Vertex coloring and related problems in the quantum query model

Jackson Morris
Department of Mathematics, University of California Los Angeles
jrexmo@ucla.edu

Fang Song
Department of Computer Science, Portland State University
fsong@pdx.edu

Abstract

Given a graph G with n vertices and maximum degree Δ, it is known that G admits a vertex coloring with $\Delta + 1$ colors such that no edge of G is monochromatic. This can be seen constructively by a simple greedy algorithm, which runs in time $O(n\Delta)$. Very recently, [Assdi et. al. SODA'19] presents a randomized algorithm for $\Delta + 1$-coloring in the query model making $\tilde{O}(n\sqrt{n})$ queries, improving over the greedy strategy. In addition, a lower bound of $\Omega(n\sqrt{n})$ for any $O(\Delta)$-coloring, including $\Delta + 1$-coloring, is established on general graphs.

The main result of this paper is a quantum algorithm in the query model that bypasses the classical lower bound. Specifically for any $\delta > 0$ satisfying $\delta^{-1} = O(1)$, our algorithm makes $\tilde{O}(\epsilon^{-3/2} n^{4/3+\delta/2})$ quantum queries and returns a valid $(1+\epsilon)\Delta$-coloring with high probability. By similar techniques, we also give a quantum algorithm form maximal-matching in the quantum query model that makes $\tilde{O}(n^{3/2+\delta/2})$ queries, bypassing the classical lower bound $\Omega(n^2)$. Complementary to these algorithmic results, we prove quantum lower bounds of $\Omega(n)$ for both 2Δ-coloring and maximal matching.

2012 ACM Subject Classification Theory of computation → Quantum computation theory

Keywords and phrases graph coloring, quantum algorithms, query model

1 Introduction

Graph coloring is a fundamental problems in discrete algorithms and graph theory. It has wide applications in scheduling, resource allocation, compiler optimization, and logistics settings. In this problem, one aims to assign every vertex a color such that no edge is *monochromatic*, i.e., both endpoints having the same color. Finding such a valid coloring with $k \geq 1$ colors is usually called the k-coloring problem. A critical graph parameter $\chi(G)$ is the minimum number of colors necessary to guarantee a valid coloring. In Karp's renowned result, 3-coloring as well as deciding $\chi(G)$ in general are proven to be NP-complete [8]. On the other hand, it is known that for any graph G, $\chi(G) \leq \Delta + 1$ where Δ is the maximum degree of G. This follows from a simple greedy algorithm for $\Delta + 1$-coloring. The basic observation is that any valid partial $\Delta + 1$-coloring of G can be extended to a complete $\Delta + 1$-coloring. This is because every vertex has at most Δ neighbors, there must be at least one choice of color for that vertex that does not conflict with any of its neighbors.

For quite some time, the $O(n\Delta)$ greedy algorithm is the best known algorithm for $\Delta + 1$ vertex coloring on general graphs. Very recently, authors of [2] give a new $\tilde{O}(n\sqrt{n})$ algorithm in the *query* model, where the graph's adjacency matrix is given as a black-box. The same paper also establishes an almost matching lower bound of $\Omega(n\sqrt{n})$, implying that their algorithm is optimal up to polylog(n) factors. The results in [2] actually apply more generally to any $O(\Delta)$ coloring, not just $\Delta + 1$.

In the related *maximal-matching* problem, we wish to find a set of edges $M \subset E(G)$ which are pairwise disjoint and cannot be extended. Namely there does not exist $M' \subset E(G)$ satisfying the pairwise disjointness property and strictly containing M. There is a simple greedy algorithm for maximal matching which has time and query complexity $O(n^2)$ [2]. The authors of [2] also prove a $\Omega(n^2)$ lower bound in the graph query model for maximal-matching. In effect, this lower bound shows that there is no hope of finding a sublinear (in $|E| = m$) algorithm in the graph query model for the problem of maximal-matching.

Our Contribution. In this work, we study solving these graph problems on a quantum computer in a standard quantum query model, where the adjacency matrix may be queried in quantum superposition. We give quantum algorithms for coloring and maximal matching that bypass the classical lower bounds in [2]. The most appealing feature of our algorithms is their simplicity. Basically, we are inspired by the $\Delta + 1$ greedy algorithm, where the algorithms progresses by extending a partial solution *locally*. We describe a simple randomized algorithm for $(1 + \epsilon)\Delta$-coloring, which is suitable to apply quantum unstructured search techniques in each "local" search subroutine. In particular, we do not need the heavy machinery in [2] such as powerful techniques on palette sparsification and list-coloring. We give a brief overview of our results below, and see also Table 1 for a summary.

Result 1: A randomized algorithm for $(1 + \epsilon)\Delta$-coloring which makes $O(\epsilon^{-1}n\sqrt{n})$ queries in expectation and always returns a valid coloring.

This algorithm uses the same general "local search" strategy that we will utilize in each of the algorithms presented and is surprisingly efficient. In fact, this algorithm has optimal expected queries complexity when $\epsilon = O(1)$ as we will prove in corollary 3.1. When $\epsilon = 1$ this gives a 2Δ-coloring algorithm with optimal expected query complexity. Developing this algorithm serves as a basic template for later results.

A central subroutine is what we call **Find-Conflict**. It takes a vertex v, a set of vertices

S such that $v \notin S$, and a parameter $\delta > 0$ and uses the graph oracle to determine if there are any edges of the form (u, v) actually present in $E(G)$ for any of the $u \in S$. This then helps us determine which colors are valid for v or if there are any edges (u, v) that can be included in our maximal matching, depending on the problem. The parameter δ allows for us to achieve arbitrarily high success probability.

Result 2: A quantum algorithm which for any $\delta > 0$ makes $\tilde{O}(\epsilon^{-3/2} n^{4/3+\delta/2})$ queries and returns a valid $(1+\epsilon)\Delta$-coloring with high probability.

Achieving this result essentially has the same strategy as that of result 1, while noting that **Find-Conflict** can be instantiated by some quantum search algorithm (variants on Grover's algorithm) [11, 13]. The analysis nonetheless requires more care to carefully contain the errors. First, we show that our method is valid for 2Δ colors and then extend the algorithm to the case of $(1+\epsilon)\Delta$ colors for $\epsilon \in [\frac{2}{\Delta}, 1]$.

We remark that although we rely on known quantum algorithmic techniques, it is crucial that they get employed in the right place. For instance, one may be tempted to assume that such for coloring we could somehow perform a search over just the color palette, thus requiring $O(\sqrt{\Delta} \log \Delta)$ queries, but such a search would not be directly feasible with just the adjacency matrix queries or neighbor used in this model. Instead, one would need some kind of "color validity" oracles for every vertex that could be dynamically updated as various colors become unavailable for some vertices. Maintaining these oracles would require an overhead that would lead to superlinear coloring algorithms.

Result 3: A quantum algorithm for maximal matching, which makes $\tilde{O}(n^{3/2+\delta/2})$ queries and returns a matching which is maximal with high probability.

The strategy here is very similar to that of the coloring algorithms as **Find-Conflict** is at the heart of the speed-up.

Result 4: Quantum query lower bound of $\Omega(n)$ for $1 + \Delta$-coloring, 2Δ-coloring, and maximal matching.

This is shown by concocting specific graph instances and reducing the unstructured search problem to them. Therefore the $\Omega(\sqrt{n})$ quantum query lower bound for unstructured search [4] will transfer.

Problem	Quantum Algorithm	Classical Algorithm	Classical Lower Bound
2Δ-coloring	$\tilde{O}(n^{4/3+\delta/2})$	$O(n\sqrt{n})^*$, $\tilde{O}(n\sqrt{n})$ [2]	$\Omega(n\sqrt{n})$ [2]
$(1+\epsilon)\Delta$-coloring	$\tilde{O}(\epsilon^{-3/2} n^{4/3+\delta/2})$	$O(\epsilon^{-1} n\sqrt{n})^*$, $\tilde{O}(n\sqrt{n})$ [1]	$\Omega(n\sqrt{n})$ [2]
Maximal Matching	$\tilde{O}(n^{3/2+\delta/2})$	$O(n^2)$ [Greedy]	$\Omega(n^2)$ [2]

Table 1 Summary of results. * indicates expected number of queries and no reference indicates the result is from this paper. We also show an $\Omega(n)$ lower bound for quantum algorithms on each of these problems.

Further discussion. One immediate problem left open by our results in to close the gap between the algorithmic bounds and the lower bound. Some other well studied graph problems that are likely to admit similar quantum speedups are variants of coloring (defective coloring, edge coloring, etc) and potentially some dynamic graph problems.

A problem that has received considerable study in the quantum query model is triangle finding. Here one aims to either output a triangle if one exists or determine that the graph is triangle free with bounded error. For a general graph this may require $\Omega(n^2)$ queries to the adjacency matrix oracle. However, a modified Grover search can be used, searching over triples of vertices, to achieve an $\tilde{O}(n\sqrt{n})$ quantum algorithm for triangle finding as in [6]. More sophisticated arguments in [7] improve upon this resulting, bring the query complexity down to $\tilde{O}(n^{5/4})$. Conversely, the best know lower bound for triangle finding in the quantum query model is the immediate $\Omega(n)$ bound established in [3]. Just as in triangle finding, a gap persists between the best known quantum lower bound and the best known quantum algorithm for $O(\Delta)$ coloring as shown in our work.

In addition, the problem of *maximum matching* in which one wishes to find the largest maximal matching, has been studied in the quantum query model. The very recent results of [9] gives an improved quantum algorithm for maximum matching in the adjacency matrix model which makes $O(n^{7/4})$ queries. However, a lower bound of $\Omega(n^{3/2})$ queries in the adjacency matrix model established in [5] show that a gap persists between the best known algorithm and lower bound for this problem as well. Other structural graph problems such as those relating to connected components and spanning forests have also been studied. A recent work [10], surprisingly, shows exponential quantum speedup for these problem, assuming a more sophisticated oracle model that can answer "cut queries".

2 Preliminaries

In this paper the notation $[n]$ will refer to the set $\{1, 2, \ldots n\}$ for any positive integer n. For a positive integer L, the L-coloring problem is as follows: given a graph $G = (V, E)$ and $n = |V|$ we wish to find an assignment $c : [L] = \{1, \ldots L\} \to V$ such that for any edge $(x, y) \in E$ we have $c(x) \neq c(y)$. L will be referred to as the palette throughout. Such an assignment is called a *valid L-coloring of G*. Let Δ be the maximum degree of any vertex in G. As previously stated a simple greedy algorithm can be shown to always produce a valid coloring and runs in time $O(n\Delta)$. Rather than time complexity, the results in this paper will mostly be stated in terms of query complexity. The main models of computation in this paper are the graph query model and the quantum graph query model. In the standard graph query model, we assume that n and the degrees of the vertices is the only knowledge available about G before any queries have been made. The standard query model for graphs supports the following types of queries:

- Pair Queries: we can query the oracle if the edge (v_i, v_j) is present in the graph for any $i, j \in \{1, \ldots n\}$. This will be denoted as

$$M[v_i, v_j] = \begin{cases} 1 & (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}$$

- Neighbor Queries: we can query the oracle for the jth neighbor of vertex v_i for any $i \in \{1, \ldots n\}$ and $j \in \{1, \ldots \deg v_i\}$

The classic greedy $\Delta + 1$-coloring algorithm can easily be implemented in the graph query model via neighbor queries. Explicitly, we can determine which colors are valid at any vertex $v \in V$ with $\deg v$ neighbor queries.

In the quantum graph query model we work over the $n(n-1)$-qubit system which includes all possible pairs of indices, e.g. for every (i, j) with $i, j \in [n]$ and $i \neq j$ the qubits $|ij\rangle$. Given a super position of k distinct edge pairs $\{(u_i, v_i)\}_{i=1}^{k}$, say $\sum_{i=1}^{k} a_i |u_i v_i\rangle$ the oracle acts as follows:

$$|s\rangle = \sum_{i=1}^{k} a_i |u_i v_i\rangle \to \sum_{i=1}^{k} (-1)^{M[u_i,v_i]} a_i |u_i v_i\rangle$$

The oracle transformation will be denoted as O_M and a single application of this unitary transformation will be referred to as a quantum query.

Note that this quantum query model is equivalent to the standard black box quantum query model where our binary function $f : \{(u, v) \mid u, v \in V\} \to \{0, 1\}$ indicates which edges are present.

Throughout this paper we use the phrase *with high probability* to mean with probability at least $1 - \frac{1}{n^k}$ for a sufficiently large constant k.

3 Algorithmic Results

In this section we give a simple optimal randomized algorithm for $(1 + \epsilon)\Delta$-Coloring. This algorithm serves as a "warm-up" and as the inspiration for the quantum algorithms presented later. We start by transforming a bound on Monte Carlo algorithms for $O(\Delta)$-Coloring to one on Las Vegas algorithms for the same class of problems. From the lower bounds established in [2]:

▶ **Corollary 1.** *Any randomized algorithm which always returns a valid 2Δ coloring requires $\Omega(n\sqrt{n})$ queries in expectation.*

Proof of corollary 1. Lemma 5.6 of [2] implies that any Monte Carlo algorithm for 2Δ coloring which makes fewer than $\frac{n\sqrt{n}}{400000}$ queries returns an invalid coloring with probability at least $\frac{1}{4}$. So, suppose that some Las Vegas algorithm which makes $O(\frac{n^{3/2}}{T(n)})$ queries in expectation for some $T : \mathbb{N} \to \mathbb{N}$ with $\lim_{n\to\infty} T(n) = \infty$. Now, consider a Monte Carlo variant of this algorithm which runs until a valid 2Δ-coloring is found or until $\frac{n\sqrt{n}}{500000}$ queries have been made. The probability that this Monte Carlo variant fails to return a valid coloring is equal to the probability that the Las Vegas algorithm makes $\frac{n\sqrt{n}}{500000}$ queries or more on the same input. Let Q denote the number of queries made by the Las Vegas algorithm. From Markov's inequality, we have

$$\begin{aligned}\mathbb{P}(Q \geq \frac{n\sqrt{n}}{500000}) &\leq \frac{\mathbb{E}[Q]}{a} \\ &\leq \frac{n\sqrt{n}/T(n)}{n\sqrt{n}/500000} \\ &= \frac{500000}{T(n)}\end{aligned}$$

Since $T(n) \to \infty$ as $n \to \infty$, there must exist some N such that for all $n > N$, $T(n) \geq 5000000$ and therefore for $n > N$

$$\mathbb{P}(Q \geq \frac{n\sqrt{n}}{500000}) \leq \frac{500000}{T(n)} \leq \frac{1}{10}$$

Thus, this Monte-Carlo variant produces a valid 2Δ-coloring with probability at least $\frac{9}{10}$ and makes at most $\frac{n\sqrt{n}}{5000000}$ queries for sufficiently large n. This contradicts lemma 5.6 of [2], proving the stated lower bound. ◀

This result will guide us in our search for randomized algorithms.

3.1 Randomized $(1+\epsilon)\Delta$-Coloring

We begin this section by providing a Las Vegas algorithm for $(1+\epsilon)\Delta$-coloring in the standard query model.

▶ **Theorem 2.** *There exists a randomized algorithm which returns a valid $(1+\epsilon)\Delta$ vertex coloring and makes $O(\epsilon^{-1}n\sqrt{n})$ queries in expectation.*

Proof of theorem 2. Recall that the greedy algorithm for $\Delta+1$-coloring algorithm makes $O(n\Delta)$ queries, so when $\Delta \ll n$ this algorithm performs very well. This helps us on our search for sublinear algorithms since we can use this just greedy algorithm when Δ is sufficiently small, allowing us to restrict our attention to larger Δ i.e. $\Delta \geq \sqrt{n}$.

Both the classical and quantum coloring algorithms in this paper will have the same general strategy: when coloring a vertex choose a random color, test whether that color is a valid choice, if it is we move on, otherwise repeat. While this is a simple approach with some obvious limitations, it yields some surprisingly efficient algorithms. The only data structure we maintain in all of our algorithms is defined as follows:

- For every $c \in [(1+\epsilon)\Delta]$ we maintain a set called $\chi(c)$ of all vertices which we have been colored with color c. Let $\chi_t(c)$ denote the collection of vertices with color c after having colored $t-1$ vertices.

Our randomzied algorithm is as follows:

Algorithm 1 $(1+\epsilon)\Delta$-Color

```
for v ∈ V
    while v is not colored
        Choose a color c uniformly at random from [(1 + ε)Δ]
        for u ∈ χ(c)
            Query M[u, v]
            if M[u, v] = 1
                break choose a new color
        Assign color c to v and update χ(c) with v
```

Note that this algorithm always returns a valid $(1+\epsilon)\Delta$-coloring since a color $c \in [(1+\epsilon)\Delta]$ is only ever assigned to v if no edge of the form (u,v) for $u \in \chi(c)$ is present in the graph. Every vertex has at least one valid color since a vertex v has at most Δ neighbors and therefore at least $(1+\epsilon)\Delta - \Delta = \epsilon\Delta \geq 1$ valid colors.

Now, we will show that $(1+\epsilon)\Delta$-Color makes $O(\epsilon^{-1}n\sqrt{n})$ queries in expectation. Let $Q(G)$ denote the number of queries made by this algorithm for a graph G and let Q_t denote the number of queries made when coloring vertex v_t. Observe that

$$\mathbb{E}[Q(G)] = \mathbb{E}\left(\sum_{t=1}^{n} Q_t\right) = \sum_{t=1}^{n} \mathbb{E}[Q_t]$$

So, bounding $\mathbb{E}[Q_t]$ will allow us to bound the total expected number of queries.

▶ **Lemma 3.** *For $\Delta \geq \sqrt{n}$ and any $t \leq n$*

$$\mathbb{E}[Q_t] = O(\epsilon^{-1}\sqrt{n})$$

Proof of lemma 3. Let c be the first color that is chosen for the vertex we are currently looking at. There are two possibilities: 1) c is valid at v_t or 2) it is not. Let $p =$

$\mathbb{P}[c \text{ is valid at } v_t]$, we have:

$$\begin{aligned}\mathbb{E}[Q_t] &= p\mathbb{E}[|\chi(c)| \ : \ c \text{ is valid at } v_t] + (1-p)(\mathbb{E}[|\chi(c)| \ : \ c \text{ is invalid at } v_t] + \mathbb{E}[Q_t]) \\ &= p\mathbb{E}[|\chi(c)| \ : \ c \text{ is valid at } v_t] + (1-p)\mathbb{E}[|\chi(c)| \ : \ c \text{ is invalid at } v_t] + (1-p)\mathbb{E}[Q_t] \\ &= \mathbb{E}[|\chi(c)|] + (1-p)\mathbb{E}[Q_t]\end{aligned}$$

When c is chosen uniformly at random from $[(1+\epsilon)\Delta]$, $\mathbb{E}[|\chi(c)|] \leq \frac{n}{(1+\epsilon)\Delta}$. Also, recall that there are at least $\epsilon\Delta$ colors valid colors for v_t, so $p \geq \frac{\epsilon\Delta}{(1+\epsilon)\Delta} = \frac{\epsilon}{1+\epsilon}$. From these facts it follows that

$$\begin{aligned}\mathbb{E}[Q_t] &= \frac{1}{p}\mathbb{E}[|\chi(c)|] \\ &\leq \frac{1+\epsilon}{\epsilon}\frac{n}{(1+\epsilon)\Delta} \\ &= \frac{n}{\epsilon\Delta}\end{aligned}$$

Hence, when $\Delta \geq \sqrt{n}$

$$\mathbb{E}[Q_t] \leq \epsilon^{-1}\sqrt{n}$$

Thus proving Lemma 3. ◀

From Lemma 3 it follows quite easily that

$$\sum_{t=1}^{n} \mathbb{E}[Q_t] = O(\epsilon^{-1}n\sqrt{n})$$

So, $(1+\epsilon)$-Color makes $O(\epsilon^{-1}n\sqrt{n})$ queries in expectation, thus proving Theorem 2. ◀

Whenever $\epsilon^{-1} = O(1)$ we have that $\mathbb{E}[Q(G)] = O(n\sqrt{n})$. Hence this algorithm achieves the query lower bound for Las Vegas algorithms established in corollary 1 for sufficiently large values of ϵ. This algorithm is not useful when $\epsilon^{-1} = \Delta$, so finding an $O(n\sqrt{n})$ algorithm (Las Vegas or Monte Carlo) for $\Delta + 1$ coloring remains as an open problem.

Of course, when $\epsilon = 1$ we have a 2Δ-Coloring algorithm which makes $O(n\sqrt{n})$ queries. The 2Δ-coloring problem will serve as a baseline for $O(\Delta)$ coloring as seen in the next section.

4 Quantum Algorithms

The simple randomized algorithm explored in the previous section involved repeatedly solving the following sub-problem:

Given a vertex $v \in V$ and a collection of vertices $S \subset V \setminus v$ determine if $\{(u,v) : u \in S\} \cap E(G) = \emptyset$. In other words, does v have any neighbors in S?

In effect, this is just an instance of unstructured search. However, we do not know how many (if any) such edges are actually present, so a standard Grover search algorithm will not work here. This is referred to as the soufflé problem in [13]. However, there are techniques for overcoming this.

▷ Claim 4. For any $\delta > 0$ there is a quantum algorithm which given a set of vertices S and $v \in V \setminus S$ determines whether or not v is adjacent to any vertex in S and returns such a vertex in S if it exists using $O(\sqrt{n^\delta|S|}\log n)$ queries to the oracle with failure probability $O((n^\delta|S|)^{-1})$.

Proof of claim 4. Algorithm 2 of [12] gives a method of finding one marked element (if one exists) where the total number of unmarked elements is unknown. The number of queries used by this algorithm is $O(\sqrt{n}\log n)$ in the worst case and if there are marked elements then one is returned with probability $1-O(1/n)$. When this algorithm is run with the search space of edges defined by edges of the form (v,u) for $u \in S$, and the adjacency matrix oracle we are able to determine if a neighbor exists with probability $1-O(1/|S|)$ and $O(\sqrt{|S|}\log|S|)$ queries. In order to satisfy the claim we can pad the uniform super-position with a ancilla qubits such that $a+|S| = O(n^\delta|S|)$ - when we do this the error probability is $O(\frac{1}{n^\delta|S|})$ and $O(\sqrt{|S|n^\delta}\log n^\delta|S|)$ queries are made. This modified version of Grover Search is sometimes reffered to as "safe Grover search" [11]. ◀

We will call this subroutine **Find-Conflict**(v,S,δ) - see [12] or Appendix A for an explicit description of this algorithm. If v truly has a neighbor in S, then **Find-Conflict**(v,S,δ) returns a vertex $u \in S$ with probability $1-O(n^{-\delta})$ if v, but if v has no such neighbor then **false** is always returned. Additionally, let $C > 0$ be a constant such that **Find-Conflict** makes at most $C\sqrt{n^\delta|S|}\log n$ queries.

In $(1+\epsilon)\Delta$-Color we needed to spend $|\chi(c)|$ queries in the worst case in order to determine if c is valid at v, but by using modified Grover Search techniques we are able to achieve a quantum speed up. The first result using these techniques is as follows:

▶ **Theorem 5.** *For any $\delta > 0$ there exists a quantum algorithm which returns a valid $2\Delta-$vertex coloring with high probability and makes $\tilde{O}(\sqrt{\frac{n^{3+\delta}}{\Delta}})$ queries.*

Proof of theorem 5. Consider the algorithm below:

Algorithm 2 Quantum-$2\Delta-$Color$(G(V,E),\delta,k)$

```
while fewer than 6Ck⌈k/δ⌉ log n √(n^{3+δ}/Δ) queries have been made
    for v ∈ V:
        while v is not colored
            Choose a color c uniformly at random from [2Δ]
            if |χ(c)| > 2n/Δ
                break choose a new color
            for j = 1 to 2⌈k/δ⌉
                a ← Find-Conflict(v, χ(c), δ)
                if a is not false
                    break choose a new color
            Assign c to v and update χ(c)
```

▶ **Lemma 6.** *Quantum-2Δ-Color returns valid 2Δ-coloring of G with high probability.*

Proof of lemma 6. First, note that that in this algorithm **Find-Conflict** is only ever called when $|\chi(c)| \le \frac{2n}{\Delta}$, so every call makes at most $C\sqrt{\frac{n^{1+\delta}}{\Delta}}\log n$ queries and has failure probability $O(n^{-\delta})$.

This algorithm only produces an invalid coloring in two cases:

1. Some vertex is assigned an invalid color
2. Not all of the vertices are colored by the time $6Ck\log n\sqrt{\frac{n^{3+\delta}}{\Delta}}$ queries have been made

For the first case, note that any $v \in V$ is assigned an invalid color only when every call of **Find-Conflict** incorrectly returns **false** $2\lceil\frac{k}{\delta}\rceil$ times in a row, this occurs with probability $O(n^{(-\delta)\lceil\frac{2k}{\delta}\rceil}) = O(n^{-2k})$. Thus, if the coloring that Quantum-2Δ-Color returns is complete (does not include any uncolored vertices) then the probability that some vertex is assigned an invalid color is at most

$$\sum_{t=1}^{n} \mathbb{P}(v_t \text{ is assigned an invalid color}) = \sum_{t=1}^{n} O(n^{-2k}) = O(n^{1-2k})\,.$$

Now, we will show that with high probability every vertex is assigned some color before using too many queries. Let Q_t be the number of queries made when attempting to color vertex v_t. Additionally, let K_t be the number of times that a new random color is tested when attempting to color v_t (the number of random colors which are actually tested with **Find-Conflict**). Since every randomly chosen color causes us to make at most $2\lceil\frac{k}{\delta}\rceil C\sqrt{\frac{n^{1+\delta}}{\Delta}}\log n$ queries, it follows that

$$\mathbb{P}(Q_t \geq 6C\lceil\frac{k}{\delta}\rceil k \log^2 n\sqrt{\frac{n^{1+\delta}}{\Delta}}) \leq \mathbb{P}(K_t \geq 3k\log n)$$

Now, note that there are at least Δ valid colors and at least $\frac{\Delta}{1}$ color classes satisfying $|\chi(c)| \leq \frac{2n}{\Delta}$. This means that there are at least $\frac{\Delta}{2}$ colors c which are valid and satisfy $|\chi(c)| \leq \frac{2n}{\Delta}$. Hence, the probability of a randomly chosen color being valid is at least $\frac{\Delta/2}{2\Delta} = \frac{1}{4}$. Using this fact we can bound K_t as follows

$$\mathbb{P}(K_t \geq 3k\log n) \leq \left(1-\frac{1}{4}\right)^{3k\log n} \leq \left(\frac{1}{2}\right)^{k\log n} \leq \frac{1}{n^k}\,.$$

Thus, we can bound the probability that in the end some vertex remains uncolored as follows:

$$\begin{aligned}
\mathbb{P}\left(\sum_{t=1}^{n} Q_t \geq 6C\lceil\frac{k}{\delta}\rceil \log^2 n\sqrt{\frac{n^{3+\delta}}{\Delta}}\right) &\leq \mathbb{P}\left(\bigcup_{t=1}^{n} Q_t \geq 6C\lceil\frac{k}{\delta}\rceil \log^2 n\sqrt{\frac{n^{1+\delta}}{\Delta}}\right)\\
&\leq \sum_{t=1}^{n} \mathbb{P}\left(Q_t \geq 6C\lceil\frac{k}{\delta}\rceil \log^2 n\sqrt{\frac{n^{1+\delta}}{\Delta}}\right)\\
&= O(n^{1-k})
\end{aligned}$$

So, taking both cases 1 & 2 into account, we can see that Quantum-2Δ-Color produces a valid coloring with probability at least

$$1 - O(n^{1-k}) - O(n^{1-2k}) = 1 - O(n^{-(k-1)})$$

and makes $O(\sqrt{\frac{n^{3+\delta}}{\Delta}}\log^2 n)$ queries. This concludes the proof of lemma 6 and thus theorem 5 as well. ◀

◀

Whenever $n^{1/3} \leq \Delta$ this algorithm makes $\tilde{O}(n^{4/3+\delta/2})$ queries and the greedy algorithm suffices whenever $\Delta \leq n^{1/3}$.

A natural extension of Quantum-2Δ-Color is a quantum $(1+\epsilon)\Delta$-Coloring algorithm.

▶ **Theorem 7.** *For any $\delta > 0$, $\epsilon \in [\frac{2}{\Delta}, 1]$, and an input graph G there exists a quantum algorithm which returns a valid $(1+\epsilon)\Delta$ coloring of G with high probability and uses $\tilde{O}(\epsilon^{-3/2}\sqrt{\frac{n^{3+\delta}}{\Delta}})$ queries.*

Proof of theorem 7. Similar to Quantum-2Δ-Color, the algorithm presented in this section will only examine color classes with $\frac{2n}{\epsilon\Delta}$ or fewer colors. **Find-Conflict**$(v, |\chi(c)|, \delta)$ has error probability $O(n^{-\delta})$ and makes at most $C\sqrt{\frac{n^{1+\delta}}{\epsilon\Delta}}\log n$ queries, whenever $|\chi(c)| \leq \frac{2n}{\epsilon\Delta}$.

The complete algorithm is very similar to 2Δ-Quantum-Color, but includes changes to account for the (potentially) smaller palette size:

Algorithm 3 Quantum-$(1+\epsilon)\Delta-$Color$(G(V,E), \delta, k)$

---

while fewer than $6\epsilon^{-1}Ck\lceil\frac{k}{\delta}\rceil\log^2 n\sqrt{\frac{n^{3+\delta}}{\epsilon\Delta}}$ queries have been made
  for $v \in V$:
    while v is not colored
      Choose a color c uniformly at random from $[2\Delta]$
      if $|\chi(c)| > \frac{2n}{\Delta\epsilon}$
        break choose a new color
      for $j = 1$ to $2\lceil\frac{k}{\delta}\rceil$
        $a \leftarrow$ **Find-Conflict**$(v, \chi(c), \delta)$
        if a is not **false**
          break choose a new color
      Assign c to v and update $\chi(c)$

▶ **Lemma 8.** *Quantum-$(1+\epsilon)\Delta$-Color returns a valid coloring with high probability.*

Proof of lemma 8. Again, to prove this we will show that with high probability every vertex receives a valid color. Note that the only way for a vertex v to receive an invalid color is if **Find-Conflict** incorrectly returns **false** $2\lceil\frac{k}{\delta}\rceil$ times in a row. We know that a single incorrect return of **false** occurs with probability $O(n^{-\delta})$, so the probability that this occurs $2\lceil\frac{k}{\delta}\rceil$ times in a row is $O(n^{-2k})$.

So, if every vertex is colored by the end of the algorithm, we can bound the probability that there is some vertex with an invalid color with a union bound as follows

$$\begin{aligned}\mathbb{P}(\text{Some vertex has an invalid color}) &\leq \sum_{t=1}^{n}\mathbb{P}(v_t \text{ is assigned an invalid color})\\ &= O(n^{1-2k})\end{aligned}$$

Now, we must show that every vertex is assigned a color before too many queries are used. As before, let Q_t be the number of queries used when attempting to color v_t and let K_t be the number of randomly chosen colors which are tested for v_t. Every time we select one of these random colors we call **Find-Conflict** at most $2\lceil\frac{k}{\delta}\rceil$ times, meaning that the algorithm makes at most $2C\lceil\frac{k}{\delta}\rceil\sqrt{\frac{n^{1+\delta}}{\epsilon\Delta}}$ queries per color. This gives,

$$\mathbb{P}[Q_t \geq 6\epsilon^{-1}Ck\lceil\frac{k}{\delta}\rceil\log^2 n\sqrt{\frac{n^{1+\delta}}{\epsilon\Delta}}] \leq \mathbb{P}[K_t \geq 3\epsilon^{-1}k\log n]$$

Recall that for any $v_t \in V$ there are at least $\epsilon\Delta$ valid colors for v_t. Also, note that since $\epsilon \in [\frac{2}{\epsilon\Delta}, 1]$ there are at most $\frac{\epsilon\Delta}{2}$ colors $c \in [(1+\epsilon)\Delta]$ with $\chi(c) > |\frac{2n}{\epsilon\Delta}|$. So, there are at least $\frac{\epsilon\Delta}{2}$ valid colors with sufficiently small color class. This means that a randomly chosen color is valid and small with probability at least $\frac{\epsilon\Delta/2}{(1+\epsilon)\Delta} = \frac{\epsilon}{2+2\epsilon}$ taking $\epsilon < 1$ we have $\frac{\epsilon}{2+2\epsilon} \geq \frac{\epsilon}{4}$. Therefore,

$$\mathbb{P}(C_t \geq 3\epsilon^{-1}k\log n) \leq \left(1 - \frac{\epsilon}{4}\right)^{3\epsilon^{-1}k\log n}$$

Now, note that $\left(1-\frac{\epsilon}{4}\right)^{\epsilon^{-1}} \leq \frac{1}{\sqrt[4]{e}}$ for $\epsilon \in (0,1]$; therefore,

$$\left(1-\frac{\epsilon}{4}\right)^{3\epsilon^{-1}k\log n} \leq \left(\frac{1}{\sqrt[4]{e}}\right)^{3k\log n} \leq 2^{-k\log n} \leq n^{-k}\,.$$

Using this and a union bound we have

$$\begin{aligned}
\mathbb{P}\Big(\sum_{t=1}^{n} Q_t \geq 6\epsilon^{-1}Ck\lceil\frac{k}{\delta}\rceil \log^2 n\sqrt{\frac{n^{3+\delta}}{\epsilon\Delta}}\Big) &\leq \mathbb{P}\Big(\bigcup_{t=1}^{n} Q_t \geq 6\epsilon^{-1}Ck\lceil\frac{k}{\delta}\rceil \log^2 n\sqrt{\frac{n^{1+\delta}}{\epsilon\Delta}}\Big)\\
&\leq \sum_{t=1}^{n}\mathbb{P}\Big(Q_t \geq 6\epsilon^{-1}Ck\lceil\frac{k}{\delta}\rceil \log^2 n\sqrt{\frac{n^{1+\delta}}{\epsilon\Delta}}\Big)\\
&\leq \frac{n}{n^k}\\
&= O(n^{1-k})
\end{aligned}$$

This shows that Quantum-$(1+\epsilon)\Delta$-Color produces a valid coloring with probability at least $1-O(n^{1-2k}-O(n^{1-k})=1-O(n^{1-k})$, concluding the proof of lemma 8 and therefore thoerem 7. ◀

◀

Of course when $\Delta \geq n^{1/3}$ this algorithm makes $\tilde{O}(\epsilon^{-3/2}n^{4/3+\delta/2})$ queries and when $\Delta \leq n^{1/3}$ the greedy algorithm does just as well.

5 Sublinear Maximal Matching

The problem of $O(\Delta)$ coloring is very closely related to the problem of maximal matching. This problem is defined as follows: given a graph $G=(V,E)$ find a matching (set of disjoing edges) which is maximal (not contained in any larger matching of G). There is no hope of finding a classical algorithm for this problem which makes $o(n^2)$ queries due to the following theorem of [2]:

▶ **Theorem 9.** *Any algorithm (possibly randomized) that can output a maximal matching of an input graph with sufficiently large constant probability requires $\Omega(n^2)$ queries to the graph.*

However, in this section we will prove the following:

▶ **Theorem 10.** *For all $\delta > 0$ there exists a algorithm which returns a maximal matching of G with high probability and makes $\tilde{O}(n^{3/2+\delta/2})$ queries.*

The basic strategy of this algorithm will be as follows: start with an empty matching M and examine the vertices of V. If $v \in V$ is already the endpoint of some edge in the matching we move on, otherwise we search for another vertex u not yet in the matching which is adjacent to v. If such a u exists we include (u,v) in the matching and move on. If no such edge exists, then we know that v can not be included in the matching. Classically, Δ queries are required in the worst case to determine if v has a neighbor that is not already used by some edge in the matching, but this sub-problem lends itself to a unstructured search-like interpretation.

M will denote the set of edges used in the matching and $V(M)$ will be the set of all vertices which are endpoints of some edge in M.

Recall that **Find-Conflict** was used in the previous two algorithms to check if for a given vertex v and a set of vertices S with $v \notin S$ there exists some edge of the form (u,v)

with $u \in S$ in order to determine if a particular color would be valid for v or not. In this algorithm our procedure will be slightly different. Now, our general strategy is to find edges which are pairwise disjoint from those which are already included in M. The algorithm is described below:

Algorithm 4 Quantum-Match(G, k, δ)

$M := \{\}$ is our initially empty set edges
for $v \in V$:
  if $v \in V(M)$
    break move to the next vertex
  for $i = 1$ to $\lceil \frac{k}{\delta} \rceil$:
    $a \leftarrow$**Find-Conflict**$(v, V \setminus V(M), \delta)$
    if $a \neq$ **false**
      break $a \in V$
  if $a \neq$ **false**:
    Insert the pair (a, v) into M
return M

▶ **Lemma 11.** *Quantum-Match(G, k) returns a maximal matching with probability at least $O(n^{1-k})$ and makes $\tilde{O}(n^{3/2+\delta/2})$.*

Proof of lemma 11. The only way for this algorithm to return a matching which is not maximal (or empty) is if at some point there exists a vertex $v \in V \setminus V(M)$ that can be included in the partial matching. In other words, v belongs to at least one edge in $E(G)$, say (u, v), such that $u \in V \setminus V(M)$ and the mistake occurs when **Find-Conflict** incorrectly returns **false** $\lceil \frac{k}{\delta} \rceil$ times in a row. For every $v \in V$ this occurs with probability at most $O(n^{-k})$ from claim 3.1, so

$$\mathbb{P}(M \text{ is maximal}) \geq 1 - \sum_{t=1}^{n} \mathbb{P}(\text{Extend-Matching}(M, v_t, k) \text{ fails}) = 1 - O(n^{1-k})\,.$$

Note that in each call of **Find-Conflict**, $S = V \setminus V(M)$ has size at most n, so **Find-Conflict** makes at most $O(\sqrt{n^{1+\delta}} \log n)$ every time it is called in this algorithm Further, **Find-Conflict** is called at most once per vertex, hence the total query complexity of Quantum-Match is $\tilde{O}(n^{3/2+\delta/2})$. This concludes the proof of lemma 11 and proves theorem 10. ◀

6 Lower Bounds

In this section, we establish some lower bounds for coloring and maximal matching with a very simple argument relying on the optimality of Grover's algorithm:

▶ **Theorem 12.** *$\Omega(n)$ queries are necessary to obtain a 2- coloring on graphs with exactly one edge.*

Proof of theorem 12. Finding a 2-coloring for these graphs, given the promise that there is exactly one edge is the same as determining for which $i, j \in [n]$ with $i \neq j$ we have $M[i, j] = 1$. Clearly, we can see this as an instance of unstructured search with exactly one marked element among the $\binom{n}{2} = O(n^2)$ possible edges. Since unstructured search requires $\Omega(\sqrt{N})$ quantum queries, $\Delta + 1$ coloring requires $\Omega(n)$ quantum queries in general. This also implies that $\Omega(n)$ queries are necessary for 2Δ coloring on general graphs since finding a $2-$coloring for graphs with exactly one edge is also equivalent to finding a 2Δ coloring $(\Delta = 1)$. ◀

▶ **Theorem 13.** *Any quantum algorithm which returns a maximal matching with high probability requires* $\Omega(n)$ *queries.*

Proof of theorem 13. Again, take a graph with n vertices and exactly one edge. In this instance there is exactly one non-empty maximal matching that we could return. Hence, doing so is equivalent to finding the one pair (u, v) such that $M[u, v] = 1$ which requires $\Omega(\sqrt{n^2}) = \Omega(n)$ queries to the oracle again due to [4]. ◀

A Details of Find-Conflict

Here, we will give an explicit description of **Find-Conflict** as well as a proof sketch of correctness.

Algorithm 5 **Find-Conflict**(v, S, δ)

Let A be a set of ancilla qubits such that $|A| + |S| = O(n^\delta |S|)$ and let $\gamma = \frac{6}{5}$
$j \leftarrow 1$
while $j \leq |S| + |A|$:
  Prepare the state $|s\rangle = \frac{1}{\sqrt{|A|+|S|}}\left(\sum_{u \in S} |uv\rangle + \sum_{|\alpha\rangle \in A} |\alpha\rangle\right)$
  Let $U_s = 2|s\rangle\langle s| - I$
  for $i = 1$ to $\frac{\pi}{4}\sqrt{\frac{|A|+|S|}{j}}$
    Apply the oracle unitary O_M and U_s to $|s\rangle$
  Measure $|s\rangle$ resulting in λ
  if λ corresponds to a pair (u, v) rather than an element of A
    Make the classical query $M[u, v]$
    if $M[u, v] = 1$
      return u
  $j \leftarrow \gamma j$ // λ must've corresponded to an ancilla qubit or (u, v) with $M[u, v] = 0$
return false

Since we don't know how many (if any) marked items are present in the space that we are searching over we must slowly increment the preseumed number of such marked items j from 1 all the way up to $n^\delta |S|$. Let j^* be the true number of marked items in S. Then at some point $\frac{5j^*}{6} \leq j \leq \frac{6j^*}{5}$ and when this is the case, the Grover search will yield λ corresponding to one of these edges with probability $1 - O((n^\delta |S|)^{-1})$. Hence, for any value of $j^* \geq 1$, such an edge is found with high probability and if no such edge exists then there is no way for the algortihm to return anything other than **false**. Note that the inner Grover search makes $O(\sqrt{n^\delta |S|})$ queries and that the outer while-loop will be iterated $O(\log n^\delta |S|)$ times, so in total, $O(\sqrt{n^\delta |S|} \log n^\delta |S|)$ queries are made, as desired.

References

1 Noga Alon and Sepehr Assadi. Palette Sparsification Beyond (Δ+1) Vertex Coloring. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020)*, volume 176 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 6:1–6:22. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.APPROX/RANDOM.2020.6.

2 Sepehr Assadi, Yu Chen, and Sanjeev Khanna. Sublinear algorithms for (Δ+ 1) vertex coloring. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 767–786. SIAM, 2019. doi:10.1137/1.9781611975482.48.

3 Aleksandrs Belovs and Ansis Rosmanis. On the power of non-adaptive learning graphs. *2013 IEEE Conference on Computational Complexity*, Jun 2013. URL: http://dx.doi.org/10.1109/CCC.2013.14, doi:10.1109/ccc.2013.14.

4 Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, Oct 1997. URL: http://dx.doi.org/10.1137/S0097539796300933, doi:10.1137/s0097539796300933.

5 Aija Berzina, Andrej Dubrovsky, Rusins Freivalds, Lelde Lace, and Oksana Scegulnaja. Quantum query complexity for some graph problems. In Peter Van Emde Boas, Jaroslav Pokorný, Mária Bieliková, and Július Štuller, editors, *SOFSEM 2004: Theory and Practice of Computer Science*, pages 140–150, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

6 Harry Buhrman, Christoph Dürr, Mark Heiligman, Peter Høyer, Frédéric Magniez, Miklos Santha, and Ronald de Wolf. Quantum algorithms for element distinctness. *SIAM Journal on Computing*, 34(6):1324–1330, Jan 2005. URL: http://dx.doi.org/10.1137/S0097539702402780, doi:10.1137/s0097539702402780.

7 Francois Le Gall. Improved quantum algorithm for triangle finding via combinatorial arguments. *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, Oct 2014. URL: http://dx.doi.org/10.1109/FOCS.2014.31, doi:10.1109/focs.2014.31.

8 Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972. doi:10.1007/978-1-4684-2001-2_9.

9 Shelby Kimmel and R. Teal Witter. A query-efficient quantum algorithm for maximum matching on general graphs. 2020. arXiv:2010.02324.

10 Troy Lee, Miklos Santha, and Shengyu Zhang. Quantum algorithms for graph problems with cut queries. arXiv:2007.08285, 2020. https://arxiv.org/abs/2007.08285.

11 Frédéric Magniez, Ashwin Nayak, Jérémie Roland, and Miklos Santha. Search via quantum walk. *SIAM journal on computing*, 40(1):142–164, 2011. doi:10.1137/090745854.

12 Fang Song. Early days following grover's quantum search algorithm. arXiv:1709.01236, 2017. https://arxiv.org/abs/1709.01236.

13 Theodore J. Yoder, Guang Hao Low, and Isaac L. Chuang. Fixed-point quantum search with an optimal number of queries. *Physical Review Letters*, 113(21), Nov 2014. URL: http://dx.doi.org/10.1103/PhysRevLett.113.210501, doi:10.1103/physrevlett.113.210501.