Optimal parallel quantum query algorithms

Stacey Jeffery[1], Frederic Magniez[2], and Ronald de Wolf[3]

[1] IQC, University of Waterloo, Canada. sjeffery@uwaterloo.ca
[2] CNRS, LIAFA, Univ Paris Diderot, Sorbonne Paris-Cité, 75205 Paris, France. frederic.magniez@univ-paris-diderot.fr
[3] CWI and University of Amsterdam, The Netherlands. rdewolf@cwi.nl

Abstract. We study the complexity of quantum query algorithms that make p queries in parallel in each timestep. We show tight bounds for a number of problems, specifically $\Theta((n/p)^{2/3})$ p-parallel queries for element distinctness and $\Theta((n/p)^{k/(k+1)})$ for k-sum. Our upper bounds are obtained by parallelized quantum walk algorithms, and our lower bounds are based on a relatively small modification of the adversary lower bound method, combined with recent results of Belovs et al. on learning graphs. We also prove some general bounds, in particular that quantum and classical p-parallel complexity are polynomially related for all total functions f when p is small compared to f's block sensitivity.

1 Introduction

Using quantum effects to speed up computation has been a prominent research-topic for the past two decades. Most known quantum algorithms have been developed in the model of quantum query complexity, the quantum generalization of decision tree complexity. Here an algorithm is charged for each "query" to the input, while intermediate computation is free (see [15] for more details). This model facilitates the proof of lower bounds, and often, though not always, quantum query upper bounds carry over to quantum time complexity. For certain functions one can obtain large quantum-speedups in this model. For example, Grover's algorithm [21] can search an n-bit database (looking for a bit-position of a 1) using $O(\sqrt{n})$ queries. In contrast, any classical algorithm needs $\Omega(n)$ queries. For some partial functions we know exponential and even unbounded speed-ups [18, 34, 33, 7].

A more recent crop of quantum speed-ups come from algorithms based on *quantum walks*. Such algorithms solve a search problem by embedding the search on a graph, and doing a quantum walk on this graph that converges rapidly to a superposition over only the "marked" vertices, which are the ones containing a solution. An important example is Ambainis's quantum algorithm for solving the *element distinctness* problem [3]. In this problem one is given an input $x \in [q]^n$, and the goal is to find a pair of distinct i and j in $[n]$ such that $x_i = x_j$, or report that none exists. Ambainis's quantum walk solves this in $O(n^{2/3})$ queries, which is optimal [1]. Classically, $\Theta(n)$ queries are required. Two generalizations of this are the *k-distinctness* problem, where the objective is to

find distinct $i_1, \ldots, i_k \in [n]$ such that $x_{i_1} = \cdots = x_{i_k}$, and the *k-sum* problem, where the objective is to find distinct $i_1, \ldots, i_k \in [n]$ such that $x_{i_1} + \cdots + x_{i_k} = 0 \mod q$. Ambainis's approach solves both problems using $O(n^{k/(k+1)})$ quantum queries. Recently, Belovs gave a $o(n^{3/4})$-query algorithm for k-distinctness for any fixed k [8] (which can also be made time-efficient for $k = 3$ [11]). In contrast, Ambainis's $O(n^{k/(k+1)})$-query algorithm is optimal for k-sum [10, 14].

Here we consider to what extent such algorithms can be *parallelized*. Doing operations in parallel is a well-known way to trade hardware for time, speeding up computations by distributing the work over many processors that run in parallel. This is becoming ever more prominent in classical computing due to multi-core processors and grid computing. In the case of quantum computing there is an additional reason to consider parallelization, namely the limited lifetime of qubits due to *decoherence*: because of unintended interaction with their environment, qubits tend to lose their quantum properties over a limited amount of time, called the *decoherence time*, and degrade to classical random bits. One way to fight this is to apply quantum error-correction[4], which can counteract the effects of certain models of decoherence. Another way is to try to parallelize as much as possible, completing the computation before the qubits decohere too much (this may of course increase the width of the computation, creating other problems).

We know of only a few results about parallel quantum algorithms, most of them in the circuit model where "time" is measured by the depth of the circuit. A particularly important and beautiful example is the work of Cleve and Watrous [16], who showed how to implement the n-qubit quantum Fourier transform using a quantum circuit of depth $O(\log n)$. As a consequence, they were able to parallelize the quantum component of Shor's algorithm: they showed that one can factor n-bit integers by means of an $O(\log n)$-depth quantum circuit with polynomial-time classical pre- and post-processing. There have also been a number of papers about quantum versions of small-depth classical Boolean circuit classes like AC and NC [29, 19, 23, 35]. Beals et al. [5] show how the quantum circuit model can be efficiently simulated by the more realistic model of a distributed quantum computer (see also [20]). The setting of *measurement-based* quantum computing (see [25] and references therein) in some cases allows more parallelization than the usual circuit model. Another example, the only one we know of in the setting of query complexity, is Zalka's tight analysis of parallelizing quantum search [36, Section 4]. Suppose one wants to search an n-bit database, with the ability to do p queries in parallel in one time-step. An easy way to make use of this parallelism is to view the database as p databases of n/p bits each, and to run a separate copy of Grover's algorithm on each of those. This finds a 1-position with high probability using $O(\sqrt{n/p})$ p-parallel queries, and Zalka showed that this is optimal.

Our results. We focus on parallel quantum algorithms in the setting of quantum query complexity. Consider a function $f : \mathcal{D} \to \{0, 1\}$, with $\mathcal{D} \subseteq [q]^n$. For standard (sequential) query complexity, let $Q(f)$ denote the bounded-error

[4] Parallelism is in fact *necessary* to do quantum error-correction against a constant noise rate: sequential operations cannot keep up with the parallel build-up of errors.

quantum query complexity of f, i.e., the minimal number of queries needed among all quantum algorithms that (for every input $x \in \mathcal{D}$) output $f(x)$ with probability at least 2/3. In the p-parallel query model, for some integer $p \geq 1$, an algorithm can make up to p quantum queries in parallel in each timestep. In that case, we let $Q^{p\|}(f)$ denote the bounded-error p-parallel complexity of f. As always in query complexity, all intermediate input-independent computation is free. For every function, we have $Q(f)/p \leq Q^{p\|}(f) \leq Q(f)$.

An extreme case of the parallel model is where p large enough so that $Q^{p\|}(f)$ becomes 1; such algorithms are called "nonadaptive," because all queries are made in parallel. Montanaro [28] showed that for total functions, such nonadaptive quantum algorithms cannot improve much over classical algorithms: every Boolean function that depends on n input bits needs $p \geq n/2$ nonadaptive quantum queries for exact computation, and $p = \Omega(n)$ for bounded-error.

Here we prove matching upper and lower bounds on the p-parallel complexity $Q^{p\|}(f)$ for a number of problems: $\Theta((n/p)^{2/3})$ queries for element distinctness and $\Theta((n/p)^{k/(k+1)})$ for the k-sum problem for any constant $k > 1$. Our upper bounds are obtained by parallelized quantum walk algorithms, and our lower bounds are based on a modification of the adversary lower bound method combined with some recent results by Belovs et al. about using so-called "learning graphs," both for upper and for lower bounds [9, 13, 10, 14]. The modification we need to make is surprisingly small, and technically we need to do little more than adapt recent progress on sequential algorithms to the parallel case. Still, we feel this extension is important because: (1) our techniques may be useful for proving future lower bounds; (2) parallel quantum algorithms are important and yet have received little attention before; and (3) the fact that the extension is easy and natural increases our confidence that the adversary method is the "right" approach in the parallel as well as the sequential case.

In Section 5 we prove some more "structural" results, i.e., bounds for $Q^{p\|}(f)$ that hold for all Boolean functions $f : \{0,1\}^n \to \{0,1\}$. Specifically, based on earlier results in the sequential model due to Beals et al. [6], we show that if p is not too large then $Q^{p\|}(f)$ is polynomially related to its classical deterministic p-parallel counterpart. We also observe that $Q^{p\|}(f) \approx n/2p$ for almost all f.

2 Preliminaries

Sequential and parallel query complexity. We use $[n] := \{1, \ldots, n\}$, $\binom{[n]}{k} := \{S \subseteq [n] : |S| = k\}$, $\binom{[n]}{\leq k} := \{S \subseteq [n] : |S| \leq k\}$, and $\binom{n}{\leq k} := \sum_{s=0}^{k} \binom{n}{s}$.

We will consider algorithms in the p-parallel quantum query model. A quantum query to an input $x \in [q]^n$ corresponds to the unitary map $|i, b\rangle \mapsto |i, b + x_i\rangle$. Here the first n-dimensional register contains the index $i \in [n]$ of the queried element, and the value of that element is added (in $\mathbb{Z}_q$) to the contents of the second (q-dimensional) register. In order to enable an algorithm to not make a query on part of its state, we extend the previous unitary map to the case $i = 0$ by $|0, b\rangle \mapsto |0, b\rangle$. In each timestep we can make up to p quantum queries in

parallel by applying the map $|i_1, b_1, \ldots, i_p, b_p\rangle \mapsto |i_1, b_1 + x_{i_1}, \ldots, i_p, b_p + x_{i_p}\rangle$ at unit cost. All intermediate input-independent computation is free.

Consider a function $f : \mathcal{D} \to \{0,1\}$, with $\mathcal{D} \subseteq [q]^n$. When $p = 1$ we have the standard sequential query complexity, and we let $Q_\varepsilon(f)$ denote the quantum query complexity of f with error probability $\leq \varepsilon$ on every input $x \in \mathcal{D}$. For general p, let $Q_\varepsilon^{p\|}(f)$ be the p-parallel complexity of f. Note that $Q_\varepsilon(f)/p \leq Q_\varepsilon^{p\|}(f) \leq Q_\varepsilon(f)$ for every function. The exact value of the error probability ε does not matter, as long as it is a constant $< 1/2$. We usually fix $\varepsilon = 1/3$, abbreviating $Q(f) = Q_{1/3}(f)$ and $Q^{p\|}(f) = Q_{1/3}^{p\|}(f)$ as in the introduction.

We will use an extension of the adversary bound for the usual sequential (1-parallel) quantum query model. An *adversary matrix* Γ for f is a real-valued matrix whose rows are indexed by $f^{-1}(0)$ and whose columns by $f^{-1}(1)$. Let Δ_j be the Boolean matrix whose rows and columns are indexed by $x \in f^{-1}(0)$ and $y \in f^{-1}(1)$, such that $\Delta_j[x,y] = 1$ if $x_j \neq y_j$, and $\Delta_j[x,y] = 0$ otherwise. The (negative-weights) adversary bound for f is given by:

$$\mathrm{ADV}(f) = \max_{\Gamma} \frac{\|\Gamma\|}{\max_{j\in[n]} \|\Gamma \circ \Delta_j\|}, \tag{1}$$

where Γ ranges over all adversary matrices for f, '$\circ$' denotes entry-wise product of two matrices, and '$\|\cdot\|$' denotes the operator norm associated to the ℓ_2 norm. This lower bound (often denoted $\mathrm{ADV}^{\pm}(f)$ instead of $\mathrm{ADV}(f)$) was introduced by Høyer et al. [22], generalizing Ambainis [2]. They showed $Q_\varepsilon(f) \geq \frac{1}{2}(1 - \sqrt{\varepsilon(1-\varepsilon)})\mathrm{ADV}(f)$ for all f. Reichardt et al. [32, 26] showed this is tight: $Q(f) = \Theta(\mathrm{ADV}(f))$ for all f.

Quantum walks. We will construct and analyze our algorithms in the quantum walk framework of [27], which we now briefly describe. Given a reversible Markov process P on state space V, and a subset $M \subset V$ of marked elements, we define three costs: the setup cost, S, is the cost to construct a superposition over all states $\sum_{v\in V} \sqrt{\pi_v}|v\rangle$, where π_v is the probability of vertex v in the stationary distribution π of P; the checking cost, C, is the cost to check if a state $v \in V$ is in M; and the update cost, U, is the cost to perform the map $|v\rangle|0\rangle \mapsto |v\rangle \sum_{u\in V} \sqrt{P_{vu}}|u\rangle$, where P_{vu} is the transition probability in P to go from v to u. Then, if δ is the spectral gap of P, and ε is a lower bound on $\sum_{v\in M} \pi_v$ whenever M is nonempty, we can determine if M is nonempty with bounded error probability in cost $O\left(\mathsf{S} + \frac{1}{\sqrt{\varepsilon}}\left(\frac{1}{\sqrt{\delta}}\mathsf{U} + \mathsf{C}\right)\right)$. If S, U and C denote query complexities, then the above expression gives the bounded-error query complexity of the quantum walk algorithm. If they denote *p-parallel* query complexities, the above expression gives the bounded-error p-parallel complexity.

3 Lower bounds for parallel quantum query complexity

3.1 Adversary bound for parallel algorithms

We start by extending the adversary bound for the usual sequential quantum query algorithms to p-parallel algorithms. For $J \subseteq [n]$, let x_J be the string x

restricted to the entries in J. Let Δ_J be the Boolean matrix whose rows are indexed by $x \in f^{-1}(0)$ and whose columns are indexed by $y \in f^{-1}(1)$, and that has a 1 at position (x, y) iff $x_J \neq y_J$ (i.e., $x_j \neq y_j$ for at least one $j \in J$). For $J = \emptyset$, Δ_J is the all-0 matrix. Define the following quantity:

$$\mathrm{ADV}^{p\|}(f) = \max_\Gamma \frac{\|\Gamma\|}{\max_{J \in \binom{[n]}{\leq p}} \|\Gamma \circ \Delta_J\|}. \tag{2}$$

The following fact (proved in our full version [24]) implies we only need to consider sets $J \in \binom{[n]}{p}$ in the above definition: $\mathrm{ADV}^{p\|}(f)$ equals $\max_\Gamma \frac{\|\Gamma\|}{\max_{J \in \binom{[n]}{p}} \|\Gamma \circ \Delta_J\|}$ up to a factor of 2. We could even use the latter as an alternative definition of $\mathrm{ADV}^{p\|}(f)$.

Fact 1 *For every set $J \subseteq K \subseteq [n]$, we have $\|\Gamma \circ \Delta_J\| \leq 2\|\Gamma \circ \Delta_K\|$.*

Theorem 2. *For every $f : \mathcal{D} \to \{0,1\}$ and $\mathcal{D} \subseteq [q]^n$, $Q^{p\|}(f) = \Theta(\mathrm{ADV}^{p\|}(f))$.*

Proof. In order to derive p-parallel lower bounds from sequential lower bounds, observe that we can make a bijection between input $x \in [q]^n$ and a larger string X indexed by all sets $J \in \binom{[n]}{\leq p}$, such that $X_J = (x_j)_{j \in J}$. That is, each index J of X corresponds to up to p indices j of x. We now define a new function $F : \mathcal{D}' \to \{0,1\}$, where $\mathcal{D}'$ is the set of X as above, in 1-to-1 correspondence with the elements of $x \in \mathcal{D}$, and $F(X)$ is defined as $f(x)$. One query to X can be simulated by p parallel queries to x, and vice versa, so we have $Q^{p\|}(f) = Q(F)$. We have $Q(F) = \Theta(\mathrm{ADV}(F))$ by [32, 26]. Now Eq. (1) applied to F gives the claimed lower bound of Eq. (2) on $Q^{p\|}(f)$. □

Sometimes we can even use the same adversary matrix Γ to obtain optimal lower bounds for F and f. An example is the n-bit OR-function. Let Γ be the all-ones $1 \times n$ matrix, with the row corresponding to input 0^n and the columns indexed by all weight-1 inputs. Then $\|\Gamma\| = \sqrt{n}$ and $\|\Gamma \circ \Delta_j\| = 1$ for all $j \in [n]$, and hence $Q(\mathrm{OR}) = \Omega(\sqrt{n})$. To get p-parallel lower bounds, we define a new function $F : X \mapsto \{0,1\}$ as in the proof of Theorem 2. We can use the same Γ, with the n columns still indexed by the weight-1 inputs to f (which induce 1-inputs to F). Now J ranges over subsets of $[n]$ of size at most p, and Δ_J will be the matrix whose (x, y)-entry is 1 if there is at least one $j \in J$ such that $x_j \neq y_j$. Note that $\|\Gamma \circ \Delta_J\| = \sqrt{|J|}$ for all J. Hence $Q^{p\|}(\mathrm{OR}) = \Omega(\mathrm{ADV}(F)) = \Omega(\sqrt{n/p})$. This is optimal and was already proved (in a different way) by Zalka [36, Section 4].

3.2 Belovs's learning graph approach

Recently Belovs [9] gave a new approach to designing quantum algorithms, introducing the model of *learning graphs* to prove upper bounds on the adversary

bound, and hence on quantum query complexity. We state it here for *certificate structures*. We define these below, slightly simpler and less general than Definitions 1 and 3 of Belovs and Rosmanis [13] (for us M denotes a minimal certificate, while in [13] it denotes the set of supersets of a minimal certificate).

Definition 1. *Let $\mathcal{C}$ be a set of incomparable subsets of $[n]$. We say $\mathcal{C}$ is a* 1-certificate structure *for a function $f : \mathcal{D} \to \{0,1\}$, with $\mathcal{D} \subseteq [q]^n$, if for every $x \in f^{-1}(1)$ there exists an $M \in \mathcal{C}$ such that for all $y \in \mathcal{D}$, $y_M = x_M$ implies $f(y) = 1$. We say $\mathcal{C}$ is* k-bounded *if $|M| \leq k$ for all $M \in \mathcal{C}$.*

The learning graph complexity of $\mathcal{C}$ is defined in the following in its primal formulation as a minimization problem (we will see an equivalent dual formulation soon). Let $\mathcal{E} = \{(S, j) : S \subseteq [n], j \in [n] \backslash S\}$. For $e = (S, j) \in \mathcal{E}$, we use $s(e) = S$ and $t(e) = S \cup \{j\}$.

$$\mathrm{LGC}(\mathcal{C}) = \min \sqrt{\textstyle\sum_{e \in \mathcal{E}} w_e} \quad \text{such that} \tag{3}$$

$$\sum_{e \in \mathcal{E}} \frac{\theta_e(M)^2}{w_e} \leq 1 \qquad \text{for all } M \in \mathcal{C} \tag{4}$$

$$\sum_{e \in \mathcal{E}: t(e) = S} \theta_e(M) = \sum_{e \in \mathcal{E}: s(e) = S} \theta_e(M) \quad \text{for all } M \in \mathcal{C}, \emptyset \neq S \subseteq [n], M \not\subseteq S \tag{5}$$

$$\sum_{e \in \mathcal{E}: s(e) = \emptyset} \theta_e(M) = 1 \qquad \text{for all } M \in \mathcal{C} \tag{6}$$

$$\theta_e(M) \in \mathbb{R}, w_e \geq 0 \qquad \text{for all } e \in \mathcal{E} \text{ and } M \in \mathcal{C} \tag{7}$$

For each M, $\theta_e(M)$ is a *flow* from $\emptyset$ to M on the graph with vertices $\{S \subseteq [n]\}$ and edges $\{\{S, S \cup \{j\}\} : (S, j) \in \mathcal{E}\}$ if $\theta_e(M)$ satisfies condition (5). Moreover, $\theta_e(M)$ is a *unit flow* if it also satisfies condition (6).

The learning graph complexity of $\mathcal{C}$ is an upper bound on $\mathrm{ADV}(f)$, and hence on $Q(f)$, for any function f with certificate structure $\mathcal{C}$. This bound is not always optimal, since it only depends on the certificate structure of f. E.g. k-distinctness has quantum query complexity $o(n^{3/4})$ even though it has the same 1-certificate structure as k-sum, whose quantum query complexity is $\Theta(n^{k/(k+1)})$ [10, 14]. However, Belovs and Rosmanis [13] proved that for a special class of functions, it turns out the upper bound $\mathrm{LGC}(\mathcal{C})$ is optimal.

Definition 2. *An* orthogonal array *of length k is a set $T \subseteq [q]^k$, such that for every $i \in [k]$ and every $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_k$ there exists exactly one $x_i \in [q]$ such that $(x_1, \ldots, x_k) \in T$.*

Theorem 3 (Belovs-Rosmanis). *Let $\mathcal{C}$ be a k-bounded 1-certificate structure for some constant k, $q \geq 2|\mathcal{C}|$, and let each $M \in \mathcal{C}$ be equipped with an orthogonal array T_M of length $|M|$. Define a Boolean function $f : [q]^n \to \{0,1\}$ by: $f(x) = 1$ iff there exists an $M \in \mathcal{C}$ such that $x_M \in T_M$. Then $Q(f) = \Theta(\mathrm{LGC}(\mathcal{C}))$.*

For example, the element distinctness problem ED on input $x \in [q]^n$ is induced by the 2-bounded 1-certificate structure $\mathcal{C} = \binom{[n]}{2}$, equipped with associated orthogonal arrays $T_{\{i,j\}} = \{(v, v) : v \in [q]\}$. Hence $Q(\mathrm{ED}) = \Theta(\mathrm{LGC}(\mathcal{C}))$.

Belovs and Rosmanis [13] show that an equivalent dual definition of the learning graph complexity as a maximization problem is the following:

$$\mathrm{LGC}(\mathcal{C}) = \max \sqrt{\textstyle\sum_{M\in\mathcal{C}} \alpha_\emptyset(M)^2} \tag{8}$$

$$\begin{aligned} \text{s.t. } & \textstyle\sum_{M\in\mathcal{C}} (\alpha_{s(e)}(M) - \alpha_{t(e)}(M))^2 \le 1 && \text{for all } e \in \mathcal{E} \\ & \alpha_S(M) = 0 && \text{whenever } M \subseteq S \\ & \alpha_S(M) \in \mathbb{R} && \text{for all } S \subseteq [n] \text{ and } M \in \mathcal{C} \end{aligned} \tag{9}$$

In particular, that means we can prove *lower* bounds on $\mathrm{LGC}(\mathcal{C})$ (and hence, for the functions described in Theorem 3, on $Q(f)$) by exhibiting a feasible solution $\{\alpha_S(M)\}$ for this maximization problem and calculating its objective value.

Before stating a similar result for p-parallel query complexity, we first adapt learning graphs. Edges, which were of type $e = (S, j)$ with $S \subseteq [n]$ and $j \in [n]\setminus S$, are now of type $e = (S, J)$ with $S \subseteq [n]$, $J \subseteq [n] \setminus S$ and $|J| \le p$.

Definition 3. *The* p-parallel learning graph complexity $\mathrm{LGC}^{p\|}(\mathcal{C})$ *of* $\mathcal{C}$ *is defined as* $\mathrm{LGC}(\mathcal{C})$ *where we replace the edge set* $\mathcal{E}$ *with* $\mathcal{E}_p = \{(S, J) : S \subseteq [n], J \subseteq [n] \setminus S, |J| \le p\}$*. Its dual is analogous. In particular, we replace* (9) *by*

$$\textstyle\sum_{M\in\mathcal{C}} (\alpha_{s(e)}(M) - \alpha_{t(e)}(M))^2 \le 1 \text{ for all } e = (S, J) \in \mathcal{E}_p,$$

where $s(e) = S$ *and* $t(e) = S \cup J$*. We call this modified constraint "parallel-*(9)*."*

As in the special case of $p = 1$, the p-parallel learning graph complexity of $\mathcal{C}$ provides an upper bound on $\mathrm{ADV}^{p\|}(f)$, and hence on $Q^{p\|}(f)$, for any function f having that same certificate structure. The proof is given in our full version [24].

Lemma 1. *Let* $\mathcal{C}$ *be a certificate structure for* f*. Then* $\mathrm{ADV}^{p\|}(f) \le \mathrm{LGC}^{p\|}(\mathcal{C})$.

We now generalize Theorem 3 to the p-parallel case. The proof, given in [24], is an adaptation of the proof of [13, Theorem 5].

Theorem 4. *Let* $\mathcal{C}$ *be a* k*-bounded 1-certificate structure for some constant* k*,* $q \ge 2|\mathcal{C}|$*, and let each* $M \in \mathcal{C}$ *be equipped with an orthogonal array* T_M *of length* $|M|$*. Define a Boolean function* $f : [q]^n \to \{0, 1\}$ *as follows:* $f(x) = 1$ *iff there exists an* $M \in \mathcal{C}$ *such that* $x_M \in T_M$*. Then* $Q^{p\|}(f) = \Theta(\mathrm{LGC}^{p\|}(\mathcal{C}))$.

4 Parallel quantum query complexity of specific functions

4.1 Algorithms

In this section we give upper bounds for element distinctness and k-sum in the p-parallel quantum query model, by way of quantum walk algorithms.

Our p-parallel algorithm for element distinctness is based on the sequential query algorithm for element distinctness of Ambainis [3]. Ambainis's algorithm uses a quantum walk on a Johnson graph, $J(n, r)$, which has vertex set $V = \{S \subseteq [n] : |S| = r\}$ and edge set $\{\{S, S'\} \subseteq V : |S \setminus S'| = 1\}$. Each state $S \in V$ represents a set of queried indices. The algorithm seeks a state S containing (i, x_i) and (j, x_j) such that $i \ne j$ and $x_i = x_j$. Such a state is said to be *marked*.

Theorem 5. *Element distinctness on* $[q]^n$ *has* $Q^{p\|}(\mathrm{ED}) = O((n/p)^{2/3})$.

Proof. We modify Ambainis's quantum walk algorithm slightly. Consider a walk $J(n, r/p)^p$, on p copies of the Johnson graph $J(n, r/p)$. Vertices are p-tuples $(S_1, S_2, \ldots, S_p)$ where, for each $i \in [p]$, $S_i \subseteq [n]$ and $|S_i| = r/p$. Two vertices $(S_1, S_2, \ldots, S_p)$ and $(S'_1, S'_2, \ldots, S'_p)$ are adjacent if, for each $i \in [p]$, $|S_i \setminus S'_i| = 1$. We call a state $(S_1, S_2, \ldots, S_p)$ *marked* if there are $j, j' \in \bigcup_{i=1}^p S_i$ such that $x_j = x_{j'}$. Since the stationary distribution is μ^p, where μ is the uniform distribution on $\binom{[n]}{r/p}$, the probability that a state is marked is at least $\varepsilon = \Omega(r^2/n^2)$.

The setup cost is only $\mathsf{S} = O(r/p)$ p-parallel queries, since it suffices to query r elements in the initial superposition over all states. Similarly, the update requires that we query and unquery p elements, but we can accomplish this in two p-parallel queries, so $\mathsf{U} = O(1)$. Also, $\mathsf{C} = 0$. Finally, the eigenvalues of the product of p copies of a graph are exactly the products of p eigenvalues of that graph. Hence if the largest eigenvalue of a graph is 1 and the second-largest is $1 - \delta$, then the same will be true for the product graph. Accordingly, the spectral gap δ of p copies of $J(n, r/p)$ is exactly the spectral gap of one copy of $J(n, r/p)$, which is $\Omega(p/r)$. We can now calculate the p-parallel query complexity of element distinctness as $O\left(\mathsf{S} + \frac{1}{\sqrt{\varepsilon}}\left(\left(\frac{1}{\sqrt{\delta}}\right)\mathsf{U} + \mathsf{C}\right)\right) = O\left(\frac{r}{p} + \left(\frac{n}{r}\right)\left(\sqrt{\frac{r}{p}}\right)\right) = O\left(\frac{r}{p} + \frac{n}{\sqrt{rp}}\right)$. Setting r to the optimal $n^{2/3}p^{1/3}$ gives an upper bound of $O((n/p)^{2/3})$. □

It is easy to generalize our element distinctness upper bound to k-sum:

Theorem 6. *k-sum on* $[q]^n$ *has* $Q^{p\|}(k\text{-sum}) = O((n/p)^{k/(k+1)})$.

Proof. Again, we walk on p copies of $J(n, r/p)$, but now we consider a state $(S_1, S_2, \ldots, S_p)$ marked if there are distinct indices $i_1, \ldots, i_k \in \bigcup_{i=1}^p S_i$ such that $\sum_{j=1}^k x_{i_j} = 0 \pmod q$. The proportion of marked states in a 1-instance is $\varepsilon = \Omega(r^k/n^k)$. All other parameters are as in Theorem 5. We get the following upper bound for k-sum: $O\left(\mathsf{S} + \frac{1}{\sqrt{\varepsilon}}\left(\frac{1}{\sqrt{\delta}}\mathsf{U} + \mathsf{C}\right)\right) = O\left(\frac{r}{p} + \frac{n^{k/2}}{r^{k/2}}\left(\sqrt{\frac{r}{p}}\right)\right) = O\left(\frac{r}{p} + \frac{n^{k/2}}{r^{(k-1)/2}\sqrt{p}}\right)$. Setting $r = n^{k/(k+1)}p^{1/(k+1)}$ gives $O((n/p)^{k/(k+1)})$. □

4.2 Lower bounds

We now use the ideas from Section 3.2 to prove p-parallel lower bounds for ED and k-sum, matching our upper bounds if the alphabet size q is sufficiently large. Our proofs are generalizations of the sequential lower bounds in [13, Section 4].

Theorem 7. *For* $q \geq 2\binom{n}{2}$, ED *on* $[q]^n$ *has* $Q^{p\|}(\mathrm{ED}) = \Omega((n/p)^{2/3})$.

Proof. Recall that element distinctness is induced by the 1-certificate structure $\mathcal{C} = \binom{[n]}{2}$, equipped with associated orthogonal arrays $T_{\{i,j\}} = \{(v, v) : v \in [q]\}$. By Theorem 4, it suffices to prove the lower bound on the p-parallel learning graph complexity of ED. For this, it suffices to exhibit a feasible solution to the

dual (8) and to lower bound its objective function. Note that the elements of $\mathcal{E}$ are now of the form (S, J), where $S \subseteq [n]$ and $J \subseteq [n] \setminus S$ with $|J| \leq p$. Define

$$\alpha_j = \tfrac{1}{2n} \max((n/p)^{2/3} - j/p, 0), \quad \text{and} \quad \alpha_S(M) = \begin{cases} 0 & \text{if } M \subseteq S \\ \alpha_{|S|} & \text{otherwise.} \end{cases}$$

To show that this is a feasible solution, the only constraint we need to verify is parallel-(9). Fix $S \subseteq [n]$ of some size s, and a set $J \subseteq [n] \setminus S$ with $|J| \leq p$. Let L denote the left-hand side of parallel-(9), which is a sum over all $\binom{n}{2}$ certificates $M \in \mathcal{C}$. With respect to $e = (S, J)$, there are four kinds of $M = \{i, j\}$:

1. $i, j \in S$. Then $\alpha_{t(e)}(M) = \alpha_{s(e)}(M) = 0$, so these M contribute 0 to L.
2. $i \in S, j \in J$. There are $s|J| \leq sp$ such M, and each contributes α_s^2 to L because $\alpha_{s(e)}(M) = \alpha_s$ and $\alpha_{t(e)}(M) = 0$.
3. $i, j \notin S$, $i, j \in J$. There are $\binom{|J|}{2} \leq \binom{p}{2}$ such M, each contributes α_s^2 to L.
4. i and/or $j \notin S \cup J$. There are $\binom{n}{2} - \binom{s+|J|}{2} \leq n^2$ such M, each contributes $|\alpha_s - \alpha_{s+|J|}|^2$ to L.

Hence, using $\alpha_s = 0$ if $s \geq n^{2/3}p^{1/3}$, $\alpha_s \leq \alpha_0 = \frac{1}{2p^{2/3}n^{1/3}}$, and $|\alpha_s - \alpha_{s+|J|}|^2 \leq 1/4n^2$, we can establish constraint parallel-(9):

$$L \leq \left(sp + \tbinom{p}{2}\right) \alpha_s^2 + n^2 |\alpha_s - \alpha_{s+|J|}|^2 \leq p(n^{2/3}p^{1/3} + p/2) \tfrac{1}{4p^{4/3}n^{2/3}} + n^2 \tfrac{1}{4n^2} \leq 1.$$

Hence our solution is feasible. Its objective value is $\sqrt{\binom{n}{2}\alpha_0^2} = \Omega((n/p)^{2/3})$. □

The lower bound proof for k-sum is similar. Here we use certificate structure $\mathcal{C} = \binom{[n]}{k}$ with the orthogonal array $T = \{(v_1, \ldots, v_k) : \sum_{i=1}^k v_i = 0 \mod q\}$, which induces k-sum. In [24], we show that the following has objective value $\sqrt{\binom{n}{k}\alpha_0^2} = \Omega\left((n/p)^{k/(k+1)}\right)$ and is feasible for $\mathrm{LGC}^{p\|}(\mathcal{C})$:

$$\alpha_j = \tfrac{1}{2n^{k/2}} \max((n/p)^{k/(k+1)} - j/p, 0) \quad \text{and} \quad \alpha_S(M) = \begin{cases} 0 & \text{if } M \subseteq S \\ \alpha_{|S|} & \text{otherwise;} \end{cases}$$

Theorem 8. *For $q \geq 2\binom{n}{k}$, k-sum on $[q]^n$ has $Q^{p\|}(k\text{-sum}) = \Omega\left((n/p)^{k/(k+1)}\right)$.*

5 Some general bounds

In this section we will relate quantum and classical p-parallel complexity. For the sequential model ($p = 1$) it is known that quantum bounded-error query complexity is no more than a 6th power less than classical deterministic complexity, for all total Boolean functions [6]. Here we will see to what extent we can prove a similar result for the p-parallel model.

We start with a few definitions, referring to [15] for more details. Let $f : \{0,1\}^n \to \{0,1\}$ be a total Boolean function. For $b \in \{0,1\}$, a *b-certificate* for f is an assignment $C : S \to \{0,1\}$ to a subset S of the n variables, such that

$f(x) = b$ whenever x is consistent with C. The *size* of C is $|S|$. The *certificate complexity* $C_x(f)$ *of* f *on* x is the size of a smallest $f(x)$-certificate that is consistent with x. The *certificate complexity* of f is $C(f) = \max_x C_x(f)$. The 1-*certificate complexity* of f is $C^{(1)}(f) = \max_{\{x:f(x)=1\}} C_x(f)$. Given an input $x \in \{0,1\}^n$ and subset $B \subseteq [n]$ of indices of variables, let x^B denote the n-bit input obtained from x by negating all bits x_i whose index i is in B. The *block sensitivity* $bs(f,x)$ of f at input x, is the maximal integer k such that there exist disjoint sets $B_1, \ldots, B_k$ satisfying $f(x) \neq f(x^{B_i})$ for all $i \in [k]$. The *block sensitivity* of f is $bs(f) = \max_x bs(f,x)$. Nisan [30] proved that

$$bs(f) \leq C(f) \leq bs(f)^2. \tag{10}$$

Via a standard reduction [31], Zalka's $\Theta(\sqrt{n/p})$ bound for OR implies:

Theorem 9. *For every* $f : \{0,1\}^n \to \{0,1\}$, $Q^{p\|}(f) = \Omega(\sqrt{bs(f)/p})$.

We now prove a general upper bound on deterministic p-parallel complexity:

Theorem 10. *For every* $f : \{0,1\}^n \to \{0,1\}$, $D^{p\|}(f) \leq \lceil C^{(1)}(f)/p \rceil bs(f)$.

Proof. Beals et al. [6, Lemma 5.3] give a deterministic decision tree for f that runs for at most $bs(f)$ rounds, and in each round queries all variables of a 1-certificate, and substituting their values into the function. This reduces the function to a constant. By parallelizing the querying of the certificate we can implement every round using $\lceil C^{(1)}(f)/p \rceil$ p-parallel steps. □

$D^{p\|}(f)$ and $Q^{p\|}(f)$ are polynomially related if p is not too big:

Theorem 11. *For every* $f : \{0,1\}^n \to \{0,1\}$, $c > 1$, $p \leq bs(f)^{1/c}$, *we have* $D^{p\|}(f) \leq O(Q^{p\|}(f)^{6+4/(c-1)})$.

Proof. We can assume $C(f) = C^{(1)}(f)$ (else consider $1-f$). By Eq. (10) we have $p \leq bs(f)^{1/c} \leq C^{(1)}(f)$. We also have $C^{(1)}(f) \leq bs(f)^2$. The assumption on p is equivalent to $p \leq (bs(f)/p)^{1/(c-1)}$. Using Theorems 9 and 10, we obtain

$$\begin{aligned} D^{p\|}(f) &\leq \lceil C^{(1)}(f)/p \rceil bs(f) \leq O(bs(f)^3/p) = O((bs(f)/p)^3 p^2) \\ &\leq O((bs(f)/p)^{3+2/(c-1)}) \leq O(Q^{p\|}(f)^{6+4/(c-1)}). \end{aligned}$$

□

For example, if $p \leq bs(f)^{1/3}$ then $Q^{p\|}(f)$ is at most an 8th power smaller than $D^{p\|}(f)$. Whether superpolynomial gaps exist for large p remains open.

We end with an observation about random functions. Van Dam [17] showed that an n-bit input string x can be recovered with high probability using $n/2 + O(\sqrt{n})$ quantum queries, hence $Q(f) \leq n/2 + O(\sqrt{n})$ for all $f : \{0,1\}^n \to \{0,1\}$. His algorithm already applies its queries in parallel, so allows us to compute x using roughly $n/2p$ p-parallel quantum queries. Ambainis et al. [4] proved an essentially optimal lower bound for random functions: almost all f have $Q(f) \geq (1/2 - o(1))n$. Since trivially $Q(f) \leq pQ^{p\|}(f)$, we obtain the p-parallel lower bound $Q^{p\|}(f) \geq (1/2 - o(1))n/p$ for almost all f.

6 Conclusion and future work

This paper is the first to systematically study the power and limitations of parallelism for quantum query algorithms. We leave open many interesting questions:

- There are many other computational problems whose p-parallel complexity is unknown, for example finding a triangle in a graph or deciding whether two given matrices multiply to a third one. For both of these problems, however, even the sequential quantum query complexity is still open.
- We suspect Theorem 11 is non-optimal, and conjecture that $D^{p\|}(f)$ and $Q^{p\|}(f)$ are polynomially related for large p as well. Montanaro's result [28] about the weakness of maximally parallel quantum algorithms is evidence.
- Can we find relations with quantum communication complexity? Nonadaptive quantum query algorithms induce one-way communication protocols, while fully adaptive ones induce protocols that are very interactive. Our p-parallel algorithms would sit somewhere in between.

Acknowledgment. We are partially supported by the French ANR Blanc project ANR-12-BS02-005 (RDAM), a Vidi grant from the Netherlands Organization for Scientific Research (NWO), an ERC Consolidator grant, the European Commission IST STREP project Quantum Algorithms (QALGO) 600700 and the US ARO. We thank Jérémie Roland for helpful discussions.

References

1. S. Aaronson and Y. Shi. Quantum lower bounds for the collision and the element distinctness problems. *J. of the ACM*, 51(4):595–605, 2004.
2. A. Ambainis. Quantum lower bounds by quantum arguments. *J. of Computer and System Sciences*, 64(4):750–767, 2002.
3. A. Ambainis. Quantum walk algorithm for element distinctness. *SIAM J. on Computing*, 37(1):210–239, 2007. Earlier version in FOCS'04. quant-ph/0311001.
4. A. Ambainis, A. Bačkurs, J. Smotrovs, R. de Wolf. Optimal quantum query bounds for almost all Boolean functions. In *Proc. 30th STACS*, pages 446–453, 2013.
5. R. Beals, S. Brierley, O. Gray, A. Harrow, S. Kutin, N. Linden, D. Shepherd, and M. Stather. Efficient distributed quantum computing. *Proc. of the Royal Society*, A469(2153), 2013.
6. R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *J. of the ACM*, 48(4):778–797, 2001.
7. N. de Beaudrap, R. Cleve, and J. Watrous. Sharp quantum vs. classical query complexity separations. *Algorithmica*, 34(4):449–461, 2002.
8. A. Belovs. Learning-graph-based quantum algorithm for k-distinctness. In *Proc. of 53rd IEEE FOCS*, pages 207–216, 2012.
9. A. Belovs. Span programs for functions with constant-sized 1-certificates. In *Proc. of 43rd ACM STOC*, pages 77–84, 2012.
10. A. Belovs. Adversary lower bound for element distinctness, arXiv:1204.5074, 2012.
11. A. Belovs, A. Childs, S. Jeffery, R. Kothari, and F. Magniez. Time-efficient quantum walks for 3-distinctness. In *Proc. of 40th ICALP(1)*, pages 105–122, 2013.
12. A. Belovs and T. Lee. Quantum algorithm for k-distinctness with prior knowledge on the input. Technical Report arXiv:1108.3022, arXiv, 2011.

13. A. Belovs and A. Rosmanis. On the power of non-adaptive learning graphs. In *Proc. of 28th IEEE CCC*, pages 44–55, 2013. References are to arXiv:1210.3279v2.
14. A. Belovs and R. Špalek. Adversary lower bound for the k-sum problem. In *Proc. of 4th ITCS*, pages 323–328, 2013.
15. H. Buhrman and R. de Wolf. Complexity measures and decision tree complexity: A survey. *Theoretical Computer Science*, 288(1):21–43, 2002.
16. R. Cleve and J. Watrous. Fast parallel circuits for the quantum Fourier transform. In *Proc. of 41st IEEE FOCS*, pages 526–536, 2000.
17. W. van Dam. Quantum oracle interrogation: Getting all information for almost half the price. In *Proc. of 39th IEEE FOCS*, pages 362–367, 1998.
18. D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. In *Proc. of the Royal Society of London*, volume A439, pages 553–558, 1992.
19. F. Green, S. Homer, C. Moore, and C. Pollett. Counting, fanout and the complexity of quantum ACC. *Quantum Inf. and Comp.*, 2(1):35–65, 2002.
20. L. Grover and T. Rudolph. How significant are the known collision and element distinctness quantum algorithms? *Quantum Inf. and Comp.*, 4(3):201–206, 2004.
21. L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proc. of 28th ACM STOC*, pages 212–219, 1996.
22. P. Høyer, T. Lee, and R. Špalek. Negative weights make adversaries stronger. In *Proc. of 39th ACM STOC*, pages 526–535, 2007.
23. P. Høyer and R. Špalek. Quantum fan-out is powerful. *Th. Comp.*, 1:81–103, 2005.
24. S. Jeffery, F. Magniez, and R. de Wolf. Optimal parallel quantum query algorithms. arXiv:1309.6116, 2013.
25. R. Jozsa. An introduction to measurement based quantum computation. In D.G. Angelakis, M. Christandl, and A. Ekert, editors, *Quantum Information Processing*, pages 137–158. IOS Press, 2006. arXiv:0508124.
26. T. Lee, R. Mittal, B. Reichardt, R. Špalek, and M. Szegedy. Quantum query complexity of state conversion. In *Proc. of 52nd IEEE FOCS*, pages 344–353, 2011. References are to arXiv:1011.3020v2.
27. F. Magniez, A. Nayak, J. Roland, and M. Santha. Search via quantum walk. *SIAM J. on Computing*, 40(1):142–164, 2011.
28. A. Montanaro. Nonadaptive quantum query complexity. *Information Processing Letters*, 110(24):1110–1113, 2010.
29. C. Moore and M. Nilsson. Parallel quantum computation and quantum codes. *SIAM J. on Computing*, 31(3):799–815, 2002.
30. N. Nisan. CREW PRAMs and decision trees. *SIAM J. on Computing*, 20(6):999–1007, 1991.
31. N. Nisan and M. Szegedy. On the degree of Boolean functions as real polynomials. *Computational Complexity*, 4(4):301–313, 1994.
32. B. Reichardt. Span programs and quantum query complexity: The general adversary bound is nearly tight for every Boolean function. In *Proc. of 50th IEEE FOCS*, pages 544–551, 2009.
33. P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. on Computing*, 26(5):1484–1509, 1997.
34. D. Simon. On the power of quantum computation. *SIAM J. on Computing*, 26(5):1474–1483, 1997.
35. Y. Takahashi and S. Tani. Collapse of the hierarchy of constant-depth exact quantum circuits. In *Proc. of 28th IEEE CCC*, 2013.
36. Ch. Zalka. Grover's quantum searching algorithm is optimal. *Physical Review A*, 60:2746–2751, 1999.