitations of lower bound (1). It also makes it fairly easy to compute $h_C(\mathscr{P})$ on specific examples.

Like the block sensitivity lower bound, lower bound (1) is stated in [6] for total problems only. Fortunately, the same technique as in the block sensitivity section shows that the theorem is also true for partial function. We define a total extension $\mathscr{P}'$ of $\mathscr{P}$ and the approximation C' for $\mathscr{P}'$ such that if $f \in Z$ then $C'(f) = C(f)$ and if $f \notin Z$ then $C'(f) = 2^{[L]}$. Then $S^w_{C,\varepsilon}(\mathscr{P}) = S^w_{C',\varepsilon}(\mathscr{P}')$, $h_C(\mathscr{P}) = h'_C(\mathscr{P}')$ and the lower bound (1) is true for $\mathscr{P}$.

In the following, C is the trivial approximation. As a first example, let us see what (1) says about $\mathscr{P}_{\text{search}}$, the unordered search problem studied in section 5.1. There is only one function in X, the zero function. The corresponding distribution P is given by

$$P(0) = 1; P(1) = 0.$$

In Y, we find the functions f_i such that $f_i(i) = 1$ and $f_i(j) = 0$ when $j \neq i$. The corresponding distributions Q_i are given by the formula

$$Q_i(0) = 1 - \frac{1}{N}; Q_i(1) = \frac{1}{N}.$$

The Hellinger distance between P and Q_i being $\sqrt{1-\sqrt{1-\frac{1}{N}}}$, we find an $\Omega(N)$ lower bound.

Our second example is the "one-to-one versus two-to-one" problem. Now X is made up of all one-to-one functions, so that the corresponding distribution P satisfies

$$P(j) = \frac{1}{2K} \text{ for all } j \text{ in } [2K].$$

For a function f in Y, the corresponding distribution Q_f is defined by

$$Q_f(j) = \begin{cases} \frac{1}{K} & \text{if } j \in \mathrm{im}(f); \\ 0 & \text{otherwise.} \end{cases}$$

The Hellinger distance between P and any Q_f is $\sqrt{1-\frac{1}{\sqrt{2}}}$, which does not depend on K, so that in this case (1) only proves a constant lower bound. This is only to be expected, since this lower bound method does not "see" that the images of black-box functions as well as the entries are permutable. As a result, this method cannot distinguish between the one-to-one versus two-to-one problem and a modified version of this problem where we are promised that the range of the two-to-one functions is some fixed subset of $[2K]$, say $[K]$. Obviously, this modified problem is a lot easier – in fact, its query complexity for a given allowed error $1-\varepsilon$ is constant.

Acknowledgments

Thanks go to Frédéric Magniez for pointing out references [3, 6] and for useful discussions.

References

[1] S. Aaronson. Lower bounds for local search by quantum arguments. In *Proc. STOC 2004*, pages 465–474. ACM, 2004.

[2] S. Aaronson and Y. Shi. Quantum Lower Bounds for the Collision and the Element Distinctness Problems. *Journal of the ACM*, 51(4):595–605, July 2004.

[3] A. Ambainis. A new quantum lower bound method, with an application to strong direct product theorem for quantum search. `www.arxiv.org/pdf/quant-ph/0508200`.

[4] A. Ambainis. Quantum lower bounds by quantum arguments. *J. Comput. Syst. Sci.*, 64(4):750–767, 2002.

[5] A. Ambainis. Polynomial degree and lower bounds in quantum complexity: Collision and element distinctness with small range. *Theory of Computing*, 1(3):37–46, 2005.

[6] Z. Bar-Yossef, R. Kumar, and D. Sivakumar. Sampling Algorithms: lower bounds and applications. In *Proc. STOC 2001*, pages 266–275. ACM, 2001.

[7] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001.

[8] A. Bogdanov and L. Trevisan. Lower bounds for testing bipartiteness in dense graphs. In *Proc. 19th IEEE Annual Conference on Computational Complexity (CCC'04)*. IEEE, 2004.

[9] B. Bollobás. *Extremal Graph Theory.* Dover Publications, Incorporated, 2004.

[10] P. J. Cameron. *Permutation groups*, volume 45 of *London Mathematical Society Student Texts.* Cambridge University Press, Cambridge, 1999.

[11] S. Fenner, F. Green, S. Homer, and Y. Zhang. Bounds on the power of constant-depth quantum circuits. In *Proc. 15th International Symposium on on Fundamentals of Computation Theory (FCT 2005)*, volume 3623 of *Lecture Notes in Computer Science*, pages 44–55. Springer, 2005.

[12] K. Friedl, G. Ivanyos, F. Magniez, M. Santha, and P. Sen. Hidden translation and orbit coset in quantum computing. In *STOC '03: Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, 2003.

[13] K. Friedl, G. Ivanyos, M. Santha, and Y. Verhoeven. On the black-box complexity of Sperner's lemma. In *Proc. 15th International Symposium on on Fundamentals of Computation Theory (FCT 2005)*, volume 3623 of *Lecture Notes in Computer Science*, pages 245–257. Springer, 2005.

[14] O. Goldreich and L. Trevisan. Three theorems regarding testing graph properties. *Random Structures and Algorithms*, 23(1):23–57, 2003.

[15] P. Koiran, V. Nesme, and N. Portier. A quantum lower bound for the query complexity of Simon's problem. In *Proc. ICALP 2005*, volume 3580 of *Lecture Notes in Computer Science*, pages 1287–1298. Springer, 2005.

[16] P. Koiran, V. Nesme, and N. Portier. The quantum query complexity of the abelian hidden subgroup problem. LIP Technical Report RR2005-17, Ecole Normale Supérieure de Lyon, 2005.

[17] S. Kutin. Quantum lower bound for the collision problem. *quant-ph/0304162*, 2003.

[18] S. Kutin. Quantum lower bound for the collision problem with small range. *Theory of Computing*, 1:29–36, 2005.

[19] S. Laplante and F. Magniez. Lower bounds for randomized and quantum query complexity using Kolmogorov arguments. In *Proc. 19th IEEE Annual Conference on Computational Complexity (CCC'04)*. IEEE, 2004.

[20] N. Nisan. CREW PRAMs and decision trees. *SIAM J. Comput.*, 20(6):999–1007, 1991.

[21] A. L. Rosenberg. On the time required to check properties of graphs: A problem. *SIGACT News*, pages 15–16, 1973.

[22] D. R. Simon. On the power of quantum computation. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 116–123, 1994.

[23] D. R. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483, 1997.

[24] R. Spalek and M. Szegedy. All quantum adversary methods are equivalent. In *Proc. ICALP 2005*, volume 3580 of *Lecture Notes in Computer Science*, pages 1299–1311. Springer, 2005.

[25] A. C. Yao. Probabilistic computations: toward a unified measure of complexity. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science*, pages 222-227, 1977.