Proceedings of the European Control Conference 2007
Kos, Greece, July 2-5, 2007

Finding quantum algorithms via convex optimization

Andrew M. Childs
Institute for Quantum Information
California Institute of Technology
Pasadena, CA 91125, USA

Andrew J. Landahl
Center for Advanced Studies
Dept. of Physics and Astronomy
University of New Mexico
Albuquerque, NM 87131, USA

Pablo A. Parrilo
Lab. Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

***Abstract*— In this paper we describe how to use convex optimization to design quantum algorithms for certain computational tasks. In particular, we consider the ordered search problem, where it is desired to find a specific item in an ordered list of N items. While the best classical algorithm for this problem uses $\log_2 N$ queries to the list, a quantum computer can solve this problem much faster. By characterizing a class of quantum query algorithms for ordered search in terms of a semidefinite program, we find quantum algorithms using $4\log_{605} N \approx 0.433\log_2 N$ queries, which improves upon the previously best known exact algorithm.**

I. Introduction

The discovery and development of quantum algorithms to date has mostly been the product of inspired guesswork by individual researchers. Our objective in this paper is to show how the search for quantum algorithms of a specific kind can be reduced to and effectively solved by using the ideas of convex optimization (in particular, semidefinite programming). We illustrate this approach for the problem of searching an ordered list (ordered search problem, OSP). For this problem, we have found an scalable quantum algorithm whose query complexity is superior to the best-known query algorithm (quantum or otherwise) for this task.

The *ordered search problem* (OSP) is the problem of finding the first occurrence of a target item in an ordered list of N items subject to the promise that the target item is somewhere in the list. Equivalently, we can remove the promise by viewing the OSP as the problem of finding the earliest insertion point for a target item in a sorted list of $N-1$ items. The OSP is ubiquitous in computation, not only in its own right, but also as a subroutine in algorithms for related problems, such as sorting.

One way of characterizing the computational difficulty of the ordered search problem is to quantify how many times the list must be queried to find the location of the target item. The minimal number of queries required to solve the problem in the worst case is known as its *query complexity*. Using information theoretic arguments, one can prove that any deterministic classical algorithm for the OSP requires at least $\lceil \log_2 N \rceil$ queries. This lower bound is achieved by the well-known binary search algorithm [1].

Quantum computers can solve the ordered search problem using a number of queries that is smaller by a constant factor than the number of queries used in the binary search algorithm. The best known lower bound, proved by Høyer, Neerbek, and Shi, shows that any quantum algorithm for the OSP that is *exact* (*i.e.*, succeeds with unit probability after a fixed number of queries) requires at least $(\ln N - 1)/\pi \approx 0.221\log_2 N$ queries [2]. In other words, at most a constant factor speedup is possible. The best published exact quantum OSP algorithm, obtained by Farhi, Goldstone, Gutmann, and Sipser, uses $3\lceil \log_{52} N \rceil \approx 0.526\log_2 N$ queries, showing that a constant factor speedup is indeed possible [3]. However, there remains a gap between the constants in these lower and upper bounds. Since the OSP is such a basic problem, it is desirable to establish the precise value of the constant factor speedup for the best possible quantum algorithm: this constant is a fundamental piece of information about the computational power of quantum mechanics.

In this paper we study the query complexity of the ordered search problem by exploiting a connection between quantum query problems and convex optimization. Specifically, we show that the existence of an algorithm for the OSP that is *translation invariant* (in the sense of [3]) is equivalent to the existence of a solution for a certain semidefinite program (SDP). By solving this semidefinite program numerically, we show that there is an exact quantum query algorithm to search a list of size $N = 605$ using just 4 queries.

Since the size of the semidefinite program increases as we increase N, we cannot directly perform a numerical search for a quantum ordered search algorithm for arbitrarily large problem instances. However, by applying the 4-query algorithm recursively, we see that there is an exact algorithm for a list of size N using $4\log_{605} N \approx 0.433\log_2 N$ queries. Thus, our result narrows the gap between the best known algorithm and the lower bound of [2]. In particular, this shows that the quantum query complexity of the OSP is strictly less than $\log_2 \sqrt{N}$, which one might have naively guessed was the query complexity of ordered search by analogy with the *unordered search problem*, whose quantum query complexity is $\Theta(\sqrt{N})$ [4], [5].

In addition to providing a way of searching for algorithms, the semidefinite programming approach has the advantage that a solution to the dual SDP provides a certificate of the non-existence of an algorithm. Thus we are able to provide some evidence (although not a proof) that $N = 605$ is the largest size of a list that can be searched with $k = 4$ queries, by showing that no algorithm exists for $N = 606$.

The remainder of the article is organized as follows. In

ISBN 978-3-9524173-8-6

Section II, we describe the class of translation invariant algorithms that we focus on and summarize known results about such algorithms. In Section III, we show how these algorithms can be characterized as the solutions of a semidefinite program, and discuss its convex duality properties. Finally, in Section IV, we present the results obtained by numerically solving this semidefinite program, followed by our conclusions.

II. Translation invariant quantum algorithms for ordered search

A. Quantum query algorithms

We briefly describe next the quantum query model used. For the full details, we refer the reader to [3], [6].

A k-query quantum algorithm is specified by an initial quantum state $|\psi_0\rangle$ and a sequence of unitary operators $U_1, U_2, \ldots, U_k$. The algorithm begins with the quantum computer in the state $|\psi_0\rangle$, and query transformations and the operations U_j are applied alternately, giving the final quantum state

$$|\phi_j\rangle := U_k G_j U_{k-1} \ldots U_1 G_j |\psi_0\rangle. \tag{1}$$

The operations G_j correspond to queries about the list of items. In the quantum mechanical version of the query model, access to the query function is provided by a unitary transformation. Specifically, we will use a *phase oracle* for g_j, a linear operator G_j defined by the following action on the computational basis states $\{|x\rangle : x \in \mathbb{Z}/2N\}$:

$$G_j|x\rangle := g_j(x)|x\rangle. \tag{2}$$

We say the algorithm is *exact* if $\langle\phi_j|\phi_{j'}\rangle = \delta_{j,j'}$ for all $j, j' \in \{0, 1, \ldots, N-1\}$, since in this case there is some measurement that can determine the result $j \bmod N$ with certainty. For each value of N, our goal is to find choices of $|\psi_0\rangle$ and $U_1, U_2, \ldots, U_k$ for k as small as possible so that the resulting quantum algorithm is exact.

The search for a good quantum algorithm for the OSP can be considerably simplified by exploiting the translation equivariance of the function g_j [3]. This equivariance manifests itself as a symmetry of the query operators. In terms of the translation operator T defined by

$$T|x\rangle := |x+1\rangle \quad \forall\, x \in \mathbb{Z}/2N \tag{3}$$

(where addition is again performed in $\mathbb{Z}/2N$), we have

$$T G_j T^{-1} = G_{j+1} \quad \forall\, j \in \mathbb{Z}/2N. \tag{4}$$

Thus, it is natural to choose the quantum algorithm to have the translation invariant initial state

$$|\psi_0\rangle = \frac{1}{\sqrt{2N}} \sum_{x=0}^{2N-1} |x\rangle \tag{5}$$

satisfying $T|\psi_0\rangle = |\psi_0\rangle$, and translation invariant unitary operations U_t, *i.e.*, unitary operators satisfying

$$T U_t T^{-1} = U_t \tag{6}$$

for $t \in \{1, 2, \ldots, k\}$. Of course, while (4) holds for all $j \in \mathbb{Z}/2N$, we are promised that $j \in \{0, 1, \ldots, N-1\}$. Correspondingly, we can require the N possible orthogonal final states to label the location of the marked item as follows:

$$|\phi_j\rangle := \begin{cases} \frac{1}{\sqrt{2}}(|j\rangle + |j+N\rangle) & k \text{ even} \\ \frac{1}{\sqrt{2}}(|j\rangle - |j+N\rangle) & k \text{ odd} \end{cases} \tag{7}$$

(where the separation into k even and odd is done for reasons explained in [3]). Overall, we refer to an algorithm with the initial state (5), unitary operations satisfying (6), and the final states (7) as an *exact, translation invariant algorithm* (in the sense of [3]).

B. Characterizing algorithms by polynomials

One of the main advantages of translation-invariant quantum algorithms for the OSP is that they have a convenient characterization in terms of univariate Laurent polynomials. A *Laurent polynomial* is a function $Q : \mathbb{C} \to \mathbb{C}$ that can be written as

$$Q(z) = \sum_{i=-D}^{D} q_i z^i \tag{8}$$

for some positive integer D, where each $q_i \in \mathbb{C}$. We call D the *degree* of $Q(z)$. We say $Q(z)$ is *nonnegative* if, on the unit circle $|z| = 1$, $Q(z)$ is real-valued and satisfies $Q(z) \geq 0$. Note that for $|z| = 1$, $z^* = z^{-1}$, so $Q(z)$ is real-valued on the unit circle if and only if $q_i = q_{-i}^*$ for all $i \in \{0, 1, \ldots, D\}$. If $Q(z) = Q(z^{-1})$ for all $z \in \mathbb{C}$, *i.e.*, if $q_i = q_{-i}$ for all $i \in \{1, 2, \ldots, D\}$, we say $Q(z)$ is *symmetric*. Thus, $Q(z)$ is nonnegative and symmetric if and only if $q_i = q_{-i} \in \mathbb{R}$ for all $z \in \{0, 1, \ldots, D\}$. An example of a nonnegative, symmetric Laurent polynomial that is relevant to the ordered search problem is the *Hermite kernel* of degree $N-1$,

$$H_N(z) := \sum_{i=-(N-1)}^{N-1} \left(1 - \frac{|i|}{N}\right) z^i \tag{9}$$

$$= \frac{1}{N} \left(\frac{z^{-N}-1}{z^{-1}-1}\right) \left(\frac{z^N - 1}{z-1}\right). \tag{10}$$

The following result of Farhi, Goldstone, Gutmann, and Sipser characterizes exact translation invariant algorithms for the ordered search problem in terms of Laurent polynomials.

Theorem 1 ([3]): There exists an exact, translation invariant, k-query quantum algorithm for the N-element OSP if and only if there exist nonnegative, symmetric Laurent polynomials $Q_0(z), \ldots, Q_k(z)$ of degree $N-1$ such that

$$Q_0(z) = H_N(z) \tag{11}$$

$$Q_t(z) = Q_{t-1}(z) \quad \text{at } z^N = (-1)^t \quad \forall\, t \in \{1, 2, \ldots, k\} \tag{12}$$

$$Q_k(z) = 1 \tag{13}$$

$$\frac{1}{2\pi} \int_0^{2\pi} Q_t(e^{i\omega})\, d\omega = 1 \qquad \forall\, t \in \{0, 1, \ldots, k\}. \tag{14}$$

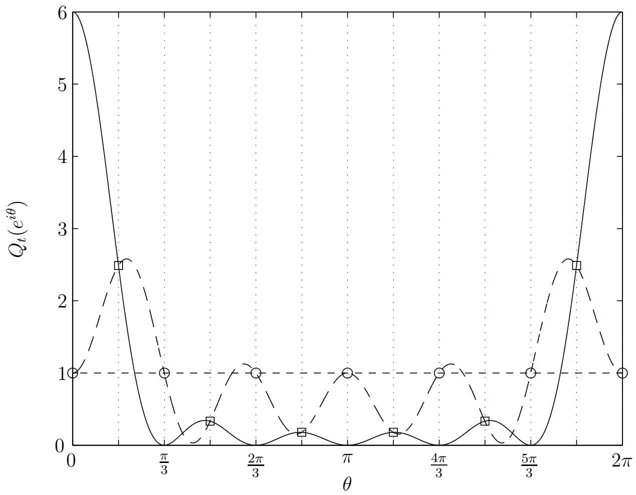

Fig. 1. $Q_t(e^{i\theta})$ as a function of θ for $k = 2$ and $N = 6$. The solid, long dashed, and short dashed lines represent $t = 0$, 1, and 2, respectively. The intersections at roots of 1 and -1 are indicated by circles and squares, respectively.

Each polynomial $Q_t(z)$ in this theorem represents the quantum state of the algorithm after t queries. Indeed, if we write

$$Q_t(z) = \sum_{i=-(N-1)}^{N-1} q_i^{(t)} z^i, \tag{15}$$

then

$$q_i^{(t)} = 2 \sum_{m=1}^{N-i} \langle \psi_t | N - m \rangle \langle N - m - i | \psi_t \rangle, \tag{16}$$

where

$$|\psi_t\rangle := U_t G_0 U_{t-1} \dots U_1 G_0 |\psi_0\rangle \tag{17}$$

is the state of the quantum computer after t queries when the target item is $j = 0$ [3]. Given polynomials satisfying (11–14), one can reconstruct all of the unitary operators U_t for the algorithm using (16).

Figure 1 shows the (unique) solution to (11–14) for $k = 2$ and $N = 6$ [3]. In general, the polynomial $Q_0(z)$ (the Hermite kernel) characterizes complete ignorance of the target location at the beginning of the algorithm, and subsequent polynomials become flatter and flatter until the final polynomial $Q_k(z) = 1$ is reached, corresponding to exact knowledge of the target location. Because each query can only change the quantum state in a restricted way, successive polynomials must agree at certain roots of ± 1. Also, each polynomial must be nonnegative and suitably normalized.

With $k = 2$, there is a unique choice for the polynomial $Q_1(z)$, which might or might not be nonnegative depending upon the value of N. For $N \leq 6$, this polynomial is nonnegative (showing that an ordered list of size $N \leq 6$ can be searched in two quantum queries), whereas for $N \geq 7$, it is not [3].

The best ordered search algorithm discovered by Farhi *et al.* was found by considering $k = 3$ queries. For fixed values of the degree $N - 1$, they numerically searched for polynomials $Q_1(z), Q_2(z)$ satisfying the constraints (11–14) of Theorem 1. The largest value of N for which they found a solution was $N = 52$. Applying this 52-item ordered search algorithm recursively gives an algorithm for instances with N arbitrarily large. Specifically, one divides the list into 52 sublists and applies the algorithm to the largest (rightmost) item of each sublist, finding the sublist that contains the target in 3 queries. This process repeats, with every 3 queries dividing the problem size by 52, leading to a query complexity of $3\lceil \log_{52} N \rceil$. (Note that although the base algorithm in this recursion is translation invariant, the scalable algorithm generated in this way is not.)

In general, recursion can be used to turn small base cases into scalable algorithms, so improved quantum algorithms for the OSP can be found by discovering improved base cases. Subsequent work by one of us (AJL) and collaborators sought such algorithms using a conjugate gradient descent search for the polynomials $Q_t(z)$ [7]. This method is guaranteed to work (for a small enough step size) because the space of polynomials satisfying (11–14) is convex. The best solutions found by this method were $N = 56$ for $k = 3$ and $N = 550$ for $k = 4$, implying a $4 \log_{550} N \approx 0.439 \log_2 N$ query recursive algorithm. Unfortunately, conjugate gradient descent (or any approach based on local optimization) can never prove that finite instance algorithms do not exist for a given number of queries k. It could always be the case that lack of progress by a solver is indicative of inadequacies of the solver (*e.g.*, the step size is too large, etc.). In the next section, we recharacterize exact translation invariant quantum OSP algorithms in a way that allows either their existence or nonexistence (whichever the case may be) to be proved efficiently.

III. A SEMIDEFINITE PROGRAM FOR TRANSLATION INVARIANT QUANTUM ALGORITHMS FOR THE OSP

A. Formulation of the SDP

In this section, we show that the problem of finding Laurent polynomials satisfying the conditions of Theorem 1 can be viewed as an instance of a particular kind of convex optimization problem, namely a *semidefinite program* [8]. The basic idea is to use the spectral factorization of nonnegative Laurent polynomials to rewrite equations (11–14) as linear constraints on positive semidefinite matrices. This property can be interpreted as providing a sum of squares representation for a nonnegative Laurent (or trigonometric) polynomial.

The spectral factorization of nonnegative Laurent polynomials follows from the Fejér-Riesz theorem:

Theorem 2 ([9], [10]): Let $Q(z)$ be a Laurent polynomial of degree D. Then $Q(z)$ is nonnegative if and only if there exists a polynomial $P(z) = \sum_{i=0}^{D} p_i z^i$ of degree D such that $Q(z) = P(z) P(1/z^*)^*$.

Let Tr_i denote the trace along the ith super-diagonal (or $(-i)$th sub-diagonal, for $i < 0$), *i.e.*, for an $N \times N$ matrix

X,

$$\operatorname{Tr}_i X = \begin{cases} \sum_{\ell=1}^{N-i} X_{\ell,\ell+i} & i \geq 0 \\ \sum_{\ell=1}^{N+i} X_{\ell-i,\ell} & i < 0. \end{cases} \tag{18}$$

The Fejér-Riesz theorem can be used to express nonnegative Laurent polynomials in terms of positive semidefinite matrices, as shown by the following lemma.

Lemma 1: Let $Q(z) = \sum_{i=-(N-1)}^{N-1} q_i z^i$ be a Laurent polynomial of degree $N-1$. Then $Q(z)$ is nonnegative if and only if there exists an $N \times N$ Hermitian, positive semidefinite matrix Q such that $q_i = \operatorname{Tr}_i Q$.

Proof: The "if" direction follows from the representation

$$Q(z) = \begin{bmatrix} 1 & \cdots & z^{-(N-1)} \end{bmatrix} Q \begin{bmatrix} 1 \\ \vdots \\ z^{N-1} \end{bmatrix}. \tag{19}$$

This $Q(z)$ is real on $|z| = 1$ since $Q = Q^\dagger$; it is nonnegative there because Q is positive semidefinite.

The converse follows from the spectral factorization of $Q(z)$. Let $Q(z) = P(z)P(1/z^*)^*$, let $\mathbf{p} := \begin{bmatrix} p_0 & \cdots & p_{N-1} \end{bmatrix}^T$, and let $\mathbf{z} := \begin{bmatrix} 1 & \cdots & z^{N-1} \end{bmatrix}^T$. Then $P(z) = \mathbf{p}^T \mathbf{z}$, and $Q(z) = \mathbf{z}^\dagger \mathbf{p}^* \mathbf{p}^T \mathbf{z}$ on $|z| = 1$. We choose $Q := \mathbf{p}^* \mathbf{p}^T$, which by construction is Hermitian and positive semidefinite. Furthermore, since $Q(z)$ on $|z| = 1$ determines the coefficients q_i, we have $q_i = \operatorname{Tr}_i Q$. ∎

Because the Laurent polynomials in Theorem 1 are not only nonnegative but also *symmetric*, we can restrict the associated matrices to be real symmetric, as the following lemma shows.

Lemma 2: If $Q(z)$ is a nonnegative, symmetric Laurent polynomial, then the matrix Q in Lemma 1 can be chosen to be real and symmetric without loss of generality.

Proof: Let Q be a Hermitian, positive semidefinite matrix such that $Q(z) = \mathbf{z}^\dagger Q \mathbf{z}$ on $|z| = 1$, where $\mathbf{z}$ is defined as in the proof of Lemma 1. Then the symmetry $Q(z) = Q(z^{-1})$ implies that $Q(z) = \mathbf{z}^\dagger Q^T \mathbf{z}$ on $|z| = 1$, and by averaging these two expressions, we have $Q(z) = \mathbf{z}^\dagger \tilde{Q} \mathbf{z}$ on $|z| = 1$, where $\tilde{Q} := (Q + Q^T)/2$ is real and symmetric. ∎

Using Lemma 2, we can recast the conditions (11–14) of Theorem 1 as the following *semidefinite program*:

Semidefinite Program 1 ($S(k, N)$): Find real symmetric positive semidefinite $N \times N$ matrices $Q_0, Q_1, \ldots, Q_k$ satisfying

$$Q_0 = E/N \tag{20}$$

$$\mathcal{T}_t Q_t = \mathcal{T}_t Q_{t-1} \qquad \forall t \in \{1, 2, \ldots, k\} \tag{21}$$

$$Q_k = I/N \tag{22}$$

$$\operatorname{Tr} Q_t = 1 \qquad \forall t \in \{0, 1, \ldots, k\} \tag{23}$$

where E is the $N \times N$ matrix in which every element is 1 and $\mathcal{T}_t : \mathcal{S}^N \to \mathbb{R}^{N-1}$ is a linear operator (on the space $\mathcal{S}^N$ of real symmetric $N \times N$ matrices) that computes signed traces along the (off-) diagonals, namely

$$(\mathcal{T}_t X)_i := \operatorname{Tr}_i X + (-1)^t \operatorname{Tr}_{i-N} X \tag{24}$$

for $i \in \{1, 2, \ldots, N-1\}$.

The existence of an exact, translation invariant quantum algorithm for the OSP is equivalent to the existence of a solution to this semidefinite program, which can be seen as follows:

Theorem 3: There exists an exact, translation invariant, k-query quantum algorithm for the N-element OSP if and only if $S(k, N)$ has a solution.

Proof: Given $Q_0, Q_1, \ldots, Q_k$ satisfying $S(k, N)$, let $Q_j(z) := \begin{bmatrix} 1 & \cdots & z^{-(N-1)} \end{bmatrix} Q_j \begin{bmatrix} 1 & \cdots & z^{N-1} \end{bmatrix}^T$. Then the symmetry of each matrix Q_j implies that each $Q_j(z)$ is a nonnegative, symmetric Laurent polynomial; and conditions (20–23) imply conditions (11–14), respectively.

Conversely, suppose $Q_0(z), Q_1(z), \ldots, Q_k(z)$ are nonnegative, symmetric Laurent polynomials of degree $N - 1$ satisfying (11–14). Let $Q_0 := E/N$, let $Q_k := I/N$, and let $Q_1, Q_2, \ldots, Q_{k-1}$ be positive semidefinite matrices obtained from $Q_1(z), Q_2(z), \ldots, Q_{k-1}(z)$ according to Lemma 2. Then (12) and (14) imply (21) and (23), respectively. ∎

This reformulation of the problem has the advantage that semidefinite programs are a well-studied class of convex optimization problems. In fact, semidefinite programming feasibility problems can be solved (modulo some minor technicalities) in polynomial time [8], [11]. Furthermore, there are several widely available software packages for solving semidefinite programs [12], [13], [14].

Note that by "solving" a semidefinite program, we mean not only finding a solution if one exists, but also generating an *infeasibility certificate* (namely, a solution to the dual semidefinite program) if one does not. Thus, by solving $S(k, N)$ for various values of k and N, not only can we extract algorithms from feasible solutions, but we can also generate lower bounds for the quantum query complexity of the OSP (assuming we restrict our attention to exact, translation invariant algorithms). In other words, this approach unifies algorithm design and lower bound analysis into a single method.

B. Improved formulation by symmetry reduction

In moving from the polynomial to the semidefinite programming formulation, we have increased the number of real parameters specifying an exact, translation invariant quantum OSP algorithm from $(N-1)(k-1)$ to $N(N+1)(k-1)/2$. As benefits, we have put the problem in a numerically tractable form, and we are now able to prove nonexistence as well as existence of algorithms. But the increase in parameters is nevertheless undesirable.

Fortunately, in our case we can reduce the size of the parameter set roughly by half by exploiting symmetry. In particular, in terms of the $N \times N$ counterdiagonal matrix (the *counteridentity matrix*)

$$J := \begin{bmatrix} 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \iddots & \vdots \\ 0 & 1 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \end{bmatrix}, \tag{25}$$

we have

Lemma 3: If $Q_0, Q_1, \ldots, Q_k$ is a solution to $S(k, N)$, then so is $JQ_0J, JQ_1J, \ldots, JQ_kJ$.

Proof: The matrices JQ_tJ are positive semidefinite since J is unitary. Clearly, $JQ_0J = Q_0$ and $JQ_kJ = Q_k$, so (20) and (22) are satisfied. Since $\operatorname{Tr}_i JQ_tJ = \operatorname{Tr}_{-i} Q_t$ by the definition of J, and since $\operatorname{Tr}_{-i} Q_t = \operatorname{Tr}_i Q_t$ because each Q_t is a symmetric matrix, (22) is satisfied. Finally, (23) is satisfied since $J^2 = I$. ■

Thus, by convexity, if $Q_0, Q_1, \ldots, Q_k$ is a solution to $S(k, N)$ then so is $\frac{1}{2}(Q_0 + JQ_0J), \frac{1}{2}(Q_1 + JQ_1J), \ldots, \frac{1}{2}(Q_k + JQ_kJ)$. In other words, we can assume that the matrices Q_t commute with J without loss of generality.

Due to this commutation property, we can block-diagonalize each Q_t into two blocks, each of which has roughly one quarter the number of elements (depending on the parity of N). For example, for N even, $Q = JQJ$ implies that Q has the form

$$Q = \begin{bmatrix} A & B \\ JBJ & JAJ \end{bmatrix} \tag{26}$$

where $A = A^T$ and $B = JB^TJ$. Thus we have

$$U^\dagger Q U = \frac{1}{2} \begin{bmatrix} I & I \\ J & -J \end{bmatrix}^T \begin{bmatrix} A & B \\ JBJ & JAJ \end{bmatrix} \begin{bmatrix} I & I \\ J & -J \end{bmatrix} \tag{27}$$

$$= \begin{bmatrix} A + BJ & 0 \\ 0 & A - BJ \end{bmatrix}, \tag{28}$$

so that Q is positive semidefinite if and only if $A \pm BJ$ are both positive semidefinite. The net effect of this symmetry reduction is to cut the number of real parameters in $S(k, N)$ to $N(N/2+1)(k-1)/2$ (for N even) or $(N+1)^2(k-1)/4$ (for N odd), *i.e.*, roughly by half.

C. Duality

The formulation of the search for algorithms as a semidefinite programming problem has other important consequences, besides computational tractability. In particular, it enables the use of *duality* methods, in order to certify the inexistence of algorithms satisfying certain performance requirements (i.e., complexity lower bounds). In the specific case of the formulation discussed, we have the following result:

Theorem 4: Let $\mathcal{T}_t^* : \mathbb{R}^{N-1} \to \mathcal{S}^N$ be the adjoint of $\mathcal{T}_t$ (as defined in (24)). If there exist $z_1, \ldots, z_k \in \mathbb{R}^{N-1}$ and $\lambda_1, \ldots, \lambda_{k-1} \in \mathbb{R}$ such that

$$\operatorname{Tr} Q_0 \mathcal{T}_1^*(z_1) - \operatorname{Tr} Q_k \mathcal{T}_k^*(z_k) > \sum_{t=1}^{k-1} \lambda_t \tag{29}$$

$$\mathcal{T}_t^*(z_t) - \mathcal{T}_{t+1}^*(z_{t+1}) \preceq \lambda_t I, \qquad t = 1, \ldots, k-1$$

then no k-query translationally invariant quantum algorithm exists.

Proof: The statement follows directly from semidefinite programming (weak) duality. For simplicity of notation, let $T_t = \mathcal{T}_t^*(z_t)$. Then, for any primal feasible Q_t and dual feasible λ_t, z_t, we have

$$\begin{aligned} &\operatorname{Tr} Q_0 T_1 - \operatorname{Tr} Q_k T_k \le \\ &\operatorname{Tr} Q_0 T_1 + \sum_{t=1}^{k-1} \operatorname{Tr} Q_t \cdot (\lambda_t I - T_t + T_{t+1}) - \operatorname{Tr} Q_k T_k = \\ &\qquad \sum_{t=1}^{k-1} \lambda_t + \sum_{t=1}^{k} \langle z_t, \mathcal{T}_t(Q_{t-1} - Q_t) \rangle = \sum_{t=1}^{k-1} \lambda_t, \end{aligned}$$

which clearly yields a contradiction with (29). ■

An appealing interpretation of the dual is that it provides a sequence $\{\mathcal{T}_t^*(z_t)\}$ of (possibly indefinite) metrics on the states, with their consecutive differences bounded by λ_t. Since the total variation between the given initial and final state is bounded below by the sum of the λ_t, no k-query operation algorithm transforming these pair of states can possibly exist.

k	N^*
2	6
3	56
4	605
5	> 5000

TABLE I

ORDERED LIST SIZES N^* SEARCHABLE A k-QUERY EXACT, TRANSLATION INVARIANT QUANTUM ALGORITHM SUCH THAT NO SUCH ALGORITHM EXISTS FOR A LIST OF SIZE $N^* + 1$.

IV. RESULTS

We solved the semidefinite program $S(k, N)$ for various values of k and N using the numerical solvers SeDuMi [12], SDPT3 [13], and SDPA [14]. These solvers use general-purpose primal-dual interior-point methods that eventually become limited by machine memory. (Although there are algorithms for solving SDPs that are not based on interior point methods, we did not attempt to use such algorithms.)

The time required to solve $S(k, N)$ was substantially reduced by exploiting the symmetry described in Section III-B. In addition, it is helpful that the constraints are fairly sparse. Nevertheless, we are ultimately limited by the fact that the maximum size of a list that can be searched increases exponentially with the number of queries, so that we can only consider fairly small values of k.

For each $k \le 4$, we found the smallest value N^* such that $S(k, N^*)$ has a solution but $S(k, N^*+1)$ does not. Although we were able to find solutions to $S(5, N)$ for some values of N, we ran out of machine memory before we could find an infeasibility certificate. A summary of the values N^* we obtained is presented in Table I.

By recursion, the $k = 4$, $N^* = 605$ query algorithm yields a scalable algorithm whose query complexity is

$$4 \log_{605} N \approx 0.433 \log_2 N. \tag{30}$$

This result also implies improvements to other algorithms; for example, it implies a quantum sorting algorithm whose query complexity is $4N \log_{605} N$.

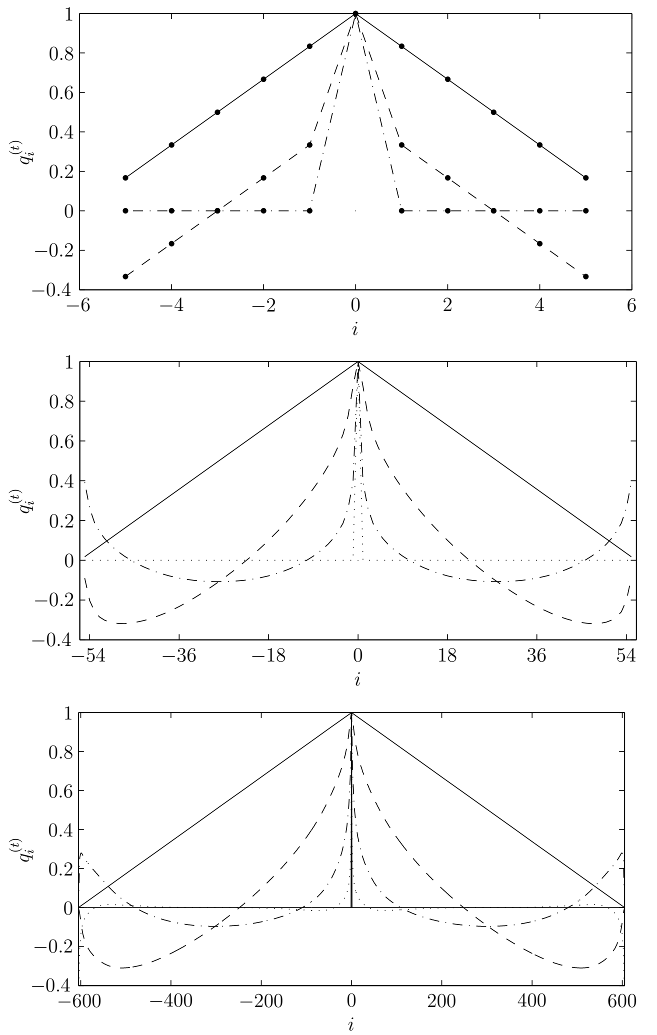

Fig. 2. Laurent polynomial coefficients $q_i^{(t)}$ as a function of i for exact, translation invariant OSP algorithms. From top to bottom, $k = 2$, 3, and 4, with $N = 6$, 56, and 605, respectively.

As mentioned in the introduction, infeasibility of $S(k, N^* + 1)$ does not necessarily imply that N^* is the largest size of a list that can be searched with a k-query exact, translation invariant algorithm. However, it seems reasonable to conjecture that this might be the case. Indeed, for $k = 2$ and 3, we have shown that the values of N^* in Table I are optimal; see [6] for details. Those results show that $N = 6$ and $N = 56$ are the largest sizes of lists that can be searched with $k = 2$ and $k = 3$ queries, respectively, even when the assumption of translation invariance is removed.

Whether the $\frac{1}{\pi} \ln N$ lower bound on the query complexity of the OSP can be saturated remains open. However, the structure of the algorithms we obtained suggests the possibility of a well-behaved analytic solution, and it would be interesting to understand the behavior of the solution in the limit of large N. Figure 2 shows the coefficients of the polynomials $Q_t(z)$ associated with the optimal feasible solutions Q_t for $k = 2, 3, 4$. Note the similarity of the coefficients for different values of N.

Very recently, Ben-Or and Hassidim [15] have developed an approach to quantum algorithms for ordered search based on adaptive learning. Their resulting algorithm is not exact, but rather is zero error, with a stochastic running time (sometimes referred to as a Las Vegas algorithm). The expected running time of their algorithm is $0.32 \log_2 N$.

V. Conclusions

We have shown the applicability of semidefinite programming to the effective search for optimal quantum algorithms. Of additional interest, given the intimate connections between spectral factorization, sum of squares, and Riccati equations, would be the possibility of a Riccati-based solution for this problem.

In particular, we have found a particular quantum algorithm for the ordered search problem, with performance superior to all known exact algorithms. We remark that the connection between quantum query complexity and convex optimization is not unique to the ordered search problem: arbitrary quantum query problems can be characterized in terms of semidefinite programs [16]. Thus, semidefinite programming appears to be a very powerful tool for the numerical and theoretical study of the complexity of quantum algorithms.

References

[1] D. E. Knuth, *The Art of Computer Programming, Volume 3: Sorting and Searching*, 2nd ed. Upper Saddle River, NJ: Addison Wesley, 1998.

[2] P. Høyer, J. Neerbek, and Y. Shi, "Quantum complexities of ordered searching, sorting, and element distinctness," *Algorithmica*, vol. 34, no. 4, pp. 429–448, 2002.

[3] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, "Invariant quantum algorithms for insertion into an ordered list," 1999, MIT CTP #2815.

[4] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, 1996, pp. 212–219.

[5] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, "Strengths and weaknesses of quantum computing," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1510–1523, 1997.

[6] A. M. Childs, A. J. Landahl, and P. A. Parrilo, "Quantum algorithms for the ordered search problem via semidefinite programming," *Phys. Rev. A*, vol. 75, p. 032335, 2007.

[7] E. M. Brookes, M. B. Jacokes, and A. J. Landahl, "An improved quantum algorithm for searching an ordered list," 2004, unpublished.

[8] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Review*, vol. 38, no. 1, pp. 49–95, 1996.

[9] L. Fejér, "Über trigonometriche Polynome," *J. Reine Agnew. Math.*, vol. 146, pp. 52–82, 1915.

[10] F. Riesz, "Über ein Problem des Herr Carathéodory," *J. Reine Agnew. Math.*, vol. 146, pp. 83–87, 1915.

[11] L. Vandenberghe and S. Boyd, *Convex Optimization*. Cambridge: Cambridge University Press, 2004.

[12] J. Sturm, *SeDuMi version 1.05*, 2001, available from http://sedumi.mcmaster.ca.

[13] K. C. Toh, R. H. Tütüncü, and M. J. Todd, *SDPT3 version 3.02*, 2002, available from http://www.math.nus.edu.sg/ mattohkc/sdpt3.html.

[14] *SDPA version 6.2.0*, 2005, available from http://grid.r.dendai.ac.jp/sdpa/publication.html.

[15] M. Ben-Or and A. Hassidim, "Quantum search in an ordered list via adaptive learning," 2007, available at `arXiv:quant-ph/0703231`.

[16] H. Barnum, M. Saks, and M. Szegedy, "Quantum query complexity and semi-definite programming," in *Proceedings of the 18th IEEE International Conference on Computational Complexity*, 2003, pp. 179–193.