Simple Vertex Coloring Algorithms in the Quantum Query Model

Introduction

In the graph coloring problem we are given a graph $G = (V, E)$ and an integer k as input and are asked to determine if there exists some coloring $C : V \to \{1, 2, \dots k\}$ such that for any $(u, v) \in E$ we have $C(u) \neq C(v)$. If such a C exists then G is said to be *k-colorable*. We denote by $\chi(c)$ the set of all vertices with color c in a particular assignment.

- If Δ denotes the maximum degree of G then it can easily be shown that G admits a $\Delta + 1$ coloring
- There exists a classical greedy algorithm for this which runs in $O(n\Delta)$ time.
- Until recently, this was the best known algorithm for the $\Delta + 1$-coloring problem (and for $O(\Delta)$ coloring).

In this work we focus on the $(1+\epsilon)\Delta$-coloring problem for $\epsilon \in (0, 1]$.

Our Results

In this work we establish the following algorithmic results:

Colors	Quantum	Classical
2Δ-coloring	$\tilde{O}(n^{4/3})$	$O(n\sqrt{n})$
$(1+\epsilon)\Delta$-coloring	$\tilde{O}(\epsilon^{-3/2} n^{4/3})$	$O(\epsilon^{-1} n\sqrt{n})$

- Additionally, we establish an $\Omega(n)$ quantum lower bound for $O(\Delta)$-coloring.

Previous Work

Amazingly, the $\Delta + 1$-coloring problem does admit sub-linear classical algorithms:

- [1] give an $\tilde{O}(n\sqrt{n})$ time algorithm.
- The same authors also provide an $\Omega(n\sqrt{n})$ lower bound not only for coloring with $\Delta + 1$ colors, but for coloring with $O(\Delta)$ colors

The Model

In the classical query model algorithms are allowed to make queries of two types:

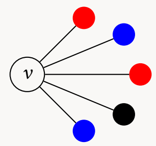
1. Pair queries: Is there an edge between u and v?
2. Neighbor queries: What is the jth neighbor of v?

In the quantum query model we also have access to a quantum oracle, O_G, which acts as follows:

$$O_G|u, v\rangle = \begin{cases} |u, v\rangle & \text{if } (u, v) \notin E \\ -|u, v\rangle & \text{if } (u, v) \in E \end{cases}$$

Greedy Algorithm

There is at least one unused color because a vertex has at most Δ neighbors and there are $\geq \Delta + 1$ available colors. We scan all neighbors to determine which colors are valid.



Randomized Approach

- The greedy algorithm is limited by the need to look at all neighbors - this can be very slow
- In our randomized approach, we instead scan a whole "color class" for a particular color c to see if using that color would result in a conflict.
- So, we are no constrained by the degree of a particular vertex

To color a particular vertex, the algorithm works as follows

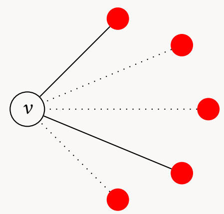
1. Pick a color $c \in [(1+\epsilon)\Delta]$ at random
2. For each u that we have previously colored with c, ask if u and v share an edge.
   - If v is not adjacent to any vertices colored with c, we use c on v and move to the next vertex
   - Otherwise we go back to 1.

Theorem

This algorithm produces a valid $(1+\epsilon)\Delta$ coloring for $\epsilon \geq \frac{2}{\Delta}$ with expected query complexity $O(\epsilon^{-1} n\sqrt{n})$, matching the lower bound of [1] when $\epsilon^{-1} = O(1)$.

Visualizing the Algorithm

If we have a vertex v and a color c we gather the set of all vertices that have color c, $\chi(c)$, and determine if there are any edges among these "potential edges" which would create a conflict.



To check if there are any edges we just make the classical pair queries (u, v) for $u \in \chi(c)$.

Quantizing

- The randomized algorithm is appealing on account of its simplicity, yet optimal performance (for sufficiently large ϵ)
- This algorithmic template can be readily extended to yield a quantum speedup, *breaking* the $\Omega(n\sqrt{n})$ quantum lower bound.

In particular, the only change we need to make is to the subroutine by which a color is checked for validity. For a vertex v and a random color c this amounts to answering the following question:

Are there any edges of the form (u, v) for $u \in \chi(c)$?

- Classically, this validity check takes $\Theta(|\chi(c)|)$ queries
- However, this can be realized as a variant of unstructured search
- Rather than finding the one "marked element" (an edge actually present in the graph), we simply wish to know if there exist *any* marked elements
- Using a modified amplitude amplification algorithm (à la Grover Search) this can be done in $O(\sqrt{|\chi(c)|})$ queries

Quantum Algorithm

The quantum algorithm is nearly the same as the randomized one:

1. Pick a random color c
2. Perform a Grover Search-like procedure looking for edges between v and $|\chi(c)|$ this takes $O(\sqrt{|\chi(c)|})$ queries.
   - If c is a valid color for v, we use it and move to the next vertex
   - Otherwise, repeat 1.

We can view the dotted edges in our previous visualization as components of our quantum state that are unaffected by the oracle and solid edges are.

Theorem

This quantum algorithm returns a valid $(1+\epsilon)\Delta$-coloring with high probability and makes $\tilde{O}(\epsilon^{-3/2} n^{4/3})$ queries.

Remarks

- Note that when $\epsilon = 1$ we get a 2Δ-coloring algorithm using $\tilde{O}(n^{4/3})$ queries, breaking the classical lower bound [1]
- This algorithm leads to super-linear (in m) complexity for the problem of $(\Delta + 1)$-coloring

Analysis

The main challenge in bounding the query complexity is ensuring that the search subroutine does not use too many queries, so we enforce the requirement that $|\chi(c)| \leq \frac{2n}{\Delta}$. This way, we avoid the possibility of using $\Theta(\sqrt{n})$ queries on some large color classes. The analysis of both algorithms is fairly standard, see full paper for details.

Other Applications

Several other problems benefit from the quantum search subroutine including the maximal matching and maximal independent set problems.

Lower Bound

In addition to the algorithmic results we were able to establish a lower bound for $\Delta + 1$-coloring (and therefore $O(\Delta)$-coloring) in the quantum query model via a simple argument. Consider a graph G with n vertices and exactly 1 edge (note that $\Delta = 1$). Coloring this graph is equivalent to an unstructured search among all unordered pairs of vertices and doing so requires $\Omega(\sqrt{n^2}) = \Omega(n)$ queries for any number of colors greater than 1. Hence, $O(\Delta)$-coloring requires $\Omega(n)$ queries, from the optimality of Grover search [2].

Open Questions

As stated, this work does not address the $1 + \Delta$-coloring problem directly, but comes close. In particular, we are left with the following open questions

1. Is there an $o(n\sqrt{n})$ query quantum algorithm for $1 + \Delta$-coloring?
2. Can the quantum lower bound on $O(\Delta)$-coloring be improved past $\Omega(n)$? What about quantum algorithms for $O(\Delta)$-coloring; it would be unlikely $O(n^{4/3})$ is optimal.
3. Could other graph-partitioning problems admit a similar quantum speed-up?

References

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.

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.

Jackson Morris, Fang Song
jrexmo.github.io, fangsong.info
UCLA, Portland State University