Quantum query complexity of minor-closed graph properties*

Andrew M. Childs[1] and Robin Kothari[2]

1 Department of Combinatorics & Optimization
and Institute for Quantum Computing
University of Waterloo
amchilds@uwaterloo.ca

2 David R. Cheriton School of Computer Science
and Institute for Quantum Computing
University of Waterloo
rkothari@cs.uwaterloo.ca

Abstract

We study the quantum query complexity of minor-closed graph properties, which include such problems as determining whether an n-vertex graph is planar, is a forest, or does not contain a path of a given length. We show that most minor-closed properties—those that cannot be characterized by a finite set of forbidden subgraphs—have quantum query complexity $\Theta(n^{3/2})$. To establish this, we prove an adversary lower bound using a detailed analysis of the structure of minor-closed properties with respect to forbidden topological minors and forbidden subgraphs. On the other hand, we show that minor-closed properties (and more generally, sparse graph properties) that can be characterized by finitely many forbidden subgraphs can be solved strictly faster, in $o(n^{3/2})$ queries. Our algorithms are a novel application of the quantum walk search framework and give improved upper bounds for several subgraph-finding problems.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Quatum query complexity, quantum algorithms, lower bounds, graph minors, graph properties

Digital Object Identifier 10.4230/LIPIcs.STACS.2011.661

1 Introduction

The decision tree model is a simple model of computation for which we can prove good upper and lower bounds. Informally, decision tree complexity, also known as query complexity, counts the number of input bits that must be examined by an algorithm to evaluate a function. In this paper, we focus on the query complexity of deciding whether a graph has a given property. The query complexity of graph properties has been studied for almost 40 years, yet old and easy-to-state conjectures regarding the deterministic and randomized query complexities of graph properties [11, 15, 20, 23] remain unresolved.

The study of query complexity has also been quite fruitful for quantum algorithms. For example, Grover's search algorithm [13] operates in the query model, and Shor's factoring algorithm [24] is based on the solution of a query problem. However, the quantum query complexity can be harder to pin down than its classical counterparts. For monotone graph properties, a wide class of graph properties including almost all the properties considered

* This work was supported in part by MITACS, NSERC, QuantumWorks, and the US ARO/DTO.

28th Symposium on Theoretical Aspects of Computer Science (STACS'11).
Editors: Thomas Schwentick, Christoph Dürr; pp. 661–672
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
SYMPOSIUM ON THEORETICAL ASPECTS OF COMPUTER SCIENCE

in this paper, the widely-believed Aanderaa–Karp–Rosenberg conjecture states that the deterministic and randomized query complexities are $\Theta(n^2)$, where n is the number of vertices. On the other hand, there exist monotone graph properties whose quantum query complexity is $\Theta(n)$, and others with query complexity $\Theta(n^2)$. In fact, one can construct a monotone graph property with quantum query complexity $\Theta(n^{1+\alpha})$ for any fixed $0 \leq \alpha \leq 1$ using known bounds for the threshold function [6].

The quantum query complexity of several specific graph properties has been established in prior work. Dürr, Heiligman, Høyer, and Mhalla [12] studied the query complexity of several graph problems, and showed in particular that connectivity has quantum query complexity $\Theta(n^{3/2})$. Zhang [28] showed that the quantum query complexity of bipartiteness is $\Theta(n^{3/2})$. Ambainis et al. [5] showed that planarity also has quantum query complexity $\Theta(n^{3/2})$. Berzina et al. [7] showed several quantum lower bounds on graph properties, including Hamiltonicity. Sun, Yao, and Zhang studied some non-monotone graph properties [26].

Despite this work, the quantum query complexity of many interesting graph properties remains unresolved. A well-studied graph property whose query complexity is unknown is the property of containing a triangle (i.e., a cycle on 3 vertices) as a subgraph. The triangle finding problem was first studied by Buhrman et al. [10], who gave an $O(n + \sqrt{nm})$ query algorithm for graphs with n vertices and m edges. With $m = \Theta(n^2)$, this approach uses $O(n^{3/2})$ queries, which matches the performance of the simple algorithm that searches for a triangle over the potential $\binom{n}{3}$ triplets of vertices. This was later improved by Magniez, Santha, and Szegedy [19] to $\tilde{O}(n^{1.3})$, and then by Magniez, Nayak, Roland, and Santha [18] to $O(n^{1.3})$, which is currently the best known algorithm. However, the best known lower bound for the triangle problem is only $\Omega(n)$ (by a simple reduction from the search problem). This is partly because one of the main lower bound techniques, the quantum adversary method of Ambainis [2], cannot prove a better lower bound due to the certificate complexity barrier [25, 28].

More generally, we can consider the H-subgraph containment problem, in which the task is to determine whether the input graph contains a fixed graph H as a subgraph. Magniez et al. also gave a general algorithm for H-subgraph containment using $\tilde{O}(n^{2-2/d})$ queries, where $d > 3$ is the number of vertices in H [19]. Again, the best lower bound known for H-subgraph containment is only $\Omega(n)$.

In this paper we study the quantum query complexity of minor-closed graph properties. A property is minor-closed if all minors of a graph possessing the property also possess the property. (Graph minors are defined in Section 2.) Since minor-closed properties can be characterized by forbidden minors, this can be viewed as a variant of subgraph containment in which we look for a given graph as a minor instead of as a subgraph. The canonical example of a minor-closed property is the property of being planar. Other examples include the property of being a forest, being embeddable on a fixed two-dimensional manifold, having treewidth at most k, or not containing a path of a given length.

While all minor-closed properties can be described by a finite set of forbidden minors, some minor-closed properties can also be described by a finite set of forbidden subgraphs, graphs that do not appear as a subgraph of any graph possessing the property. We call a graph property (which need not be minor closed) a *forbidden subgraph property* (FSP) if the property can be described by a finite set of forbidden subgraphs.

Our main result is that the quantum query complexity of minor-closed properties depends crucially on whether the property is FSP. In particular, Figure 1 summarizes our understanding of the quantum query complexity of minor-closed graph properties. All subgraph-closed properties, which include minor-closed properties and FSPs, have an easy lower bound

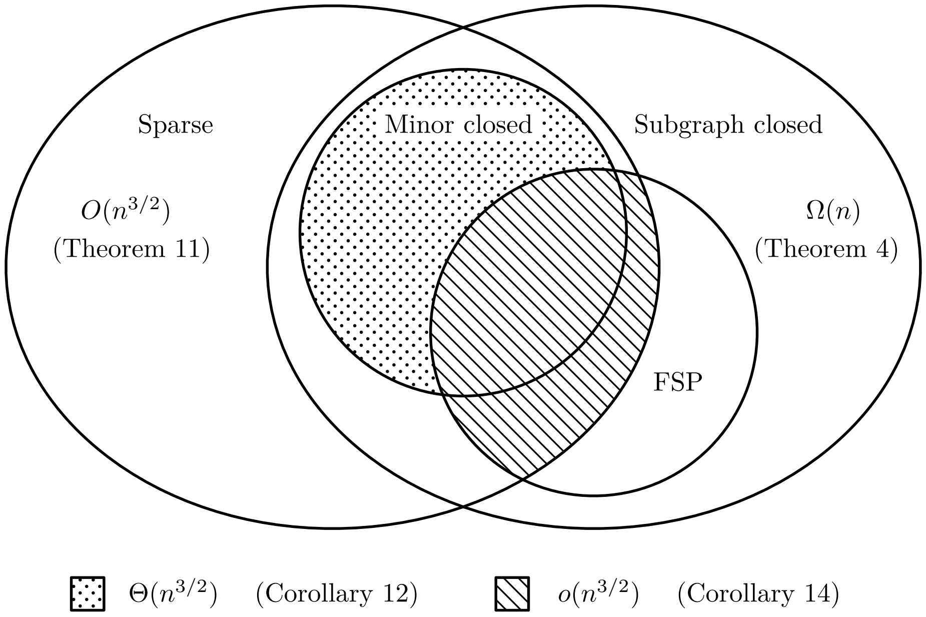

Figure 1 Summary of the main results.

of of $\Omega(n)$ (Theorem 4). Furthermore, all sparse graph properties, which are defined in Section 2 and which include all minor-closed properties, have an easy upper bound of $O(n^{3/2})$ (Theorem 11). On the lower bound side, our main contribution is to show that minor-closed properties that are not FSP require $\Omega(n^{3/2})$ queries (Theorem 8), which tightly characterizes their quantum query complexity. Regarding upper bounds, our main contribution is a quantum algorithm for all sparse graph properties that are FSP, using $O(n^{\alpha})$ queries for some $\alpha < 3/2$ that depends on the property (Corollary 14).

Our lower bounds (Section 3) use the quantum adversary method [2]. The basic idea of the lower bound is similar to the connectivity lower bound of Dürr et al. [12]. However, it is nontrivial to show that this approach applies using only the hypothesis that the property is minor-closed and not FSP. In fact, we show a slightly stronger result, assuming only that the property is not FSP and can be described by finitely many forbidden topological minors.

Our upper bounds (Section 4) use the quantum walk search formalism [18]. Our approach differs from previous applications of this formalism in several respects. We use several quantum walks occurring simultaneously on different Hamming graphs (whereas most previous quantum walk search algorithms used a single Johnson graph). Although this can be viewed as a single walk on a larger graph, the salient feature is that the walks on different graphs proceed at different speeds, i.e., in each time step a different number of steps are taken on each graph. In addition, we make essential use of the sparsity of the input graph.

By exploiting sparsity, we also improve upon known algorithms for many sparse graph properties, even if they are not necessarily minor closed. For example, we give improved algorithms for finding paths of a given length, as well as an algorithm that outperforms the general H-finding algorithm of Magniez et al. [19] whenever H is a bipartite graph.

Finally, as another application, we consider the C_4-subgraph containment problem. This can be viewed as a natural extension of the triangle problem, which is C_3-subgraph containment. Surprisingly, we show that C_4 finding can be solved with only $\tilde{O}(n^{1.25})$ queries, even faster than the best known upper bound for triangle finding, which is $O(n^{1.3})$.

2 Preliminaries

In this paper, all graphs are simple and undirected. Thus a graph on n vertices is specified by $\binom{n}{2}$ bits. In the query complexity model, the input graph is accessed by querying a black box to learn any of these $\binom{n}{2}$ bits. Deterministic and randomized algorithms have access to a black box taking two inputs, u and v, and returning a bit indicating whether (u, v) is an edge in the graph. To accommodate quantum algorithms, we define a quantum black box in the standard way. The quantum black box is a unitary gate that maps $|u, v, b\rangle$ to $|u, v, b \oplus e\rangle$ where $(u, v) \in V \times V$, b is a bit, and e is 1 if and only if $(u, v) \in E$.

Let the deterministic, randomized, and quantum query complexities of determining whether a graph possesses property $\mathcal{P}$ be denoted as $D(\mathcal{P})$, $R(\mathcal{P})$, and $Q(\mathcal{P})$, respectively, where for R and Q we consider two-sided bounded error. Clearly, $Q(\mathcal{P}) \leq R(\mathcal{P}) \leq D(\mathcal{P}) \leq \binom{n}{2}$. Also note that these query complexities are the same for a property $\mathcal{P}$ and its complement $\bar{\mathcal{P}}$, since any algorithm for $\mathcal{P}$ can be turned into an algorithm for $\bar{\mathcal{P}}$ by negating the output, using no additional queries.

A graph property on n vertices is a property of n-vertex graphs that is independent of vertex labeling, i.e., isomorphic graphs are considered equivalent. For a graph G on n vertices and an n-vertex graph property $\mathcal{P}_n$, we write $G \in \mathcal{P}_n$ to mean that graph G has property $\mathcal{P}_n$. A graph property $\mathcal{P} := \{\mathcal{P}_n\}_{n=1}^{\infty}$ is a collection of n-vertex graph properties $\mathcal{P}_n$ for all $n \in N$. For example, the property "the first vertex is isolated" is not a graph property because it depends on the labeling, and in particular it depends on which vertex we decide to call the first one. However, the property "contains an isolated vertex" is a graph property.

An n-vertex graph property $\mathcal{P}_n$ is nontrivial if there exists a graph that possesses it and one that does not. A graph property $\mathcal{P} = \{\mathcal{P}_n\}_{n=1}^{\infty}$ is nontrivial if there exists an n_0 such that $\mathcal{P}_n$ is nontrivial for all $n > n_0$. Thus a property such as "contains a clique of size 5" is nontrivial, although it is trivial for graphs with fewer than 5 vertices.

In this paper, K_n and C_n refer to the complete graph and cycle on n vertices, respectively. $K_{s,t}$ is the complete bipartite graph with s vertices in one part and t vertices in the other. A d-path is a path with d edges (i.e., with $d + 1$ vertices). For a graph G, $V(G)$ and $E(G)$ denote the vertex and edge sets of the graph; $n := |V(G)|$ and $m := |E(G)|$.

A graph H is said to be a subgraph of G, denoted $H \leq_S G$, if H can be obtained from G by deleting edges and isolated vertices. A graph H is said to be a minor of G, denoted $H \leq_M G$, if H can be obtained from G by deleting edges, deleting isolated vertices, and contracting edges. To contract an edge (u, v), we delete the vertices u and v (and all associated edges) and create a new vertex that is adjacent to all the original neighbors of u and v. The name "edge contraction" comes from viewing this operation as shrinking the edge (u, v) to a point, letting the vertices u and v coalesce to form a single vertex.

Another way to understand graph minors is to consider reverse operations: $H \leq_M G$ if G can be obtained from H by adding isolated vertices, adding edges, and performing vertex splits. In a vertex split, we delete a vertex u and add two new adjacent vertices v and w, such that each original neighbor of u becomes a neighbor of either v or w, or both. In general, this operation does not lead to a unique graph, since there may be many different ways to split a vertex.

A related operation, which is a special case of a vertex split, is known as an elementary subdivision. This operation replaces an edge (u, v) with two edges (u, w) and (w, v), where w is a new vertex. A graph H is said to be a topological minor of G, denoted $H \leq_T G$, if G can be obtained from H by adding edges, adding isolated vertices, and performing elementary subdivisions. We call G a subdivision of H if it is obtained from H by performing any number of elementary subdivisions.

Some graph properties can be expressed using a forbidden graph characterization. Such a characterization says that graphs have the property if and only if they do not contain any of some set of forbidden graphs according to some notion of graph inclusion, such as subgraphs or minors. For example, a graph is a forest if and only if it contains no cycle as a subgraph, so forests are characterized by the forbidden subgraph set $\{C_k : k \geq 3,\ k \in \mathbb{N}\}$. The property of being a forest can also be characterized by the single forbidden minor C_3, since a graph is a forest if and only if it does not contain C_3 as a minor. If a property can be expressed using a finite number of forbidden subgraphs, we call it a forbidden subgraph property (FSP). A property is said to be subgraph closed if every subgraph of a graph possessing the property also possess the property. Similarly, a property is said to be minor closed if all minors of a graph possessing the property also possess the property. In a series of 20 papers spanning over 20 years, Robertson and Seymour proved the following theorem [22]:

▶ **Theorem 1** (Graph minor theorem). *Every minor-closed graph property can be described by a finite set of forbidden minors.*

We also require the following consequence of the graph minor theorem, which follows using well-known facts about topological minors [21, Theorem 2.1].

▶ **Corollary 2.** *Every minor-closed graph property can be described by a finite set of forbidden topological minors.*

We call a graph property *sparse* if there exists a constant c such that every graph G with the property has $|E(G)| \leq c\,|V(G)|$. Nontrivial minor-closed properties are sparse, which is an easy corollary of Mader's theorem [17].

▶ **Theorem 3.** *Every nontrivial minor-closed graph property is sparse.*

We use $\tilde{O}$ notation to denote asymptotic upper bounds that neglect logarithmic factors. Specifically, $f(n) = \tilde{O}(g(n))$ means $f(n) = O(g(n) \log^k g(n))$ for some constant k.

3 Lower bounds

The following lower bound follows easily using the adversary methods of Ambainis [2] and Aaronson [1].

▶ **Theorem 4.** *For any nontrivial subgraph-closed graph property $\mathcal{P}$, $Q(\mathcal{P}) = \Omega(n)$, $R(\mathcal{P}) = \Theta(n^2)$, and $D(\mathcal{P}) = \Theta(n^2)$.*

With the exception of general sparse properties, this theorem covers all the properties considered in this paper, since every property (or its complement) is closed under subgraphs. Thus all the properties considered in this paper are classically uninteresting from the viewpoint of query complexity, since their classical (deterministic or randomized) query complexity is exactly $\Theta(n^2)$.

In the remainder of this section, we describe our main lower bound result: Every minor-closed property that is not FSP has $Q(\mathcal{P}) = \Omega(n^{3/2})$. We begin by considering H-topological minor containment properties that are not also H-subgraph containment properties, while describing some of the tools used to show the more general result.

As a motivating example, consider $H = C_3$. C_3-topological minor containment (which is equivalent to C_3-minor containment) is the property of being cyclic; its complementary property is that of being a forest. We show that $\Omega(n^{3/2})$ queries are required for this property.

This lower bound is similar to the connectivity lower bound of Dürr et al. [12]. The intuition is that a long path and a long cycle look the same locally. Since algorithms only have access to local information, these two graphs should be hard to distinguish. Unfortunately this is not sufficient, since a path can be easily distinguished from a cycle by searching the entire graph for a degree-1 vertex, which can be done with $O(n)$ queries. Instead, we try to distinguish a path from the disjoint union of a cycle and a path. Now both graphs have 2 degree-1 vertices. We require both the cycle and the path to be long, since a short cycle or path could be quickly traversed. Considering these instances, an adversary argument shows the following.

▶ **Theorem 5.** *Deciding if a graph is a forest requires $\Omega(n^{3/2})$ queries.*

This construction does not use any particular property of forests, except that all subdivisions of C_3 are not forests, and that if we delete an edge from a subdivision of C_3, the resulting graph is a forest. More precisely, we use the existence of a graph G (in this case C_3) and an edge $(u, v) \in E(G)$ (in this case it can be any edge) such that if (u, v) is subdivided any number of times, the resulting graph still does not have the property (in this case, of being a forest) and if (u, v) is replaced by two disjoint paths the resulting graph does have the property. The following lemma formalizes this intuition.

▶ **Lemma 6.** *Let $\mathcal{P}$ be a graph property closed under topological minors. If there exists a graph $G \notin \mathcal{P}$ and an edge (u, v) of G, such that replacing the edge (u, v) by two disjoint paths of any length, one connected to vertex u and the other connected to vertex v, always results in a graph $G' \in \mathcal{P}$, then $Q(\mathcal{P}) = \Omega(n^{3/2})$.*

It can be shown that any graph H for which H-topological minor containment does not coincide with H-subgraph containment contains such an edge, and thus we obtain an $\Omega(n^{3/2})$ lower bound in this case. In particular, a graph H satisfies this condition if and only if it is cyclic or contains 2 vertices of degree at least 3 in the same connected component, and any edge on a cycle or on a path between 2 vertices of degree at least 3 can serve as the edge (u, v) in Lemma 6.

From H-topological minor containment, we move on to properties that can be described by a finite set of forbidden topological minors. While this case is conceptually similar to the case of a single forbidden minor, it is more technically challenging. The final result, however, is easy to state:

▶ **Lemma 7.** *For any graph property $\mathcal{P}$ that is not FSP and that is described by a finite set of forbidden topological minors, there exists a graph $G \notin \mathcal{P}$ and an edge $(u, v) \in E(G)$ satisfying the conditions of Lemma 6.*

Combining this with Corollary 2 and Lemma 6, we get our main lower bound result.

▶ **Theorem 8.** *For any nontrivial minor-closed property $\mathcal{P}$ that is not FSP, $Q(\mathcal{P}) = \Omega(n^{3/2})$.*

This lower bound cannot be improved due to a matching algorithm shown in Section 4. It cannot be extended to minor-closed properties that are also FSP because, as we also show in Section 4, every property of this type has query complexity $o(n^{3/2})$.

4 Algorithms

We now turn to quantum algorithms for deciding minor-closed graph properties, as well as related algorithms for subgraph-finding problems.

4.1 Sparse graph detection and extraction

We begin by describing some basic tools that allow us to detect whether a graph is sparse and to optimally extract the adjacency matrix of a sparse graph.

To tell whether a graph is sparse, we can apply quantum counting to determine approximately how many edges it contains. In particular, Theorem 15 of [9] can be applied to show the following.

▶ **Lemma 9.** *For any constant $\epsilon > 0$ and function $f : \mathbb{Z}^+ \to \mathbb{Z}^+$ there is a quantum algorithm using $O(\sqrt{n^2/f(n)} \log \frac{1}{\delta})$ queries that accepts graphs with $m \geq (1+\epsilon)f(n)$ and rejects graphs with $m \leq (1-\epsilon)f(n)$ with probability at least $1-\delta$.*

We also use a procedure for extracting all marked items in a search problem.

▶ **Lemma 10.** *Let $f : \{1,\ldots,N\} \to \{0,1\}$ be a black-box function with $|f^{-1}(1)| = K$. The bounded-error quantum query complexity of determining $f^{-1}(1)$ is $O(\sqrt{NK})$ if $K > 0$, and $O(\sqrt{N})$ if $K = 0$.*

This result and its optimality appear to be folklore (see for example [4]).

An easy consequence of these results is that sparse graph properties can be decided in $O(n^{3/2})$ queries.

▶ **Theorem 11.** *If $\mathcal{P}$ is a sparse graph property, then $Q(\mathcal{P}) = O(n^{3/2})$.*

Combining this with Theorem 3 and Theorem 8, an immediate consequence is

▶ **Corollary 12.** *If $\mathcal{P}$ is nontrivial, minor closed, and not FSP, then $Q(\mathcal{P}) = \Theta(n^{3/2})$.*

Note that this provides an alternative proof that the quantum query complexity of planarity is $\Theta(n^{3/2})$ [5].

For minor-closed graph properties that are also FSP, the lower bounds from Section 3 do not rule out the possibility of an improvement over Theorem 11. In fact, we show that an improvement is possible for all such properties.

4.2 Quantum walk search

Our algorithms use the quantum walk search framework of Magniez et al. (Theorem 3 of [18]), which builds on the work of Ambainis [3] and Szegedy [27]. However, our application of this framework differs from previous applications in several ways.

In nearly all previous quantum walk search algorithms, the graph on which the walk occurs is the Johnson graph $J(N, K)$, whose vertices are the $\binom{N}{K}$ subsets of $\{1,\ldots,N\}$ of size K, with an edge between subsets that differ in exactly one item. The parameters N and K are chosen based on the input size. For example, the triangle finding algorithm [19] fixes $N = n$ and $K = n^{3/5}$, where n is the number of vertices in the input graph (not to be confused with the graph on which the quantum walk occurs). Each vertex of the Johnson graph has an associated data structure. Populating this data structure typically requires queries to the input. In the setup step, the walk begins in a uniform superposition over all the vertices of the Johnson graph. A step of the quantum walk corresponds to moving to one of the neighbors of the current vertex and updating the data structure; this is called the update step. Some of the vertices are designated as "marked" and the objective of the walk is to determine if there are any such vertices. Every few steps of the walk, we have a checking step which determines if the current vertex is marked.

Our walk differs from this in several ways. For our purposes it will be more convenient to replace the Johnson graph $J(N,K)$ by the Hamming graph $H(N,K)$, with vertex set $\{1,\dots,N\}^K$ and edges between two K-tuples that differ in exactly one coordinate. This choice simplifies the implementation of our setup step. Although the order of the items has no significance, and the possibility of repeated items only slows down the algorithm, the effect is not significant.

Furthermore, our algorithms involve a walk on several different Hamming graphs. Each Hamming graph may have very different value of N and K, and these values depend not only on the input size, but on the actual input itself. This means that queries are required even to decide which graph is being walked on. Moreover, the walks on different graphs may occur at different speeds.

4.3 Detecting subgraphs of sparse graphs

We now describe algorithms that determine whether a sparse graph G contains a given subgraph H. We begin with an informal overview before stating the general result.

For concreteness, consider the problem of detecting a clique of size 5 in a sparse graph. The idea is to look for the 5 vertices separately (using 5 quantum walks), and use the checking step to verify that the 5 vertices do form a clique. The data structure stores a list of the neighbors of each vertex under consideration.

First, let us assume that a clique of size 5 exists; if our algorithm fails to detect one we can conclude that our assumption was incorrect and reject the input. Second, we guess the approximate degrees of all the vertices we are looking for, up to a constant multiplicative factor. Since $\Theta(\log n)$ guesses cover the entire range between 1 and n, up to a multiplicative overhead of $\mathrm{poly}(\log n)$, we can assume that we already know the approximate degrees of all the vertices we are looking for. Let these degrees be q_i for $1 \le i \le 5$.

Then we use quantum counting to estimate the number of vertices with approximate degree q_i. Let this value be t_i. Now we set up 5 quantum walks, with the i^{th} walk searching over k_i-tuples of the t_i vertices of degree near q_i. To optimize the overall walk, we take α_i steps for the i^{th} component, with carefully chosen values of the k_is and the α_is. Since we store the list of neighbors for each vertex, no queries are needed to check whether a set of 5 vertices forms a clique; queries are used only in the setup step (preparing the initial superposition) and the update step (taking α_i steps on the i^{th} Hamming graph for each i).

Storing the list of neighbors is costly for high-degree vertices, but cheap for low-degree vertices. However, since we know the graph is sparse, there cannot be too many high-degree vertices, so we are able to accept a high cost for such vertices. On the other hand, there may be many low-degree vertices, but since they are cheap to process, the total cost associated with all the vertices of a given approximate degree is about the same. This is why we break up the search space by grouping together vertices of similar degree. Note that the sparsity of the input graph is essential to upper bound the number of high-degree vertices in the graph.

Finally, note that it was unnecessary to search for all 5 vertices of the clique. Since we store the list of neighbors of each vertex, we can instead search for only 4 vertices, using the neighbor lists to determine whether they share a common neighbor. This idea naturally generalizes to storing a *vertex cover* of the graph H we are looking for, a subset C of the vertices of H such that each edge of H involves at least one vertex from C. Given neighbor lists for any particular subset of vertices, we can determine whether that subset includes a vertex cover of H with no further queries.

Our general strategy is to search over tuples of the vertices of G for one containing a vertex cover of H. We exploit sparsity by separately considering cases where the vertices of the vertex cover have given (approximate) degrees. Letting $\mathrm{vc}(H)$ denote the smallest number of vertices in any vertex cover of H, we have the following result.

▶ **Theorem 13.** *Let $\mathcal{P}$ be the property that a graph either has more than cn edges (for some constant c) or contains a given subgraph H. Then $Q(\mathcal{P}) = \tilde{O}\left(n^{\frac{3}{2}-\frac{1}{\mathrm{vc}(H)+1}}\right)$.*

The analysis proceeds roughly as follows. First we reject non-sparse graphs using Lemma 9. Then the quantum walk search algorithm uses $O(S + \frac{1}{\sqrt{\epsilon}}(\frac{1}{\sqrt{\delta}}U + C))$ queries, where S, U, C are the setup, update, and checking costs, respectively; ϵ is the fraction of marked states; and δ is the spectral gap of the walk [18]. We have $S = O(\sum_i k_i n/\sqrt{t_i})$, $U = O(\sum_i \alpha_i n/\sqrt{t_i})$, and $C = 0$. Furthermore, $\epsilon = \Omega(\prod_i k_i/t_i)$ and $\delta = \Omega(\min_i \alpha_i/k_i)$. By choosing $\alpha_i/\alpha_j = k_i/k_j = \sqrt{t_i/t_j}$, $\alpha_1 = 1$ (assuming without loss of generality that this is the smallest α_i), and $k_1 = \sqrt{t_1} n^{\frac{1}{2}-\frac{1}{\mathrm{vc}(H)+1}}$, we obtain the stated running time. Various technical issues that arise only increase the running time by a polylogarithmic factor.

We can apply this algorithm to decide sparse graph properties, and in particular minor-closed properties, that are also FSP: we simply search for each of the forbidden subgraphs, accepting if none of them are present. For minor-closed properties, the non-sparseness condition of Theorem 13 can be removed due to Theorem 3. Thus, since $\mathrm{vc}(H)$ is a constant for any fixed graph H, we have the following.

▶ **Corollary 14.** *If $\mathcal{P}$ is sparse and FSP, then $Q(\mathcal{P}) = o(n^{3/2})$.*

For many subgraphs, we can improve Theorem 13 further by storing more information about the vertices in the vertex cover: in addition to storing their neighborhoods, we can store basic information about their second neighbors. In particular, we have the following.

▶ **Theorem 15.** *Let $\mathcal{P}$ be the property that a graph either has more than cn edges (for some constant c) or contains a given subgraph H. Let H' be the graph obtained by deleting all degree-one vertices of H that are not part of an isolated edge. Then $Q(\mathcal{P}) = \tilde{O}\left(n^{\frac{3}{2}-\frac{1}{\mathrm{vc}(H')+1}}\right)$.*

Theorem 15 gives an improvement over Theorem 13 for properties that are characterized by a single forbidden minor (and equivalently, a single forbidden subgraph). For example, we have the following.

▶ **Theorem 16.** *A d-path with $d \geq 3$ can be detected using $\tilde{O}(n^{\frac{3}{2}-\frac{1}{\lceil d/2 \rceil}})$ quantum queries. In particular, the quantum query complexity of detecting a d-path for $d \in \{1,2,3,4\}$ is $\tilde{\Theta}(n)$.*

Note that even the improved result from Theorem 15 has zero checking cost. We can sometimes obtain a further improvement by performing nontrivial checking. For example, we can detect 7-paths in the same complexity that Theorem 16 gives for 5- and 6-paths and we can detect 9- and 10-paths in the same complexity that Theorem 16 gives for 7- and 8-paths:

▶ **Theorem 17.** *H-subgraph containment has query complexity $\tilde{O}(n^{7/6})$ if H is a 7-path and $\tilde{O}(n^{5/4})$ if H is a 9- or 10-path.*

Similar improvements are also possible for longer paths; we omit the details here.

4.4 Relaxing sparsity

So far we have focused on sparse graphs, since this is the relevant case for minor-closed properties. However, our algorithms easily generalize to the case where the number of edges is at most any prescribed upper bound, leading to further applications.

▶ **Theorem 18.** *Let $\mathcal{P}$ be the property that an n-vertex graph has at most $\bar{m}$ edges (where $\bar{m} = \Omega(n)$) and contains a given subgraph H. Let H' be the graph obtained by deleting all degree-one vertices of H that are not part of an isolated edge. Then $Q(\mathcal{P}) = \tilde{O}(\sqrt{\bar{m}}n^{1-\frac{1}{\mathrm{vc}(H')+1}})$.*

In conjunction with the Kővári-Sós-Turán theorem [16], this algorithm has applications to subgraph-finding problems that are not equivalent to minor-finding problems.

▶ **Theorem 19** (Kővári-Sós-Turán)**.** *If a graph G on n vertices does not contain $K_{s,t}$ as a subgraph, where $1 \le s \le t$, then $|E(G)| \le c_{s,t}\, n^{2-\frac{1}{s}}$, where $c_{s,t}$ is a constant depending only on s and t.*

Suppose H is a d-vertex bipartite graph. Theorem 19 shows that if $|E(G)| > c\,n^{2-\frac{2}{d}}$ (for some constant c), then G must contain $K_{s,d-s}$ for all $1 \le s \le d/2$, and in particular, must contain H. Combining this with the fact that $\mathrm{vc}(H') \le \mathrm{vc}(H) \le d/2$, we have the following.

▶ **Theorem 20.** *If H is a d-vertex bipartite graph, then H-subgraph containment has quantum query complexity $\tilde{O}(n^{2-\frac{1}{d}-\frac{2}{d+2}}) = \tilde{O}(n^{2-\frac{3d+2}{d(d+2)}})$.*

Recall that for $d > 3$, Theorem 4.6 of [19] gives an upper bound of $\tilde{O}(n^{2-\frac{2}{d}})$ for finding a d-vertex subgraph. For bipartite subgraphs, Theorem 20 is a strict improvement.

Note that a better bound may be possible by taking the structure of H into account. In general, if H is a bipartite graph with the i^{th} connected component having vertex bipartition $V_i \cup U_i$ with $1 \le |V_i| \le |U_i|$, then we can replace $d/2$ by $\sum_i |V_i|$, since a graph that contains $K_{\sum_i |V_i|, \sum_i |U_i|}$ must contain H, and $\mathrm{vc}(H') \le \mathrm{vc}(H) = \sum_i |V_i|$. As a simple example, if $H = K_{1,t}$ is a star on $t+1$ vertices, then H-subgraph containment can be solved with $\tilde{O}(n)$ quantum queries (which is essentially optimal due to Theorem 4).

Just as mentioned at the end of Section 4.3, we can sometimes improve over Theorem 18 by introducing a nontrivial checking cost. The following is a simple example of such an algorithm, using a result on the sparsity of graphs that exclude C_4 [8].

▶ **Theorem 21.** *C_4-subgraph containment can be solved in $\tilde{O}(n^{1.25})$ quantum queries.*

This may seem unexpected, since C_4 finding is a natural generalization of triangle finding to a larger subgraph. Indeed, the previous best known quantum algorithm for C_4 finding used $\tilde{O}(n^{1.5})$ queries [19], more than the $O(n^{1.3})$ queries for triangle finding. Our improvement shows that 4-cycles can be found in fewer quantum queries than in the best known quantum algorithm for finding 3-cycles.

5 Conclusions and open problems

In this paper, we have studied the quantum query complexity of minor-closed graph properties. The difficulty of such problems depends crucially on whether the property can also be characterized by a finite set of forbidden subgraphs. Minor-closed properties that are not characterized by forbidden subgraphs have matching upper and lower bounds of $\Theta(n^{3/2})$ (Corollary 12), whereas all minor-closed properties that can be expressed in terms of forbidden subgraphs can be solved strictly faster, in $o(n^{3/2})$ queries (Corollary 14).

Since the best known lower bound for the latter class of problems is the simple $\Omega(n)$ lower bound from Theorem 4, an obvious open question is to give improved upper or lower bounds for subgraph-finding problems. While the standard quantum adversary method cannot prove a better lower bound, it might be possible to apply the negative weights adversary method [14] or the polynomial method [6]. Note that sparsity makes forbidden subgraph properties potentially more difficult to lower bound; this is precisely the feature we took advantage of in the algorithms of Section 4. Proving a superlinear lower bound for any subgraph-finding problem—even one for which dense graphs might not contain the subgraph, such as in the case of triangles—remains a major challenge. On the algorithmic side, note that while our algorithms take advantage of sparsity, minor-closed families of graphs have other special properties, such as bounded degeneracy, that might also be exploited.

The algorithms described in Section 4 have several features not shared by previous quantum walk search algorithms for graph properties: queries are required even to identify which vertices of the input graph to search over (namely, to find vertices of a certain degree), and the performance of the walk is optimized by making different transitions at different rates. We hope these techniques might prove useful in other quantum algorithms.

Note that Theorem 13 can be applied to find induced subgraphs (just as with the algorithms of [19]). However, the improvements described in Theorem 15 and Theorem 18 do not apply to induced subgraphs, and in general it could be easier or more difficult to decide whether a given graph is present as an induced subgraph rather than a (not necessarily induced) subgraph. It might be fruitful to explore induced subgraph finding more generally.

It might also be interesting to focus on finding natural families of subgraphs such as paths. Recall that we showed the quantum complexity of this problem is $\tilde{\Theta}(n)$ for lengths up to 4 and $\tilde{O}(n^{7/6})$ for lengths of 5, 6, and 7, with nontrivial algorithms for longer paths as well (Theorem 16 and Theorem 17). The case of paths of length 5, the smallest case for which our algorithm is not known to be optimal, appears to be a natural target for future work.

Acknowledgements We thank Frédéric Magniez for suggesting that a strategy similar to the lower bound for connectivity in [12] could be applied to C_3-minor finding. R. K. thanks Jim Geelen for helpful discussions about graph minors, and Vinayak Pathak and Hrushikesh Tilak for interesting conversations about many aspects of this work.

References

1 S. Aaronson. Lower bounds for local search by quantum arguments. *SIAM Journal on Computing*, 35(4):804–824, 2006. Preliminary version in STOC 2004.

2 A. Ambainis. Quantum lower bounds by quantum arguments. *Journal of Computer and System Sciences*, 64(4):750–767, 2002. Preliminary version in STOC 2000.

3 A. Ambainis. Quantum walk algorithm for element distinctness. *SIAM Journal on Computing*, 37(1):210–239, 2007. Preliminary version in FOCS 2004.

4 A. Ambainis. A new quantum lower bound method, with an application to strong direct product theorem for quantum search. *Theory of Computing*, 6(1):1–25, 2010.

5 A. Ambainis, K. Iwama, M. Nakanishi, H. Nishimura, R. Raymond, S. Tani, and S. Yamashita. Quantum query complexity of boolean functions with small on-sets. In *Proceedings of the 19th International Symposium on Algorithms and Computation*, pages 907–918, 2008.

6 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. Preliminary version in FOCS 1998.

7 A. Berzina, A. Dubrovsky, R. Freivalds, L. Lace, and O. Scegulnaja. Quantum query complexity for some graph problems. In *SOFSEM 2004: Theory and Practice of Computer Science*, volume 2932, pages 1–11, 2004.

8 J. A. Bondy and M. Simonovits. Cycles of even length in graphs. *Journal of Combinatorial Theory, Series B*, 16(2):97–105, 1974.

9 G. Brassard, P. Høyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and estimation. In *Quantum computation and information*, volume 305 of *Contemporary Mathematics*, pages 53–74. AMS, 2002.

10 H. Buhrman, C. Dürr, M. Heiligman, P. Høyer, F. Magniez, M. Santha, and R. de Wolf. Quantum algorithms for element distinctness. *SIAM Journal on Computing*, 34(6):1324–1330, 2005. Preliminary version in CCC 2000.

11 A. Chakrabarti and S. Khot. Improved lower bounds on the randomized complexity of graph properties. *Random Structures and Algorithms*, 30(3):427–440, 2007. Preliminary version in ICALP 2001.

12 C. Dürr, M. Heiligman, P. Høyer, and M. Mhalla. Quantum query complexity of some graph problems. *SIAM Journal on Computing*, 35(6):1310–1328, 2006.

13 L. K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Physical Review Letters*, 79(2):325–328, 1997. Preliminary version in STOC 1996.

14 P. Høyer, T. Lee, and R. Spalek. Negative weights make adversaries stronger. In *STOC '07: Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, pages 526–535, 2007.

15 J. Kahn, M. Saks, and D. Sturtevant. A topological approach to evasiveness. In *SFCS '83: Proceedings of the 24th Annual Symposium on Foundations of Computer Science*, pages 31–33, 1983.

16 T. Kövari, V. T. Sós, and P. Turán. On a problem of K. Zarankiewicz. *Colloquium Mathematicum*, 3:50–57, 1954.

17 W. Mader. Homomorphieeigenschaften und mittlere Kantendichte von Graphen. *Mathematische Annalen*, 174:265–268, 1967.

18 F. Magniez, A. Nayak, J. Roland, and M. Santha. Search via quantum walk. In *STOC '07: Proceedings of the 39th ACM Symposium on Theory of Computing*, pages 575–584, 2007.

19 F. Magniez, M. Santha, and M. Szegedy. Quantum algorithms for the triangle problem. *SIAM Journal on Computing*, 37(2):413–424, 2007. Preliminary version in SODA 2005.

20 R. L. Rivest and J. Vuillemin. A generalization and proof of the Aanderaa-Rosenberg conjecture. In *STOC '75: Proceedings of the 7th Annual ACM Symposium on Theory of Computing*, pages 6–11, 1975.

21 N. Robertson and P. D. Seymour. Graph minors. VIII. A Kuratowski theorem for general surfaces. *Journal of Combinatorial Theory, Series B*, 48(2):255–288, 1990.

22 N. Robertson and P. D. Seymour. Graph minors. XX. Wagner's conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325–357, 2004.

23 A. L. Rosenberg. On the time required to recognize properties of graphs: a problem. *SIGACT News*, 5(4):15–16, 1973.

24 P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997. Preliminary version in FOCS 1994.

25 R. Špalek and M. Szegedy. All quantum adversary methods are equivalent. *Theory of Computing*, 2(1):1–18, 2006. Preliminary version in ICALP 2005.

26 X. Sun, A. C. Yao, and S. Zhang. Graph properties and circular functions: How low can quantum query complexity go? In *CCC '04: Proceedings of the 19th Annual IEEE Conference on Computational Complexity*, pages 286–293, 2004.

27 M. Szegedy. Quantum speed-up of Markov chain based algorithms. In *FOCS '04: Proceedings of the 45th IEEE Symposium on Foundations of Computer Science*, pages 32–41, 2004.

28 S. Zhang. On the power of Ambainis lower bounds. *Theoretical Computer Science*, 339(2):241–256, 2005. Preliminary version in ICALP 2004.