Parameterized Quantum Query Algorithms for Graph Problems

Tatsuya Terao
Research Institute for Mathematical Sciences, Kyoto University, Japan

Ryuhei Mori
Graduate School of Mathematics, Nagoya University, Japan

Abstract

In this paper, we consider the parameterized quantum query complexity for graph problems. We design parameterized quantum query algorithms for k-VERTEX COVER and k-MATCHING problems, and present lower bounds on the parameterized quantum query complexity. Then, we show that our quantum query algorithms are optimal up to a constant factor when the parameters are small. Our main results are as follows.

Parameterized quantum query complexity of vertex cover. In the k-VERTEX COVER problem, we are given an undirected graph G with n vertices and an integer k, and the objective is to determine whether G has a vertex cover of size at most k. We show that the quantum query complexity of the k-VERTEX COVER problem is $O(\sqrt{k}n + k^{3/2}\sqrt{n})$ in the adjacency matrix model. For the design of the quantum query algorithm, we use the method of kernelization, a well-known tool for the design of parameterized classical algorithms, combined with Grover's search.

Parameterized quantum query complexity of matching. In the k-MATCHING problem, we are given an undirected graph G with n vertices and an integer k, and the objective is to determine whether G has a matching of size at least k. We show that the quantum query complexity of the k-MATCHING problem is $O(\sqrt{k}n + k^2)$ in the adjacency matrix model. We obtain this upper bound by using Grover's search carefully and analyzing the number of Grover's searches by making use of potential functions. We also show that the quantum query complexity of the maximum matching problem is $O(\sqrt{p}n + p^2)$ where p is the size of the maximum matching. For small p, it improves known bounds $\tilde{O}(n^{3/2})$ for bipartite graphs [Blikstad–v.d.Brand–Efron–Mukhopadhyay–Nanongkai, FOCS 2022] and $O(n^{7/4})$ for general graphs [Kimmel–Witter, WADS 2021].

Lower bounds on parameterized quantum query complexity. We also present lower bounds on the quantum query complexities of the k-VERTEX COVER and k-MATCHING problems. The lower bounds prove the optimality of the above parameterized quantum query algorithms up to a constant factor when k is small. Indeed, the quantum query complexities of the k-VERTEX COVER and k-MATCHING problems are both $\Theta(\sqrt{k}n)$ when $k = O(\sqrt{n})$ and $k = O(n^{2/3})$, respectively.

2012 ACM Subject Classification Theory of computation → Quantum query complexity

Keywords and phrases Quantum query complexity, parameterized algorithms, vertex cover, matching, kernelization

Digital Object Identifier 10.4230/LIPIcs.ESA.2024.99

Related Version *Full Version*: https://arxiv.org/abs/2408.03864

Funding *Tatsuya Terao*: KAKENHI Grant Number JP24KJ1494.
Ryuhei Mori: JST FOREST Program Grant Number JPMJFR216V and KAKENHI Grant Numbers JP20H04138, JP20H05966, and JP22H00522.

Acknowledgements The authors thank Yusuke Kobayashi and Kazuhisa Makino for their insightful comments.

1 Introduction

The query complexity for Boolean functions has been studied much both on the classical and quantum settings. In contrast to the circuit complexity, the quantum advantages on the query complexity have been rigorously proved for several problems [11, 41]. While the classical query complexities are trivial for many graph problems, the quantum query complexities are non-trivially small for many graph problems [4–9, 12, 14, 15, 17, 18, 22, 23, 27, 28, 35, 36, 38, 45–47, 49, 51, 52, 54, 55, 57, 62–64].

In this paper, we consider the *parameterized* quantum query complexity for graph problems. In terms of the classical time complexity, the theory of parameterized time complexity classifies NP-hard problems on a finer scale. Some graph problems can be solved in polynomial time, e.g., the maximum matching problem. On the other hand, several graph problems are NP-hard, e.g., the minimum vertex cover problem, which can not be solved in polynomial time unless P = NP. For the detailed analysis of the NP-hard problems, we have studied the parameterized complexity, where we design and analyze algorithms for problems with fixed parameters. Then, our goal is to design algorithms that perform efficiently when the parameters are small. Formally, the definition of a *parameterized problem* is a language $L \subseteq \Sigma^* \times \mathbb{N}$, where Σ is a fixed finite alphabet. For an instance $(x, k) \in \Sigma^* \times \mathbb{N}$, we call k the *parameter*. The main focus in the field of the parameterized complexity is whether there exists an algorithm to solve the parameterized problem in $f(k) \cdot |x|^{O(1)}$ time, where f is an arbitrary computable function and $|x|$ is the size of x. A parameterized problem is called *Fixed Parameter Tractable (FPT)* if such an algorithm exists. The running time of the FPT algorithms is polynomial in the input size $|x|$ with a constant exponent independent of the parameter k if the parameter k is constant. This parameter is typically taken to be the size of the solution. Then, we consider the problem of deciding whether there is a solution of size bounded by k. For example, the k-vertex cover problem, which asks whether a given graph has a vertex cover of size at most k, can be solved in $O(2^k \cdot n)$ time. This means that the k-vertex cover problem is FPT.

For the adjacency matrix model, the quantum query complexity for graph problems is trivially $O(n^2)$, where n is the number of vertices. For the parameterized graph problems, our objective is to design quantum query algorithms with the query complexity $f(k) \cdot n^c$, where c is some constant smaller than 2. We will introduce a class of quantum query algorithms called *Fixed Parameter Improved (FPI)* algorithms. For the definition of FPI, we consider an unparameterized problem of the parameterized problem. For example, the unparameterized problems of the k-vertex cover and k-matching problems are the minimum vertex cover and the maximum matching problems, respectively. If the query complexity of the unparameterized problem is $\Omega(n^e)$, the parameterized problem is FPI if its query complexity is at most $f(k) \cdot n^c$ for some constant $c < e$ independent of k. When we can take $f(k)$ as a polynomial, we say that the parameterized problem is polynomial FPI.

For the minimum vertex cover problem and the maximum matching problem, the quantum query complexity is $\Omega(n^{3/2})$ [63]. In this work, we prove that the quantum query complexities of the k-vertex cover problem and k-matching problem are $O(\sqrt{k}n+k^{3/2}\sqrt{n})$ and $O(\sqrt{k}n+k^2)$, respectively. Hence, the k-vertex cover and k-matching problems are polynomial FPI.

We also prove lower bounds $\Omega(\sqrt{k}n)$ of the quantum query complexities of the k-vertex cover and k-matching problems using the adversary method. Hence, the quantum query algorithms in this paper are optimal up to a constant factor for small k.

Note that no non-trivial quantum query complexity $O(n^{2-\epsilon})$ has been obtained for the minimum vertex cover problem. The best known upper bound on the quantum query complexity of the maximum matching problem is $O(n^{7/4})$ [49].

In summary, this work considers the parameterized quantum query complexity for graph problems, introduces FPI algorithms, and presents quantum query algorithms for the k-vertex cover and k-matching problems that are polynomial FPI and optimal for small k.

1.1 Our Contribution

1.1.1 Parameterized Quantum Query Complexity of Vertex Cover

In this paper, we consider the quantum query complexity of the k-VERTEX COVER problem, which is one of the central problems in the study of parameterized computation. This problem is also one of Karp's 21 NP-complete problems [48]. A vertex set $S \subseteq V(G)$ is a *vertex cover* if every edge of the graph has at least one endpoint in S. In the k-VERTEX COVER problem, we are given an undirected graph $G = (V, E)$ and an integer k, and the objective is to determine whether G has a vertex cover of size at most k.

Here, we mention some known results related to our work. There have been many studies to improve the running time of an FPT algorithm for the k-VERTEX COVER in the literature [10, 21, 24–26, 37, 43, 58, 59, 61]. The state-of-the-art algorithm, given by Harris–Narayanaswamy [43], has the time complexity $O(1.2575^k \cdot \mathrm{poly}(n))$.

The minimum vertex cover problem is the optimization version of the k-VERTEX COVER problem. In this problem, the objective is to find the smallest vertex cover of G. The trivial upper bound $O(n^2)$ on the quantum query complexity of the minimum vertex cover problem in the adjacency matrix model has not been improved. The largest known lower bound on the quantum query complexity of the minimum vertex cover problem is $\Omega(n^{3/2})$ in the adjacency matrix model [63].[1]

The first main contributions of this paper is to obtain an upper bound on the quantum query complexity of the k-VERTEX COVER problem.

▶ **Theorem 1.** *The quantum query complexity for finding a vertex cover of size at most k or determining that there does not exist a vertex cover of size at most k with bounded error is $O(\sqrt{kn} + k^{3/2}\sqrt{n})$ in the adjacency matrix model.*

We also provide a lower bound on the quantum query complexity of the k-VERTEX COVER problem.

▶ **Theorem 2.** *For any constant $\epsilon > 0$, given a graph G with n vertices in the adjacency matrix model and an integer $k \leq (1-\epsilon)n$, the quantum query complexity of deciding whether G has a vertex cover of size at most k with bounded error is $\Omega(\sqrt{kn})$.*

The upper and lower bounds on the quantum query complexity of the k-VERTEX COVER problem are summarized in Figure 1. From Theorems 1 and 2, we obtain the following corollary.

▶ **Corollary 3.** *The quantum query complexity of deciding whether G has a vertex cover of size at most k with bounded error is $\Theta(\sqrt{kn})$ when $k = O(\sqrt{n})$ in the adjacency matrix model.*

[1] Zhang [63] showed that the quantum query complexity of the maximum matching problem is $\Omega(n^{3/2})$. For bipartite graphs, the minimum vertex cover has the same cardinality as the maximum matching by well-known König's theorem (see e.g., [60, Theorem 16.2]). Thus, we can easily obtain an $\Omega(n^{3/2})$ lower bound for the minimum vertex cover problem.

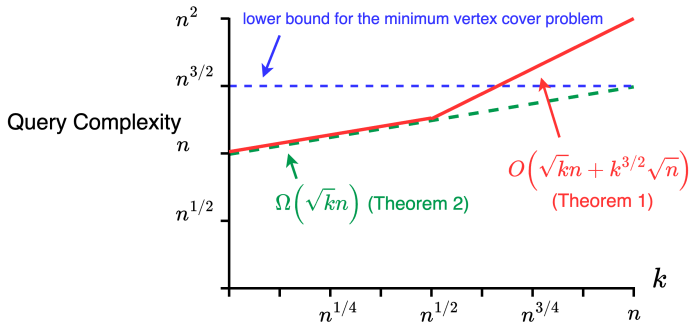

Figure 1 The quantum query complexity of the k-VERTEX COVER problem.

A graph G has a vertex cover of size at most k if and only if the complement graph $\overline{G}$ has a clique of size at least $n-k$. Hence, our result given in Corollary 3 implies new bounds on the quantum query complexity of the k-clique problem, which is a well-studied problem in the quantum query model [14, 23, 27, 47, 51, 52, 54, 55, 57, 64]. In the k-clique problem, the objective is to find a clique of size at least k. The best known upper bound on the quantum query complexity of the k-clique problem in the adjacency matrix model is $O(n^{2-2/k-g(k)})$ where $g(k) = O(1/k^3)$ is a strictly positive function [54, 64]. Note that only the trivial $\Omega(n)$ lower bound is known. While non-trivial optimal query complexity of the k-clique problem has not been known for any k, Corollary 3 establishes non-trivial optimal query complexity $\Theta(\sqrt{kn})$ of detecting cliques of size at least $n-k$ for any $k = O(\sqrt{n})$.

We also provide a lower bound on the randomized query complexity of the k-VERTEX COVER problem.

▶ **Proposition 4.** *Given a graph G with n vertices in the adjacency matrix model and an integer $k < n-1$,[2] the randomized query complexity of deciding whether G has a vertex cover of size at most k with bounded error is $\Omega(n^2)$.*

1.1.2 Parameterized Quantum Query Complexity of Matching

We also consider the quantum query complexity of the k-MATCHING problem, which is the parameterized version of the maximum matching problem. An edge set $M \subseteq E(G)$ is a *matching* if each vertex in $V(G)$ appears in at most one edge in M. In the k-MATCHING problem, we are given an undirected graph $G = (V, E)$ and an integer k, and the objective is to determine whether G has a matching of size at least k.

We obtain an upper bound on the quantum query complexity of the k-MATCHING problem.

▶ **Theorem 5.** *The quantum query complexity of finding a matching of size at least k or determining that there does not exist a matching of size at least k with bounded error is $O(\sqrt{kn}+k^2)$ in the adjacency matrix model.*

We also provide a lower bound on the quantum query complexity of the k-MATCHING problem.

[2] We consider only the setting where $k < n-1$, since an input instance is always a yes-instance in the setting where $k \geq n-1$.

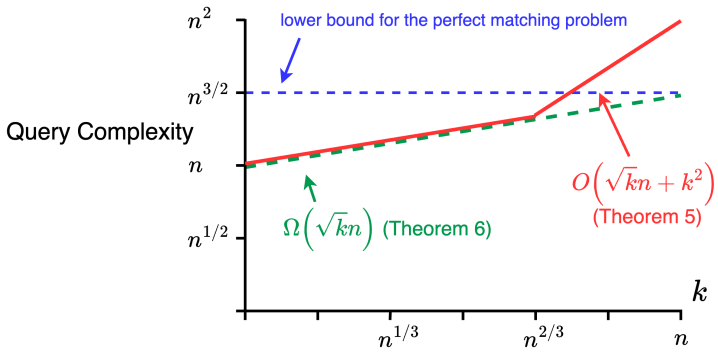

Figure 2 The quantum query complexity of the k-MATCHING problem.

▶ **Theorem 6.** *Given a graph G with n vertices in the adjacency matrix model and an integer $k \leq n/2$, the quantum query complexity of deciding whether G has a matching of size at least k with bounded error is $\Omega(\sqrt{kn})$.*

Zhang [63] showed that the quantum query complexity of deciding whether G has a perfect matching is $\Omega(n^{3/2})$. Then, Theorem 6 is a generalization of the result of Zhang.

The upper and lower bounds on the quantum query complexity of the k-MATCHING problem are summarized in Figure 2. From Theorems 5 and 6, we obtain the following corollary.

▶ **Corollary 7.** *The quantum query complexity for deciding whether G has a matching of size at least k with bounded error is $\Theta(\sqrt{kn})$ when $k = O(n^{2/3})$ in the adjacency matrix model.*

We also provide a lower bound on the randomized query complexity of the k-MATCHING problem.

▶ **Proposition 8.** *Given a graph G with n vertices in the adjacency matrix model and an integer $k \leq n/2$, the randomized query complexity of deciding whether G has a matching of size at least k with bounded error is $\Omega(n^2)$.*

By using Theorem 5, we also obtain the quantum query complexity of the maximum matching problem. Here, let p denote the size of the maximum matching.

▶ **Theorem 9.** *The quantum query complexity for finding a maximum matching with bounded error is $O(\sqrt{pn} + p^2)$ in the adjacency matrix model.*

This query complexity improves upon known results when p is small. Lin–Lin [56] obtained an $O(p^{1/4}n^{3/2})$ upper bound on the quantum query complexity of the maximum matching problem for bipartite graphs in the adjacency matrix model[3], which is the first nontrivial upper bound. By using the method developed by Lin–Lin, Kimmel–Witter [49]

[3] Their paper only states that the quantum query complexity of the bipartite maximum matching problem is $O(n^{7/4})$ in the adjacency matrix model. We see that this upper bound can be easily improved to $O(p^{1/4}n^{3/2})$. Their quantum query algorithm relies on the classical algorithm by Hopcroft–Karp [44]. In the algorithm of Hopcroft–Karp, they repeatedly find a maximal set of vertex disjoint shortest augmenting paths. To obtain an $O(n^{7/4})$ upper bound, Lin–Lin use the fact that $O(\sqrt{n})$ iterations of these processes suffice. It is known that only $O(\sqrt{p})$ iterations suffice; see [44, Theorem 3]. Then, we can obtain an $O(p^{1/4}n^{3/2})$ upper bound for the bipartite maximum matching problem.

obtained an $O(p^{1/4}n^{3/2})$ upper bound for general graphs. Moreover, Blikstad–v.d.Brand–Efron–Mukhopadhyay–Nanongkai [18] showed that the quantum query complexity of the bipartite maximum matching problem is $\tilde{O}(n^{3/2})$ in the adjacency matrix model[4].

1.1.3 Fixed Parameter Improved Algorithms

For a better understanding of the main results of this work, we introduce a new class of query algorithms for graph problems, which we call *fixed parameter improved (FPI)* algorithms. First, we define parameterized graph problems.

▶ **Definition 10** (Parameterized graph problem)**.** *In a parameterized graph problem, inputs are the number of vertices n, an oracle for a n-vertex graph G, typically in the adjacency matrix model, and a nonnegative integer k as a parameter. Here, the parameter k is polynomially bounded in n. The output of the problem is some property of the graph G parameterized by k.*

Then, we introduce unparameterized graph problems of parameterized graph problems.

▶ **Definition 11** (Unparameterized graph problem)**.** *In an unparameterized graph problem of a parameterized graph problem $\mathcal{P}$, inputs are the number of vertices n, and an oracle for a n-vertex graph G. The output of the unparameterized problem is a list of outputs of the parameterized problem $\mathcal{P}$ for all possible parameters k.*

▶ Remark 12. In many cases, a parameterized graph problem $\mathcal{P}$ is binary, i.e., the output is YES or NO, and monotone, i.e., if the output for (G, k) is YES or NO, then the output for $(G, k+1)$ is also YES or NO, respectively. In this case, the unparameterized graph problem $\mathcal{Q}$ for $\mathcal{P}$ is essentially an optimization problem that asks the minimum k for which (G, k) is a YES or NO instance of $\mathcal{P}$.

Now, we define the fixed parameter improved algorithms.

▶ **Definition 13** (Fixed parameter improved algorithms)**.** *Let $\mathcal{P}$ and $\mathcal{Q}$ be a parameterized graph problem and its unparameterized graph problem, respectively. Let*

$$C(\mathcal{Q}) := \inf\{c \mid \textit{The query complexity of } \mathcal{Q} \textit{ is } O(n^c)\}.$$

A query algorithm $\mathcal{A}$ for $\mathcal{P}$ is said to be fixed parameter improved if there exists a constant $c < C(\mathcal{Q})$ and a function $f(k)$ such that the query complexity of $\mathcal{A}$ is at most $f(k)n^c$. When $f(k)$ is a polynomial, $\mathcal{A}$ is said to be polynomial FPI.

The concept of FPI can be regarded as a natural generalization of FPT. The correspondence between FPI and FPT is summarized in Table 1. The quantum query algorithms in Theorems 1 and 5 are polynomial FPI since $C(\mathcal{Q}) \geq 3/2$ for both the problems [63]. On the other hand, the quantum query algorithms for the k-clique problem in [54, 57, 64] with query complexity $n^{2-O(1/k)}$ are not FPI.

Note that the quantum algorithms in this paper are *fixed parameter linear (FPL)*, which means that the query complexity is at most $f(k)n$ for some function f. Since we can take $f(k)$ as a polynomial, the quantum algorithms in this paper are polynomial FPL.

[4] The $\tilde{O}$ notation hides factors polynomial in $\log n$.

Table 1 The correspondence between FPI and FPT.

	Unparameterized complexity	Not FPT/FPI	FPT/FPI
Time complexity	$n^{\omega(1)}$	$n^{\omega_k(1)}$	$\exists c < \infty \quad f(k)n^c$
Query complexity	$O\left(n^{C(\mathcal{Q})+\epsilon}\right) \forall \epsilon > 0$	$n^{C(\mathcal{Q})-o_k(1)}$	$\exists c < C(\mathcal{Q}) \quad f(k)n^c$

1.2 Overview of Our Upper Bounds

1.2.1 Grover's Search Algorithm

To obtain our quantum query algorithms for the k-VERTEX COVER and k-MATCHING problems, we only use some variants of Grover's search algorithm [41], which is a fundamental tool for quantum query algorithms.

In Grover's search algorithm, we are given oracle access to a function $f\colon [N] \to \{0,1\}$ where $K := |f^{-1}(1)|$ is positive. Grover's search algorithm finds some $x \in f^{-1}(1)$ with $\Theta(\sqrt{N/K})$ expected number of queries [19]. This algorithm does not require prior knowledge of K.

As a simple application of Grover's search algorithm, for given $1 \leq R \leq N$, we can compute some subset $S \subseteq f^{-1}(1)$ of size $\min\{R, K\}$ with probability at least $2/3$ with query complexity $O(\sqrt{RN})$ queries (see e.g., [20], [38, Fact 2.1], and [8, Theorem 13]).

In our quantum query algorithms, we use Grover's search algorithm to find some edges in G. Note that, in the adjacency matrix model, we are given query access to a function $E_{\mathrm{G}}\colon \binom{V(G)}{2} \to \{0,1\}$ where $E_{\mathrm{G}}(v,u) = 1$ if and only if $(v,u) \in E(G)$.

1.2.2 Overview of Our Parameterized Quantum Query Algorithm for Vertex Cover

The parameterized quantum query algorithm for the k-vertex cover. It is easy to obtain a bounded error quantum algorithm for the k-VERTEX COVER problem with $O(\sqrt{k}n^{3/2})$ queries. To obtain this algorithm, we use the following easy observation: if there are more than $k(n-1)$ edges in the input graph G, G does not have a vertex cover of size at most k. In the algorithm, we check whether there are at most $k(n-1)$ edges in G, and if so, we find all edges in G. By Grover's search algorithm, this requires $O(\sqrt{k(n-1)\cdot n^2}) = O(\sqrt{k}n^{3/2})$ queries. After finding all edges in G, we simply apply a known classical algorithm for the k-VERTEX COVER problem. Then, the quantum query complexity of the k-VERTEX COVER problem is $O(\sqrt{k}n^{3/2})$. However, this upper bound is far from the optimal.

Kernelization. For the improvement of the query complexity, we use the technique of *kernelization*, which is a important tool for designing parameterized algorithms [40, 42]. A kernelization is an efficient preprocessing algorithm that transforms a given instance to a smaller instance called a *kernel*. Here, we formally define a kernelization for a parameterized problem $L \subseteq \Sigma^* \times \mathbb{N}$, where Σ is a fixed finite alphabet. A kernelization is an algorithm that maps an instance (x,k) of L to another instance (x',k') of L with running time polynomial in $|x|$ and k, requiring that $|x'| + k'$ is bounded by a function of k. Here, we also require that $(x,k) \in L$ if and only if $(x',k') \in L$. Then, it is sufficient to check whether $(x',k') \in L$. The output (x',k') is called a *kernel* of (x,k).

Typically, for graph problems, a kernel (G',k') of an instance (G,k) satisfies $V(G') \subseteq V(G)$ and $E(G') \subseteq E(G)$, i.e., G' is a subgraph of G. In this work, we consider a quantum query algorithm that outputs a kernel (G',k') directly from k and the given oracle for G. Since

the size of kernels is bounded by a function of k, the number of edges in G' is bounded by a function of k. This observation suggests that the kernelization technique may yield low-query quantum algorithms for parameterized problems. In this paper, we demonstrate, for the first time, the utility of the kernelization technique in the context of parameterized quantum query complexity.

Quantum query algorithm for kernelization. We introduce a new framework, called *quantum query kernelization*, to obtain better quantum query complexity. A quantum query kernelization algorithm outputs a kernel (G', k') as a bit string. In the quantum query setting, the input graph G is provided as a quantum oracle, through which quantum algorithms access G. Once we obtain a kernel (G', k'), we do not require more queries since we can decide whether (G', k') is a yes-instance by an arbitrary classical algorithm for (G', k').

Our quantum query kernelization algorithm is based on a known classical kernelization algorithm. However, not all classical efficient kernelization algorithms give efficient quantum query kernelization algorithms since they are not necessarily suitable for Grover's search algorithm. Hence, we have to choose appropriate kernels to derive better upper bounds.

Classical kernelization algorithm for the k-vertex cover problem. In this section, we introduce the classical kernelization algorithm for the k-vertex cover problem which will be used for designing the quantum query algorithm in Theorem 1. We first find a maximal matching M of G. Note that a maximal matching is a matching that is not a subset of any other matching. If the size of M is at least $k+1$, then we conclude that the input instance (G, k) is a no-instance. Let $V(M) \subseteq V$ denote the set of all endpoints of edges in M. We initially set $(G', k') = (G, k)$. For each vertex $v \in V(M)$, if v has degree at least $k+1$, then we update (G', k') to $(G' - v, k' - 1)$.[5] Then, we return the instance (G', k').

Now, the instance (G', k') obtained by the above procedure is a yes-instance if and only if (G, k) is a yes-instance. This follows from the following simple observation.

▶ **Observation 14** ([21]; see also [32, Section 2.2.1]). *If G has a vertex v of degree at least $k+1$, then v must be in every vertex cover of size at most k.*

This is because if a vertex v is not in a vertex cover, then the vertex cover must contain all neighbors of v to cover all edges incident to v.

Furthermore, (G', k') obtained by the above procedure has size depending only on k and not on n. Let $S \subseteq V(M)$ be a subset of $V(M)$ that are contained in G'. Then, G' is a subgraph of G induced by S and it's neighborhoods in G since M is a maximal matching of G. Note that any $v \in S$ has degree at most k in G'. This fact implies that G' has at most $k \cdot |S| \leq k \cdot |V(M)| \leq 2k^2$ edges. Note that G' consists of the maximal matching M (with some missing vertices due to the deletion procedure) and a independent set $V \setminus V(M)$ as in Figure 3.

A similar idea was used in a parameterized streaming algorithm for the vertex cover problem [31].

The optimal quantum query complexity of the k-vertex cover for small k. Now, we present a quantum query kernelization algorithm based on the above classical kernelization algorithm. In the above classical kernelization algorithm, there are multiple possible choices

[5] For a vertex $v \in V(G)$, let $G - v$ denote the subgraph of G induced by $V(G) \setminus \{v\}$.

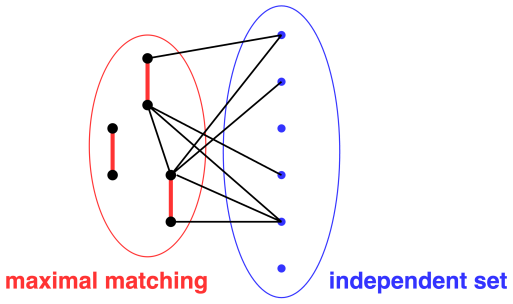

Figure 3 The kernel for the k-vertex cover problem consists of maximal matching (with some missing vertices) and independent set. The size of the maximal matching is at most k. All non-missing vertices in the maximal matching have degree at most k.

of the maximal matching and the order of the vertex eliminations of degrees greater than k. Let $\mathcal{I}^{\mathrm{vc}}_{(G,k)}$ denote the set of all possible instances that can be returned by the above classical kernelization algorithm for an input instance (G,k). In our quantum query kernelization algorithm, we obtain a kernel $(G',k') \in \mathcal{I}^{\mathrm{vc}}_{(G,k)}$ as a bit string with a small number of queries.

▶ **Theorem 15.** *Given a graph G with n vertices in the adjacency matrix model and an integer k, there is a bounded error quantum algorithm with $O(\sqrt{kn}+k^{3/2}\sqrt{n})$ queries that outputs an instance $(G',k') \in \mathcal{I}^{\mathrm{vc}}_{(G,k)}$, or otherwise determines that the instance (G,k) is a no-instance of the k-vertex cover problem.*

Obviously, Theorem 15 implies Theorem 1. In our quantum query kernelization algorithm, we first find a matching of size at least $k+1$ or a maximal matching. Grover's search algorithm yields a quantum algorithm with $O(\sqrt{kn})$ queries. If we find a matching of size at least $k+1$, we conclude that the input instance (G,k) is a no-instance. Assume that we obtain a maximal matching M of size at most k. We initialize sets $V'=V(G)$, $E'=\emptyset$ and an integer $k'=k$. For each vertex $v \in V(M)$, we find all edges incident to v if v has degree at most k or, otherwise determine that v has degree at least $k+1$. If v has degree at most k, we add all edges incident to v to E'. If v has degree at least $k+1$, then we remove v from V' and update k' to $k'-1$. For each $v \in V(M)$, Grover's search algorithm finds at most $k+1$ edges incident to v from $n-1$ candidates with $O(\sqrt{kn})$ queries. Hence, in total, we need $O(k^{3/2}\sqrt{n})$ queries. Then, we obtain a kernel $(G'=(V',E'),k') \in \mathcal{I}^{\mathrm{vc}}_{(G,k)}$. The total number of queries in the above quantum query kernelization algorithm is $O(\sqrt{kn}+k^{3/2}\sqrt{n})$.

1.2.3 Overview of Our Parameterized Quantum Query Algorithm for Matching

In our algorithm for the k-matching problem, we first find a matching of size at least k or a maximal matching by Grover's search with $O(\sqrt{kn})$ queries. If we find a matching of size at least k, we conclude that the input instance (G,k) is a yes-instance. Now, we assume that we obtain a maximal matching M of size at most $k-1$. Then, our algorithm relies on the theory of the augmenting path, a widely used approach for computing the maximum matching problem. We repeatedly find an M-augmenting path and update the current maximal matching M until the size of M is equal to k or there is no M-augmenting path, which means that M is a largest matching.

For a matching M, a path P on G is an M-augmenting path if the both endpoints of P does not appear in M, and P contains edges from M and $E \setminus M$, alternatively. It is easy to see that the symmetric difference $M \triangle E(P) = (M \setminus E(P)) \cup (E(P) \setminus M)$ is a matching of size $|M| + 1$. It is well known that the size of a matching M is largest if and only if there is no M-augmenting path (see e.g., [50, Theorem 10.7]).

In the each step of the algorithm, we have a maximal matching M and all edges between vertices in M. Then, we can classically find a candidate path that may be extended to an M-augmenting path by adding both endpoints. Then, we check whether or not the candidate path can be extended to an M-augmenting path by applying Grover's search algorithm for finding neighborhoods of both the endpoints of the candidate path. Hence, each step requires $O(\sqrt{n})$ queries. For estimating the number of steps, we introduce the types of vertices and a potential function.

For each $v \in V(M)$, let $N_M(v)$ denote $\{u \in V(G) \setminus V(M) \mid \{v, u\} \in E(G)\}$. In the algorithm, we classify each vertex v in $V(M)$ into the following three types:

- `type0` We have determined that $N_M(v) = \emptyset$.
- `type1` We have determined that $|N_M(v)| = 1$.
- `type2` No information about $N_M(v)$.

At the beginning of the algorithm with some maximal matching M, all vertices in $V(M)$ are `type2`. In the algorithms, the types of vertices in $V(M)$ are stored and updated. For each vertex v of `type1`, the unique neighborhood in $N_M(v)$ is stored as well. Let $G[S]$ denote the subgraph of G induced by $S \subseteq V(G)$. For a maximal matching M of graph G, a path Q on $G[V(M)]$ is a candidate path if Q satisfies the following conditions:

1. The both ends of Q are `type1` or `type2`. If the both are `type1`, their neighbors in $V(G) \setminus V(M)$ are different.
2. Q contains edges from M and $E(G[V(M)]) \setminus M$, alternatively.
3. Q contains edges from M at the both ends.

It is obvious that for any M-augmenting path P, the path Q obtained by deleting the vertices at the both ends of P is a candidate path. In each step of the algorithm, we have a maximal matching M, the induced graph $G[V(M)]$ and the types of the vertices in $V(M)$, and search for a candidate path. If there is no candidate path, we can conclude that the maximal matching M is a maximum matching since there is no M-augmenting path. In this case, the algorithm outputs NO and terminates. Assume that there exists a candidate path Q. If the both ends of Q are `type1`, then Q can be extended to an M-augmenting path. If at least one of the ends of Q is `type2`, we apply Grover's search algorithm for finding the neighbors in $V(G) \setminus V(M)$ of the vertices of `type2`. If we find the different neighbors of the end vertices of Q, the candidate path Q can be extended to a M-augmenting path. If not, we can update the type of one of the `type2` vertex to `type0` or `type1`. If an M-augmenting path P is found, the maximal matching M is updated to $M \triangle E(P)$ of size $|M| + 1$. In this case, the types of the two new vertices in the maximal matching are set to `type0` since $V(G) \setminus V(M)$ is an independent set. We update the type of `type1` vertices that connect to either of the two new vertices to `type0`. Finally, we queries classically for all the new pairs in $V(M)$ for obtaining the graph $G[V(M)]$. This is the single step of the algorithm. If the size of maximal matching M achieves k, then the algorithm outputs YES, and terminates.

For each step, the size of maximal matching increases by one, or the number of vertices of `type2` decreases by one. From this observation, we define the potential function $\Phi := k - |M| + N_2$ where N_2 is the number of vertices of `type2`. Then, it is easy to see that $0 \leq \Phi \leq k - |M| + 2|M| \leq 2k$. The above observation implies that the value of the potential function decreases by one at each step. Hence, the number of iterations is at most $2k$.

Finally, we estimate the query complexity of the quantum algorithm. The query complexity for finding a maximal matching is $O(\sqrt{kn})$. The query complexity for finding neighbors of the both ends of a candidate path is $\tilde{O}(\sqrt{n})$ at each step. The number of classical queries for obtaining all edges in $G[V(M)]$ is $O(k^2)$ in total. Hence, the query complexity of the algorithm is $O(\sqrt{kn}) + \tilde{O}(k\sqrt{n}) + O(k^2) = O(\sqrt{kn} + k^2)$.

1.3 Overview of Our Lower Bounds

Our lower bound is based on the adversary method introduced by Ambainis [1]. Although the technique is standard for proving lower bounds of quantum query complexity, we adapt this technique for the parameterized setting, and prove the lower bounds which matches to the upper bounds obtained in this paper for small k.

To obtain the lower bound for the k-VERTEX COVER problem, we consider the problem of distinguishing graphs that consist of k disjoint edges and graphs that consist of $k+1$ disjoint edges. Any quantum algorithm for the k-VERTEX COVER problem must distinguish the above two classes of graphs. By applying the adversary method for this problem, we obtain a lower bound $\Omega(\sqrt{kn})$ when $k \leq n/3$.

To prove a lower bound $\Omega(\sqrt{kn})$ for any $k \leq (1-\epsilon)n$, we consider generalized version of the problem. In the problem, we distinguish graphs that consist of $\lfloor \frac{k}{t-1} \rfloor$ cliques of size t and graphs that consist of $\lceil \frac{k+1}{t-1} \rceil$ cliques of size t. The proof of our lower bound for the k-MATCHING problem is similar to that for the k-vertex cover problem.

1.4 Additional Related Work

Ambainis-Balodis-Iraids-Kokainis-Prūsis-Vihrovs [2] and Le Gall-Seddighin [53] combine known standard quantum algorithms, such as Grover's algorithm, with classical techniques to design quantum algorithms with low time-complexities.

There are few previous works on the parameterized quantum query complexity. For the graph collision problem, we are given an undirected graph G as a bit string and oracle access to Boolean variables $\{x_v \in \{0,1\} \mid v \in V(G)\}$ and, the objective is to decide whether there are two vertices v and u connected by an edge in G such that $x_v = x_u = 1$. Belovs [13] showed that the quantum query complexity of this problem is $O(\sqrt{n}\alpha^{1/6})$, where α is the size of the largest independent set of G. Ambainis–Balodis–Iraids–Ozols–Smotrovs–Juris [3] also showed that the quantum query complexity is $O(\sqrt{n}t^{1/6})$, where t is the treewidth of G. The only trivial $\Omega(\sqrt{n})$ lower bound is known for this problem. We note that these studies did not use kernelization.

The usefulness of the concept of the parameterized complexity has been successfully demonstrated in other computational models, such as streaming model [29–31, 39] and distributed model [16], as well. The study of parameterized streaming algorithms initiated by Fafianie–Kratsch [39] and Chitnis–Cormode–Hajiaghayi–Monemizadeh [31] has been developed further by several authors. In parameterized streaming algorithms, by analogy of FPT, Chitnis–Cormode [29] introduced a hierarchy of space complexity classes and tight classification for several graph problems.

There have also been many studies of the quantum query complexity of graph problems in the adjacency matrix model. Dürr–Heiligman–Høyer–Mhalla [38] showed the quantum query complexity of testing connectivity is $\Theta(n^{3/2})$. Ambainis–Iwama–Nakanishi–Nishimura–Raymond–Tani–Yamashita [5] showed that the quantum query complexity of testing the planarity is also $\Theta(n^{3/2})$. Belovs–Reichardt [15] presented a quantum algorithm that uses $O(\sqrt{dn})$ queries to decide whether vertices s and t are connected, under the promise that they

are connected by a path of length at most d, or are disconnected. Apers–Lee [8] presented a quantum algorithm to solve the weighted minimum cut problem using $O(n^{3/2}\sqrt{\tau})$ queries and time in the adjacency matrix model, where each edge weight is at least 1 and at most τ.

Now, we will mention some known results closely related to our work. For bipartite graphs, the minimum vertex cover has the same cardinality as the maximum matching by König's theorem (see e.g., [60, Theorem 16.2]). Moreover, Blikstad–v.d.Brand–Efron–Mukhopadhyay–Nanongkai [18] showed that the quantum query complexity of the bipartite maximum matching problem is $\tilde{O}(n^{3/2})$. Then, for bipartite graphs, the quantum query complexity of the minimum vertex cover problem is $\tilde{O}(n^{3/2})$, which is almost optimal.

Childs–Kothari [28] showed that any non-trivial minor-closed property that can not be described by a finite set of forbidden subgraphs has quantum query complexity $\Theta(n^{3/2})$. On the other hand, they also show that any minor-closed properties that can be characterized by finitely many forbidden subgraphs can be solved in $O(n^{\alpha})$ queries for some $\alpha < 3/2$. To show this, they proved that, for any constant $c > 0$, a graph property that a graph either has more than cn edges or contains a given subgraph H can be decided in $\tilde{O}(n^{3/2-1/(\mathrm{vc}(H)+1)})$ queries, where $\mathrm{vc}(H)$ denote the size of minimum vertex cover of H.

Here, it is easy to verify that the property of having a vertex cover of size at most k is minor-closed. Dinneen–Lai [34, Corollary 2.5] showed that the property of having a vertex cover of size at most k can be characterized by finitely many forbidden subgraphs. Then, the result of Childs–Kothari implies that the quantum query complexity of the k-VERTEX COVER problem is $\tilde{O}(n^{3/2-1/(k+2)})$ when k is constant.

2 Concluding Remarks

In this paper, we consider the parameterized quantum query complexities of the k-VERTEX COVER and k-MATCHING problems. We obtain the optimal query complexities for the k-VERTEX COVER and k-MATCHING problems for small k.

For the k-VERTEX COVER problem, our key observation in this paper is that the method of kernelization is useful for designing quantum query algorithms for parameterized problems. We emphasize that this paper is the first work to demonstrate the usefulness of kernelization for the parameterized quantum query complexity. On the other hand, there is a limitation of the techniques based on the method of kernelization. We defined a quantum query kernelization algorithm as an algorithm that converts an input graph given as a quantum oracle into a kernel as a bit string. Here, we observe that on the basis of our quantum query kernelization approach, it is impossible to show that the quantum query complexity of the k-VERTEX COVER problem is $k^{2-\Omega(1)}$ since there is no kernel (G', k') consisting of $O(k^{2-\epsilon})$ edges unless the polynomial hierarchy collapses [33].

It would be interesting to consider quantum-to-quantum kernelization algorithms, which convert an instance given as a quantum oracle into a smaller instance expressed by a quantum oracle.

For the k-MATCHING problem, we designed a quantum query algorithm, which iteratively finds an augmenting path or a vertex with at most one neighbor outside of the current maximal matching. While our k-matching algorithm is optimal for small k, its query complexity is $\Theta(n^2)$ for $k = \Omega(n)$. On the other hand, the currently known best non-parameterized upper bound is $O(n^{7/4})$ [49]. We would be able to improve the parameterized quantum query complexity for the k-MATCHING problem for large k by combining our approach with existing or new approaches.

References

1 Andris Ambainis. Quantum lower bounds by quantum arguments. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing (STOC 2000)*, pages 636–643, 2000. doi:10.1145/335305.335394.

2 Andris Ambainis, Kaspars Balodis, Jānis Iraids, Martins Kokainis, Krišjānis Prūsis, and Jevgēnijs Vihrovs. Quantum speedups for exponential-time dynamic programming algorithms. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2019)*, pages 1783–1793. SIAM, 2019. doi:10.1137/1.9781611975482.107.

3 Andris Ambainis, Kaspars Balodis, Jānis Iraids, Raitis Ozols, and Juris Smotrovs. Parameterized quantum query complexity of graph collision. arXiv preprint arXiv:1305.1021, 2013. doi:10.48550/arXiv.1305.1021.

4 Andris Ambainis, Andrew M Childs, and Yi-Kai Liu. Quantum property testing for bounded-degree graphs. In *Proceedings of the 14th International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX 2011)*, pages 365–376. Springer, 2011. doi:10.1007/978-3-642-22935-0_31.

5 Andris Ambainis, Kazuo Iwama, Masaki Nakanishi, Harumichi Nishimura, Rudy Raymond, Seiichiro Tani, and Shigeru Yamashita. Quantum query complexity of boolean functions with small on-sets. In *Proceedings of the 19th International Symposium on Algorithms and Computation (ISAAC 2008)*, pages 907–918. Springer, 2008. doi:10.1007/978-3-540-92182-0_79.

6 Noel T Anderson, Jay-U Chung, Shelby Kimmel, Da-Yeon Koh, and Xiaohan Ye. Improved quantum query complexity on easier inputs. *Quantum*, 8:1309, 2024. doi:10.22331/q-2024-04-08-1309.

7 Simon Apers and Ronald de Wolf. Quantum speedup for graph sparsification, cut approximation, and Laplacian solving. *SIAM Journal on Computing*, 51(6):1703–1742, 2022. doi:10.1137/21m1391018.

8 Simon Apers and Troy Lee. Quantum complexity of minimum cut. In *Proceedings of the 36th Computational Complexity Conference (CCC 2021)*, volume 200, pages 28:1–28:33, 2021. doi:10.4230/LIPIcs.CCC.2021.28.

9 Agnis Āriņš. Span-program-based quantum algorithms for graph bipartiteness and connectivity. In *International Doctoral Workshop on Mathematical and Engineering Methods in Computer Science*, pages 35–41. Springer, 2015. doi:10.1007/978-3-319-29817-7_4.

10 R Balasubramanian, Michael R Fellows, and Venkatesh Raman. An improved fixed-parameter algorithm for vertex cover. *Information Processing Letters*, 65(3):163–168, 1998. doi:10.1016/S0020-0190(97)00213-5.

11 Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM (JACM)*, 48(4):778–797, 2001. doi:10.1145/502090.502097.

12 Salman Beigi and Leila Taghavi. Quantum speedup based on classical decision trees. *Quantum*, 4:241, 2020. doi:10.22331/q-2020-03-02-241.

13 Aleksandrs Belovs. Learning-graph-based quantum algorithm for k-distinctness. In *Proceedings of the 53rd Annual Symposium on Foundations of Computer Science (FOCS 2012)*, pages 207–216. IEEE, 2012. doi:10.1109/FOCS.2012.18.

14 Aleksandrs Belovs. Span programs for functions with constant-sized 1-certificates. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC 2012)*, pages 77–84, 2012. doi:10.1145/2213977.2213985.

15 Aleksandrs Belovs and Ben W Reichardt. Span programs and quantum algorithms for st-connectivity and claw detection. In *Proceedings of the 20th Annual European Symposium (ESA 2012)*, pages 193–204. Springer, 2012. doi:10.1007/978-3-642-33090-2_18.

16 Ran Ben-Basat, Ken-ichi Kawarabayashi, and Gregory Schwartzman. Parameterized distributed algorithms. In *Proceedings of the 33rd International Symposium on Distributed Computing (DISC 2019)*, volume 146, pages 6:1–6:16, 2019. doi:10.4230/LIPIcs.DISC.2019.6.

17 Aija Berzina, Andrej Dubrovsky, Rusins Freivalds, Lelde Lace, and Oksana Scegulnaja. Quantum query complexity for some graph problems. In *Proceedings of the 30th Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2004)*, pages 140–150. Springer, 2004. doi:10.1007/978-3-540-24618-3_11.

18 Joakim Blikstad, Jan Van Den Brand, Yuval Efron, Sagnik Mukhopadhyay, and Danupon Nanongkai. Nearly optimal communication and query complexity of bipartite matching. In *Proceedings of the 63rd Annual Symposium on Foundations of Computer Science (FOCS 2022)*, pages 1174–1185. IEEE, 2022. doi:10.1109/FOCS54457.2022.00113.

19 Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. Tight bounds on quantum searching. *Fortschritte der Physik: Progress of Physics*, 46(4-5):493–505, 1998. doi:10.1002/(SICI)1521-3978(199806)46:4/5<493::AID-PROP493>3.0.CO;2-P.

20 Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002. doi:10.1090/conm/305.

21 Jonathan F Buss and Judy Goldsmith. Nondeterminism within P^*. *SIAM Journal on Computing*, 22(3):560–572, 1993. doi:10.1137/0222038.

22 Chris Cade, Ashley Montanaro, and Aleksandrs Belovs. Time and space efficient quantum algorithms for detecting cycles and testing bipartiteness. *Quantum Information and Computation*, 18(1-2):18–50, 2018. doi:10.26421/QIC18.1-2-2.

23 Titouan Carette, Mathieu Laurière, and Frédéric Magniez. Extended learning graphs for triangle finding. *Algorithmica*, 82(4):980–1005, 2020. doi:10.1007/s00453-019-00627-z.

24 L Sunil Chandran and Fabrizio Grandoni. Refined memorisation for vertex cover. In *Proceedings of the 1st International Workshop on Parameterized and Exact Computation (IWPEC 2004)*, pages 61–70. Springer, 2004. doi:10.1007/978-3-540-28639-4_6.

25 Jianer Chen, Iyad A Kanj, and Weijia Jia. Vertex cover: Further observations and further improvements. *Journal of Algorithms*, 41(2):280–301, 2001. doi:10.1006/jagm.2001.1186.

26 Jianer Chen, Iyad A Kanj, and Ge Xia. Improved upper bounds for vertex cover. *Theoretical Computer Science*, 411(40-42):3736–3756, 2010. doi:10.1016/j.tcs.2010.06.026.

27 Andrew M Childs and Jason M Eisenberg. Quantum algorithms for subset finding. *Quantum Information and Computation*, 5(7):593–604, 2005. doi:10.26421/QIC5.7-7.

28 Andrew M Childs and Robin Kothari. Quantum query complexity of minor-closed graph properties. *SIAM Journal on Computing*, 41(6):1426–1450, 2012. doi:10.1137/110833026.

29 Rajesh Chitnis and Graham Cormode. Towards a theory of parameterized streaming algorithm. In *Proceedings of the 14th International Symposium on Parameterized and Exact Computation (IPEC 2019)*, volume 148, pages 7:1–7:15, 2019. doi:10.4230/LIPIcs.IPEC.2019.7.

30 Rajesh Chitnis, Graham Cormode, Hossein Esfandiari, MohammadTaghi Hajiaghayi, Andrew McGregor, Morteza Monemizadeh, and Sofya Vorotnikova. Kernelization via sampling with applications to finding matchings and related problems in dynamic graph streams. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2016)*, pages 1326–1344. SIAM, 2016. doi:10.1137/1.9781611974331.ch92.

31 Rajesh Chitnis, Graham Cormode, MohammadTaghi Hajiaghayi, and Morteza Monemizadeh. Parameterized streaming: Maximal matching and vertex cover. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2014)*, pages 1234–1251. SIAM, 2014. doi:10.1137/1.9781611973730.82.

32 Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*, volume 4. Springer, 2015. doi:10.1007/978-3-319-21275-3.

33 Holger Dell and Dieter Van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. *Journal of the ACM (JACM)*, 61(4):1–27, 2014. doi:10.1145/2629620.

34 Michael J Dinneen and Rongwei Lai. Properties of vertex cover obstructions. *Discrete Mathematics*, 307(21):2484–2500, 2007. doi:10.1016/j.disc.2007.01.003.

35 Sebastian Dörn. Quantum complexity bounds for independent set problems. arXiv preprint quant-ph/0510084, 2005. doi:10.48550/arXiv.quant-ph/0510084.

36 Sebastian Dörn. Quantum algorithms for matching problems. *Theory of Computing Systems*, 45(3):613–628, 2009. doi:10.1007/s00224-008-9118-x.

37 Rodney G Downey and Michael R Fellows. Fixed-parameter tractability and completeness. In *Complexity Theory: Current Research*, pages 191–225, 1992. doi:10.1109/SCT.1992.215379.

38 Christoph Dürr, Mark Heiligman, Peter Høyer, and Mehdi Mhalla. Quantum query complexity of some graph problems. *SIAM Journal on Computing*, 35(6):1310–1328, 2006. doi:10.1137/050644719.

39 Stefan Fafianie and Stefan Kratsch. Streaming kernelization. In *Proceedings of the 39th International Symposium on Mathematical Foundations of Computer Science (MFCS 2014)*, pages 275–286. Springer, 2014. doi:10.1007/978-3-662-44465-8_24.

40 Fedor V Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. *Kernelization: Theory of Parameterized Preprocessing*. Cambridge University Press, 2019. doi:10.1017/9781107415157.

41 Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC 1996)*, pages 212–219, 1996. doi:10.1145/237814.237866.

42 Jiong Guo and Rolf Niedermeier. Invitation to data reduction and problem kernelization. *ACM SIGACT News*, 38(1):31–45, 2007. doi:10.1145/1233481.1233493.

43 David G. Harris and N. S. Narayanaswamy. A faster algorithm for vertex cover parameterized by solution size. In *Proceedings of the 41st International Symposium on Theoretical Aspects of Computer Science (STACS 2024)*, volume 289, pages 40:1–40:18, 2024. doi:10.4230/LIPIcs.STACS.2024.40.

44 John E Hopcroft and Richard M Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, 1973. doi:10.1137/0202019.

45 Michael Jarret, Stacey Jeffery, Shelby Kimmel, and Alvaro Piedrafita. Quantum algorithms for connectivity and related problems. In *Proceedings of the 26th Annual European Symposium on Algorithms (ESA 2018)*, volume 112, pages 49:1–49:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ESA.2018.49.

46 Stacey Jeffery, Shelby Kimmel, and Alvaro Piedrafita. Quantum algorithm for path-edge sampling. In *Proceedings of the 18th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2023)*, volume 266, pages 5:1–5:28, 2023. doi:10.4230/LIPIcs.TQC.2023.5.

47 Stacey Jeffery, Robin Kothari, and Frédéric Magniez. Nested quantum walks with quantum data structures. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2013)*, pages 1474–1485. SIAM, 2013. doi:10.1137/1.9781611973105.106.

48 Richard M Karp. *Reducibility Among Combinatorial Problems*. Springer, 2010. doi:10.1007/978-3-540-68279-0_8.

49 Shelby Kimmel and R Teal Witter. A query-efficient quantum algorithm for maximum matching on general graphs. In *Proceedings of the 17th International Symposium on Algorithms and Data Structures (WADS 2021)*, pages 543–555, 2021. doi:10.1007/978-3-030-83508-8_39.

50 Bernhard H Korte and Jens Vygen. *Combinatorial Optimization*. Springer, sixth edition, 2006. doi:10.1007/978-3-662-56039-6.

51 François Le Gall. Improved quantum algorithm for triangle finding via combinatorial arguments. In *Proceedings of the 55th Annual Symposium on Foundations of Computer Science (FOCS 2014)*, pages 216–225. IEEE, 2014. doi:10.1109/FOCS.2014.31.

52 François Le Gall and Shogo Nakajima. Quantum algorithm for triangle finding in sparse graphs. *Algorithmica*, 79:941–959, 2017. doi:10.1007/s00453-016-0267-z.

53 François Le Gall and Saeed Seddighin. Quantum meets fine-grained complexity: Sublinear time quantum algorithms for string problems. *Algorithmica*, 85(5):1251–1286, 2023. doi:10.1007/s00453-022-01066-z.

54 Troy Lee, Frédéric Magniez, and Miklos Santha. Learning graph based quantum query algorithms for finding constant-size subgraphs. *Chicago Journal of Theoretical Computer Science*, 10:1–21, 2012. doi:10.4086/cjtcs.2012.010.

55 Troy Lee, Frédéric Magniez, and Miklos Santha. Improved quantum query algorithms for triangle finding and associativity testing. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2013)*, pages 1486–1502. SIAM, 2013. doi:10.1137/1.9781611973105.107.

56 Cedric Yen-Yu Lin and Han-Hsuan Lin. Upper bounds on quantum query complexity inspired by the Elitzur–Vaidman bomb tester. *Theory of Computing*, 12(18):1–35, 2016. doi:10.4086/toc.2016.v012a018.

57 Frédéric Magniez, Miklos Santha, and Mario Szegedy. Quantum algorithms for the triangle problem. *SIAM Journal on Computing*, 37(2):413–424, 2007. doi:10.1137/050643684.

58 Rolf Niedermeier and Peter Rossmanith. Upper bounds for vertex cover further improved. In *Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science (STACS 1999)*, pages 561–570. Springer, 1999. doi:10.1007/3-540-49116-3_53.

59 Rolf Niedermeier and Peter Rossmanith. On efficient fixed-parameter algorithms for weighted vertex cover. *Journal of Algorithms*, 47(2):63–77, 2003. doi:10.1016/S0196-6774(03)00005-1.

60 Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume 24. Springer, 2003.

61 Ulrike Stege and Michael Ralph Fellows. An improved fixed parameter tractable algorithm for vertex cover. *Technical report/Departement Informatik, ETH Zürich*, 318, 1999. doi:10.3929/ethz-a-006653305.

62 Xiaoming Sun, Andrew C Yao, and Shengyu Zhang. Graph properties and circular functions: How low can quantum query complexity go? In *Proceedings of the 19th IEEE Annual Conference on Computational Complexity (CCC 2004)*, pages 286–293. IEEE, 2004. doi:10.1109/CCC.2004.1313851.

63 Shengyu Zhang. On the power of Ambainis's lower bounds. In *Proceedings of the 31st International Colloquium on Automata, Languages and Programming (ICALP 2004)*, pages 1238–1250. Springer, 2004. doi:10.1007/978-3-540-27836-8_102.

64 Yechao Zhu. Quantum query complexity of constant-sized subgraph containment. *International Journal of Quantum Information*, 10(03):1250019, 2012. doi:10.1142/S0219749912500190.