Quantum Algorithms for Graph Coloring and Other Partitioning, Covering, and Packing Problems

Serge Gaspers
UNSW Sydney, Australia

Jerry Zirui Li
James Ruse Agricultural High School, Carlingford, Australia
UNSW Sydney, Australia

Abstract

Let U be a universe on n elements, let k be a positive integer, and let $\mathcal{F}$ be a family of (implicitly defined) subsets of U. We consider the problems of partitioning U into k sets from $\mathcal{F}$, covering U with k sets from $\mathcal{F}$, and packing k non-intersecting sets from $\mathcal{F}$ into U. Classically, these problems can be solved via inclusion–exclusion in $2^n n^{O(1)}$ time [8]. Quantumly, there are faster algorithms for graph coloring with running time $O(1.9140^n)$ [26] and for Set Cover with a small number of sets with running time $O\left(1.7274^n |\mathcal{F}|^{O(1)}\right)$ [1]. In this paper, we give a quantum speedup for Set Partition, Set Cover, and Set Packing whenever there is a classical enumeration algorithm that lends itself to a quadratic quantum speedup, which, for any subinstance on a set $X \subseteq U$, enumerates at least one member of a k-partition, k-cover, or k-packing (if one exists) restricted to (or projected onto, in the case of k-cover) the set X in $c^{|X|}n^{O(1)}$ time with $c < 2$. Our bounded-error quantum algorithm runs in time $(2+c)^{n/2}n^{O(1)}$ for Set Partition, Set Cover, and Set Packing. It is obtained by combining three algorithms that have the best running time for some values of c. When $c \leq 1.147899$, our algorithm is slightly faster than $(2+c)^{n/2}n^{O(1)}$; when c approaches 1, it matches the $O\left(1.7274^n |\mathcal{F}|^{O(1)}\right)$ running time of [1] for Set Cover when $|\mathcal{F}|$ is subexponential in n.

For covering, packing, and partitioning into maximal independent sets, maximal cliques, maximal bicliques, maximal cluster graphs, maximal triangle-free graphs, maximal cographs, maximal claw-free graphs, maximal trivially-perfect graphs, maximal threshold graphs, maximal split graphs, maximal line graphs, and maximal induced forests, we obtain bounded-error quantum algorithms with running times ranging from $O(1.8554^n)$ to $O(1.9629^n)$. Packing and covering by maximal induced matchings can be done quantumly in $O(1.8934^n)$ time.

For Graph Coloring (covering with k maximal independent sets), we further improve the running time to $O(1.7956^n)$ by leveraging faster algorithms for coloring with a small number of colors to better balance our divide-and-conquer steps. For Domatic Number (packing k minimal dominating sets), we obtain a $O((2-\varepsilon)^n)$ running time for some $\varepsilon > 0$.

Several of our results should be of interest to proponents of classical computing:

- We present an inclusion-exclusion algorithm with running time $O^*\left(\sum_{i=0}^{\lfloor \alpha n \rfloor} \binom{n}{i}\right)$, which determines, for each $X \subseteq U$ of size at most αn, $0 \leq \alpha \leq 1$, whether $(X, \mathcal{F})$ has a k-cover, k-partition, or k-packing. This running time is best-possible, up to polynomial factors.
- We prove that for any linear-sized vertex subset $X \subseteq V$ of a graph $G = (V, E)$, the number of minimal dominating sets of G that are subsets of X is $O((2-\varepsilon)^{|X|})$ for some $\varepsilon > 0$.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Theory of computation → Quantum complexity theory

Keywords and phrases Graph algorithms, quantum algorithms, graph coloring, domatic number, set cover, set partition, set packing

Digital Object Identifier 10.4230/LIPIcs.ICALP.2024.69

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://doi.org/10.48550/arXiv.2311.08042 [20]

51st International Colloquium on Automata, Languages, and Programming (ICALP 2024).
Editors: Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson;
Article No. 69; pp. 69:1–69:20

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Graph Coloring is an example of a problem requiring to partition an n-element set U into k sets from a family $\mathcal{F}$. In this case U is the vertex set of a graph G and $\mathcal{F}$ is implicitly defined as the independent sets of G. We can also view Graph Coloring as a covering problem where the vertex set needs to be covered with k maximal independent sets.

In 2006, Björklund and Husfeldt [4] and Koivisto [25] independently solved Graph Coloring in $O^*(2^n)$ time via a new inclusion–exclusion approach, along with other partitioning and covering problems. The approach has been used for packing problems as well, and has been generalized to solve more generic subset convolution problems [5, 7, 8].

In this work, we give faster quantum algorithms for a range of partitioning, covering, and packing problems, including Graph Coloring and Domatic Number. To do this, we use the framework of Ambainis et al. [1] where a preprocessing step computes solutions to small subinstances and stores them in QRAM. These solutions are then accessed by a divide-and-conquer algorithm which enjoys a quadratic speedup in quantum models of computation via techniques such as Grover's search [22]. For the preprocessing step (Section 3), we adapt the afore-mentioned inclusion-exclusion approach [8] to compute the solutions to all subinstances induced by a small subset of U up to roughly $n/4$ elements. For the divide-and-conquer step (Section 4) one would ideally like to divide U into two halves; unfortunately, optimal partitions[1] may not allow for such a balanced split. However, we can restrict the divide-and-conquer step to divide U into two parts where in the larger part (equivalently, in both parts) the removal of one set of an optimal partition results in at most $n/2$ elements. Finding one set of the optimal partition is done via an algorithm that enumerates all relevant candidate sets; for Graph Coloring, it enumerates the maximal independent sets in the graph induced by the subset $X \subseteq U$ under consideration. Importantly, we need that this enumeration can be done in $O^*\left(c^{|X|}\right)$ time, for some $c < 2$ by a classical algorithm that has a quadratic quantum speedup, so that after two levels of divide-and-conquer, the overall quadratic quantum speedup outperforms the classical $O^*(2^n)$ running time.

Our algorithm differs from the previously fastest quantum algorithm for Graph Coloring by Shimizu and Mori [26] in both the preprocessing step and the divide-and-conquer step. Our preprocessing step is deterministic and its running time is optimal, matching the size of the output up to polynomial factors; the preprocessing step of [26] is a bounded-error quantum algorithm whose running time is a multiplicative factor of $3^{|X|/6}$ slower than ours for each small (up to size roughly $n/4$) subset $X \subseteq U$. For the divide-and-conquer step, our divide-then-enumerate strategy is described above; [26] employ an enumerate-then-divide strategy, where the enumeration is done on the set to be divided and the remainder is then divided into two sets of size at most half the original set. It turns out that blending the the divide-then-enumerate and the enumerate-then-divide strategy gives faster algorithms when $c \leq 1.147899$ (Section 5). For $c \leq 1.0872$ case, we also use a third level of divide-and-conquer, and when c approaches 1, our $O(1.7274^n)$ running time matches the running time for Set Cover with a subexponential number of sets of Ambainis et al. [1].

This gives improved algorithms for a range of partitioning, covering, and packing problems (Section 6). We further improve the running time for Graph Coloring (Section 7) to $O(1.7956^n)$ by leveraging faster algorithms for a small number of colors [26]. Our algorithm considers large subsets of vertices ($\geq 0.48n$) and checks whether they are 5-colorable, 6-colorable with

[1] For conciseness, we will mainly discuss partitions in the introduction. The treatment of covers and packings is similar.

no large 5-colorable subset, and in some cases 7-colorable via a new 7-Coloring algorithm that relies on the preprocessing step. The advantage of excluding such cases from further consideration is that we can make the divide-and-conquer steps more balanced.

For Domatic Number, at first glance it seems that our approach cannot be used. The issue is that when considering a vertex subset X, even though there is an algorithm that enumerates all minimal dominating sets of $G[X]$ in $O(1.7159^n)$ time [16], this is insufficient for our purposes: we need to enumerate minimal vertex subsets of X that dominate all of G, not just $G[X]$, in $O^*\left(c^{|X|}\right)$ time for some $c < 2$. In Section 8, we show that such an enumeration algorithm (with a quadratic quantum speedup) indeed exists provided that $|X| = \Omega(n)$. This then also gives a bounded-error quantum algorithm for Domatic Number running in $O((2-\varepsilon)^n)$ time.

2 Preliminaries

For a proposition P, the *Iverson bracket* $[P]$ is a function that returns 1 if P is true and 0 otherwise.

Asymptotic notation

The O^*-notation is similar to the usual O-notation but allows to hide polynomial factors in the input size. The $\tilde{O}$-notation hides polylogarithmic factors. We make heavy use of Stirling's approximation for factorials, which implies that $\binom{n}{k} = O^*\left(\left(\frac{n}{k}\right)^k \cdot \left(\frac{n}{n-k}\right)^{n-k}\right)$, and of the binomial theorem, $\sum_{k=0}^{n}\binom{n}{k}x^k y^{n-k} = (x+y)^n$.

Set Systems

An *implicit set system* [15] is a function Φ that takes as input a string $I \in \{0,1\}^*$ and outputs a set system $(U_I, \mathcal{F}_I)$, where U_I is a universe and $\mathcal{F}_I$ is a collection of subsets of U_I. The string I is referred to as an *instance* and we denote by $|U_I| = n$ the size of the universe and by $|I| = N$ the size of the instance. We assume that $N \geq n$. The implicit set system Φ is said to be *polynomial time computable* if (a) there exists a polynomial time algorithm that given I, produces U_I, and (b) there exists a polynomial time algorithm that given I, U_I and a subset S of U_I, determines whether $S \in \mathcal{F}_I$. Throughout this paper, we consider only polynomial time computable implicit set systems. We define a *subset polynomial time computable* implicit set system Φ to be a polynomial time computable set system, where (c) there exists a polynomial time algorithm that given I, U_I and a subset S of U_I, determines whether $S \subseteq S'$ for some $S' \in \mathcal{F}_I$. This is equivalent to determining whether $S \in \mathcal{F}\downarrow$, where the downward closure $\mathcal{F}\downarrow$ of $\mathcal{F}$ contains all sets in $\mathcal{F}$ and their subsets.

For any subset of elements $X \subseteq U$, an ordered tuple $(S_1, \ldots, S_k)$ of k sets from $\mathcal{F}$ is a *k-cover* for X if the union of these sets is X; it is a *k-packing* for X if the S_i's are contained in X and are pairwise non-intersecting; it is a *k-partition* for X if it is both a k-cover and a k-packing for X.

For a subset polynomial time computable implicit set system Φ, the input of the Φ-Set Cover problem is an instance I and an integer k, and the question is whether the set system $\Phi(I) = (U_I, \mathcal{F}_I)$ has a k-cover. This is equivalent to asking whether $(U_I, \mathcal{F}_I\downarrow)$ has a k-cover and therefore, we assume that $\mathcal{F}_I = \mathcal{F}_I\downarrow$ whenever discussing k-covers. For a polynomial time computable implicit set system Φ, the input of the Φ-Set Partition and Φ-Set Packing problem is an instance I and an integer k, and the question is whether the set system $\Phi(I) = (U_I, \mathcal{F}_I)$ has a k-partition or k-packing, respectively. We generally omit Φ and the subscript I when they are clear from context.

Graphs

In a graph $G = (V, E)$, the *open neighborhood* of a vertex v, denoted $N_G(v)$ is the set containing all vertices adjacent to v in G. The *closed neighborhood* of v in G also contains v itself, and is denoted $N_G[v] = \{v\} \cup N_G(v)$. Again, we may omit the subscript G. For a vertex subset $X \subseteq V$, the graph $G - X$ is obtained from G by removing the vertices in X and all their incident edges; the graph $G[X]$ induced on X is the graph $G - (V \setminus X)$.

Quantum Algorithms

It is known that most classical branching algorithms have a quadratic speedup on quantum machines. As [26], we also rely on the following results.

▶ **Theorem 1** ([22, 9]). *Let $A : \{1, 2, \ldots, N\} \to \{0, 1\}$ be a bounded-error quantum algorithm with running time T. Then, there is a bounded-error quantum algorithm computing $\bigvee_{x\in\{1,\ldots,N\}} A(x)$ with running time $\tilde{O}(\sqrt{N}T)$.*

▶ **Theorem 2** ([13]). *Let $A : \{1, 2, \ldots, N\} \to \{0, 1\}$ be a bounded-error quantum algorithm with running time T. Then, there is a bounded-error quantum algorithm computing $\min_{x\in\{1,\ldots,N\}} A(x)$ with running time $\tilde{O}(\sqrt{N}T)$.*

In our context, A is an algorithm exploring paths in superposition from the root to the leaves of the search tree of a classical branching algorithm. The amplitudes of this exploration depend on estimates of the sizes of the subtrees, either by relying on an analysis of the classical branching algorithm [18, 26] or by on-the-fly estimations [2]. We speak of a *simple* branching algorithm when the exploration of one root-to-leaf path is independent of the other paths; this excludes, for example, algorithms relying on clause learning, re-use of computation done in earlier branches, and branch-and-bound. For a simple branching algorithm with running time $O^*(c^n)$, one obtains a bounded-error quantum algorithm with running time $O^*(c^{n/2})$ in this way; we simply say that we apply Grover's search to the branching algorithm.

3 Preprocessing small subsets

For $\alpha \in [0, 1]$, a subset X of U is *α-small* if $|X| \leq \alpha n$. Denote by $s(n, \alpha) = \sum_{i=0}^{\lfloor \alpha n \rfloor} \binom{n}{i}$ the number of α-small subsets of U. In this section, we consider the problem of counting the number of k-covers, k-packings, and k-partitions for each α-small subset of U. When considering k-covers, we assume that $\mathcal{F}$ has been replaced by $\mathcal{F}\downarrow$. This is because when we would like to cover a subset of elements of X, we may use a set from $\mathcal{F}$ that also contains elements outside of X. Since Φ is subset polynomial time computable in the Φ-Set Cover problem, we may as well replace $\mathcal{F}$ by $\mathcal{F}\downarrow$; this makes the discussion of covering, partitioning, and packing problems more uniform. Our algorithms run in $O^*(s(n, \alpha))$ time, which is best possible, since the output is a list of $s(n, \alpha)$ integers.

This section heavily relies on previous $O^*(2^n)$ inclusion-exclusion approaches [7, 8] to compute the number of k-covers, k-packings, and k-partitions for U and these results are well-known when $\alpha = 1$. The work by [6] is also related, but their running times depends on the number of supersets of $\mathcal{F}$.

We start by defining the α-small zeta transform, which is central to this section.

▶ **Definition 3.** *Let f be a function from subsets of the universe U to an algebraic ring R. The* α-small zeta transform *of f, denoted $f\zeta_\alpha$ is*

$$f\zeta_\alpha(X) = \sum_{Y\subseteq X} f(Y)$$

for any α-small $X \subseteq U$.

The 1-small zeta transform is also called the *zeta transform* and the α-small zeta transform is precisely the restriction of the zeta transform to α-small sets. Throughout this paper we assume that arithmetic operations in the ring R take $O^*(1)$ time and each ring element is represented using $O^*(1)$ space.

▶ **Definition 4.** *Let f be a function from subsets of the universe U to an algebraic ring R. The* α-small Möbius transform *of f, denoted $f\mu_\alpha$ is*

$$f\mu_\alpha(X) = \sum_{Y \subseteq X} (-1)^{|X \setminus Y|} f(Y)$$

for any α-small $X \subseteq U$.

It is well-known (see, e.g., [17]) that $f\zeta_\alpha\mu_\alpha = f\mu_\alpha\zeta_\alpha = f$ when $\alpha = 1$, and the same is true when $\alpha \neq 1$.

▶ **Lemma 5.** *The α-small zeta transform $f\zeta_\alpha$ and the α-small Möbius transform $f\mu_\alpha$ can be computed in $O^*(s(n, \alpha))$ time.*

Proof. We start with $f\zeta_\alpha$ and proceed as in Yates's method [27]. Consider an arbitrary ordering of the elements of $U = \{v_1, \ldots, v_n\}$. The algorithm considers each α-small $X \subseteq U$ by increasing order of cardinality.

Set $g_0(X) = f(X)$. Then, iterate over the elements of U in the ordering fixed above. When processing element v_i, set

$$g_i(X) = g_{i-1}(X) + [v_i \in X] \cdot g_{i-1}(X \setminus \{v_i\}).$$

Finally, set $f\zeta_\alpha(X) = g_n(X)$.

Correctness can be shown by induction on i by observing that

$$g_i(X) = \sum_{\{v_{i+1}, \ldots, v_n\} \cap X \subseteq Y \subseteq X} f(Y).$$

For each set X the computation takes $O^*(1)$ time, and the number of sets X to be considered is $s(n, \alpha)$.

To compute $f\mu_\alpha$, we use the fact that $\mu_\alpha = \sigma_\alpha\zeta_\alpha\sigma_\alpha$, where the α-small odd-negation transform is

$$f\sigma_\alpha(X) = (-1)^{|X|} f(X),$$

defined for any α-small $X \subseteq U$. Indeed,

$$\begin{aligned} f\sigma_\alpha\zeta_\alpha\sigma_\alpha(X) &= (-1)^{|X|} \cdot \sum_{Y \subseteq X} (-1)^{|Y|} \cdot f(Y) \\ &= \sum_{Y \subseteq X} (-1)^{|X|+|Y|} f(Y) \\ &= \sum_{Y \subseteq X} (-1)^{|X \setminus Y|} f(Y) \end{aligned}$$

since $|X|+|Y|$ and $|X|-|Y| = |X \setminus Y|$ have the same parity. The result now follows, because, for a function g and a set X, $g\sigma_\alpha(X)$ can be computed in $O^*(1)$ time. ◀

We refer to these algorithms as the *fast α-small zeta transform* and the *fast α-small Möbius transform*.

By inclusion-exclusion, the number of k-covers for a subset $X \subseteq U$ is [7]

$$c_k(\mathcal{F}, X) = \sum_{Y \subseteq X} (-1)^{|X \setminus Y|} a(Y)^k, \tag{1}$$

where $a(Y)$ is the number of subsets $Z \subseteq Y$ that belong to $\mathcal{F} = \mathcal{F}\downarrow$.

For the number of k-partitions, we use an indeterminate z in the ring R that allows us to keep track of the sum of the cardinalities of the sets in the cover. The number of k-partitions for a subset $X \subseteq U$ is given [7] by the coefficient of the monomial $z^{|X|}$ in the polynomial

$$d_k(\mathcal{F}, X) = \sum_{Y \subseteq X} (-1)^{|X \setminus Y|} \left(\sum_{j=0}^{|Y|} a_j(Y) z^j \right)^k, \tag{2}$$

where $a_j(Y)$ is the number of size-j subsets $Z \subseteq Y$ that belong to $\mathcal{F}$.

For the number of k-packings of X, we compute the number of $(k+1)$-partitions where the first k members of the $(k+1)$-tuple belong to $\mathcal{F}$ and the last member is an arbitrary subset of X. Noting that $(1+z)^{|Y|} = \sum_{i=0}^{|Y|} \binom{|Y|}{i} z^i$, the number of k-packings for $X \subseteq U$ is [7] the coefficient of $z^{|X|}$ in

$$p_k(\mathcal{F}, X) = \sum_{Y \subseteq X} (-1)^{|X \setminus Y|} (1+z)^{|Y|} \left(\sum_{j=0}^{|Y|} a_j(Y) z^j \right)^k. \tag{3}$$

The first algorithmic step is to compute the values for $a(Y)$ in (1) and the polynomials $\sum_{j=0}^{|Y|} a_j(Y) z^j$ in (2) and (3). Observe that $|Y| \leq |X| \leq \alpha n$. To compute the values $a(Y)$ for all α-small $Y \subseteq U$, observe that $a(Y) = \sum_{Z \subseteq Y} [Z \in \mathcal{F}]$ is the α-small zeta transform of the indicator function of $\mathcal{F}$. Since the implicit set system is polynomial time computable, the indicator function can be evaluated in polynomial time. Therefore, the fast α-small zeta transform allows us to compute all relevant values of $a(\cdot)$ in $O^*(s(n,\alpha))$ time. Similarly, the polynomial $\sum_{j=0}^{|Y|} a_j(Y) z^j$ equals $h\zeta_\alpha(Y)$ where $h(Z) = [Z \in \mathcal{F}] \cdot z^{|Z|}$ and can be computed in the same time bound by the fast α-small zeta transform.

The second algorithmic step is to to use the fast α-small Möbius transform and apply it to the functions that associate with each α-small $Y \subseteq U$ the values $a(Y)^k$; $\left(\sum_{j=0}^{|Y|} a_j(Y) z^j\right)^k$; and $(1+z)^{|Y|} \left(\sum_{j=0}^{|Y|} a_j(Y) z^j\right)^k$, respectively.

We conclude that $c_k(\mathcal{F}, X)$, $d_k(\mathcal{F}, X)$, and $p_k(\mathcal{F}, X)$ for each α-small $X \subseteq U$ can be computed in $O^*(s(n,\alpha))$ time.

▶ **Theorem 6.** *Given a polynomial time computable implicit set family $\Phi(I) = (U, \mathcal{F})$ with $|U| = n$, there is a $O^*(s(n,\alpha))$ time algorithm which determines, for all $k \leq \alpha n$ and all α-small $X \subseteq U$, whether $(X, \mathcal{F})$ has a k-cover (if we assume that $\mathcal{F}$ is closed under subsets), k-partition, or k-packing.*

4 Divide-and-conquer algorithm

Let $\Phi(I) = (U, \mathcal{F})$ be a polynomial time computable implicit set family for which we would like to determine whether there is a k-cover (assuming $\mathcal{F} = \mathcal{F}\downarrow$), k-partition, or k-packing.

We say that a simple branching algorithm *enumerates* a family $\mathcal{F}'$ of subsets of U if, at each leaf of its search tree it finds at most one member of $\mathcal{F}'$, and collectively the leaves find all members of $\mathcal{F}'$ (duplicates are allowed).

Our algorithm uses a divide-and-conquer strategy where the universe is twice partitioned into two.

▶ **Definition 7.** *For a family of subsets $\mathcal{F}$ of a universe U, and a subset $X \subseteq U$, the* restriction *of $\mathcal{F}$ to X is $r(\mathcal{F}, X) = \{S \subseteq X : S \in \mathcal{F}\}$.*

Observe that $(U, \mathcal{F})$ has a k-partition (resp., a k-packing or a k-cover) for $k \geq 2$ iff there is a set $L \subseteq U$ and a positive integer $k_l < k$ such that $(L, r(\mathcal{F}, L))$ has a k_l-partition (resp., a k_l-packing or a k_l-cover) and $(R, r(\mathcal{F}, R))$ has a k_r-partition (resp., a k_r-packing or a k_r-cover), where $R = U \setminus L$ and $k_r = k - k_l$. We say that a k-partition, k-packing, or k-cover $(S_1, \ldots, S_k)$ does not *straddle* X if for each $i \in \{1, \ldots, k\}$, either $S_i \subseteq X$ or $S_i \cap X = \emptyset$.

In this section we prove the following theorem.

▶ **Theorem 8.** *Suppose there is a simple (classical) branching algorithm A, which, given an instance I with $\Phi(I) = (U, \mathcal{F})$ and a subset $X \subseteq U$, enumerates a family $e(X, \mathcal{F}_X)$ of subsets of X from $\mathcal{F}_X$ such that*

- *if $(U, \mathcal{F})$ has a k-cover (resp., a k-partition or a k-packing) that does not straddle X, then $(U, \mathcal{F})$ has a k-cover (resp., a k-partition or a k-packing) $(S_1, \ldots, S_k)$ with $S_1 \in e(X, \mathcal{F}_X)$, and*
- *the algorithm runs in $O^*(c^{|X|})$ time for some $c \leq 2$.*

Then, there is a bounded-error quantum algorithm, which determines whether $(U, \mathcal{F})$ has a k-cover (resp., a k-partition or a k-packing) in $O^*\left(\binom{n}{n/4} + (2+c)^{n/2}\right)$ time, where $n = |U|$.

From now on, we focus on Set Cover; the discussion of Set Partition and Set Packing is analogous. The first step is to use the algorithm from Theorem 6 with $\alpha = \frac{1}{4}$ and store the result in QRAM.

Ideally, we would want to divide the universe U into equal sized sets L and R and compute a k-cover where each set of the cover is responsible for covering elements of either L or R. However, we cannot guarantee that such a k-cover exists. Instead, we can focus on partitions of U into L and R with $|L| \geq n/2$ where the removal of one member of the k-cover decreases the size of L to at most $n/2$.

▶ **Lemma 9.** *For any $x \in \{0, \ldots, n\}$, $(U, \mathcal{F})$ has a k-cover, $k \geq 2$, iff there is a partition of U into (L, R) with $|L| \geq x$ and integers $k_L, k_R \geq 1$ with $k = k_L + k_R$ such that $(L, r(\mathcal{F}, L))$ has a k_L-cover $(S_1, \ldots, S_{k_L})$ and $(R, r(\mathcal{F}, R))$ has a k_r-cover such that $|S_i| \geq |L| - x$ for every $i \in \{1, \ldots, k_L\}$.*

Proof. For the backward direction, assume that $(L, r(\mathcal{F}, L))$ has a k_l-cover and $(R, r(\mathcal{F}, R))$ has a k_r-cover. Then $(L \cup R, r(\mathcal{F}, L) \cup r(\mathcal{F}, R))$ has a k-cover where $k = k_l + k_r$. This k-cover is of the form $(S_1, S_2, \cdots, S_k)$, where $S_i \subseteq r(\mathcal{F}, L) \cup r(\mathcal{F}, R)$. To turn it into a k-cover for $(U, \mathcal{F})$, we replace each S_i by a set S'_i with $S_i \subseteq S'_i \in \mathcal{F}$, and we note that such a set S'_i exists in $\mathcal{F}$, since S_i is the restriction of some set in $\mathcal{F}$ to either L or R. Hence, $(U, \mathcal{F})$ has a k-cover.

For the forward direction, assume that $(U, \mathcal{F})$ has a k-cover $(S_1, S_2, \cdots, S_k)$ and assume, w.l.o.g., that $|S_1| \geq |S_2| \geq \cdots \geq |S_k|$. Let $L = \bigcup_{i=1}^{k_L} S_i$ and $R = U \setminus L$ where k_L is the smallest value such that $|L| > x$. Obviously, $(S_1, S_2, \cdots, S_{k_L})$ is a k_L-cover of $(L, r(\mathcal{F}, L))$ and $(S_{k+1} \cap R, S_{k+2} \cap R, \cdots, S_k \cap R)$ is a k_R-cover of $(R, r(\mathcal{F}, R))$ where $k_R = k - k_L$.

Focusing on S_{k_L}, the smallest set in the cover of L, we have that

$$\begin{aligned}
&\bigcup_{i=1}^{k_L} S_i \setminus S_{k_L} \subseteq \bigcup_{i=1}^{k_L-1} S_i \\
\implies &|\bigcup_{i=1}^{k_L} S_i| - |S_{k_L}| \leq |\bigcup_{i=1}^{k_L-1} S_i| \\
\implies &\quad |L| - |S_{k_L}| \leq x \\
\implies &\quad |S_i| \geq |S_{k_L}| \geq |L| - x \quad \text{for all } i \in \{1, \ldots, k_L\}.
\end{aligned}$$

Therefore, $(U, \mathcal{F})$ has a k-cover if and only if the conditions are satisfied. ◀

In particular, this means that removing any member S_i of the k_L-cover gives a $(k_L - 1)$-cover of $L \setminus S_i$ and $|L \setminus S_i| \leq x$.

We use Grover's search to divide the elements into two sets (L, R) where $|L| \geq n/2$ and k into $k_L + k_R$. Then, we solve each of these two instances independently, again using Grover's search on $(L, r(\mathcal{F}, L))$ (resp., on $(R, r(\mathcal{F}, R))$), dividing it into (LL, LR) where $|LL| \geq n/4$ and $k_{LL} + k_{LR} = k_L$ (similar for R). For $(LL, r(\mathcal{F}, LL))$ (and, similarly, on the corresponding instances for LR, RL, RR), we use Grover's search on the branching algorithm A to enumerate candidate sets $S_1 \in e(LL, r(\mathcal{F}, LL))$; if $|LL| - |S_1| > n/4$, then this branch is unsuccessful; otherwise, look up whether $LL \setminus S_1$ has a $(k_{LL} - 1)$-cover in QRAM.

The running time of this algorithm is

$$O^*\left(\binom{n}{n/4}\right)$$

for running the algorithm from Theorem 6; the steps using Grover's search take

$$O^*\left(\sqrt{\sum_{l=\lceil n/2 \rceil}^{n} \binom{n}{l} \sum_{l'=\lceil n/4 \rceil}^{l} \binom{l}{l'} c^{l'}}\right) = O^*\left((2+c)^{n/2}\right)$$

time to achieve constant success probability. This proves Theorem 8.

Discussion

When $c \geq \frac{16}{3^{3/2}} - 2 \approx 1.079201$, then the running time is $O^*\left((2+c)^{n/2}\right)$. When $c \leq \frac{16}{3^{3/2}} - 2$, then the running time is dominated by the term $\binom{n}{n/4} \approx 1.7548^n$. For $c \leq 1.147899$, the next section gives faster algorithms.

5 Divide-and-conquer algorithms for small c

In this section, we again assume that there is an enumeration algorithm A, as in Theorem 8, with running time $O^*\left(c^{|X|}\right)$. We present two divide-and-conquer algorithms which are faster for small values of c.

Throughout this section, we focus on Set Cover; the discussion of Set Partition and Set Packing is analogous.

▶ **Theorem 10.** *There is a bounded-error quantum algorithm, which determines whether $(U, \mathcal{F})$ has a k-cover (resp., a k-partition or a k-packing) in $O^*\left(\binom{n}{n/4} + (1+c)^{\frac{3}{4} \cdot n}\right)$ time, where $n = |U|$.*

The first step is to use the algorithm from Theorem 6 with $\alpha = \frac{1}{4}$ and store the result in QRAM. Our algorithm is similar to the algorithm in Theorem 8, but we use the branching algorithm A to remove a subset from L before dividing it into (LL, LR). Again, Lemma 9 is central to our algorithm.

We use Grover's Search to divide the elements into two sets (L, R) where $|L| \geq n/2$ and k into $k_L + k_R$. Then, we solve each of these two instances independently, again using Grover's search on $(R, r(\mathcal{F}, R))$, dividing it into (RL, RR) where $|RL| \geq n/4$ and $k_{RL} + k_{RR} = k_R$. For the instance $(L, r(\mathcal{F}, L))$, we use Grover's search on the branching algorithm A to enumerate candidate sets $S_1 \in e(L, r(\mathcal{F}, L))$; if $|L| - |S_1| > n/2$ then this branch is unsuccessful; otherwise, we use Grover's search on the subinstance $(L \setminus S_1, r(\mathcal{F}, L \setminus S_1))$, dividing it into (LL, LR) where $|LL| \geq n/4$ and $k_{LL} + k_{LR} = k_L - 1$.

We process $(LL, r(\mathcal{F}, LL))$ (and, similarly, the corresponding instances for LR, RL, RR) as we did in Theorem 8 – we use Grover's search on the branching algorithm A to enumerate candidate sets $S_1 \in e(LL, r(\mathcal{F}, LL))$; if $|LL| - |S_1| > n/4$, then this branch is unsuccessful; otherwise, look up whether $LL \setminus S_1$ has a $(k_{LL} - 1)$-cover in QRAM.

The running time of this algorithm is

$$O^*\left(\binom{n}{n/4}\right)$$

for the running the algorithm from Theorem 6; the steps using Grover's search take

$$O^*\left(\sqrt{\sum_{l=\lceil n/2\rceil}^{n}\binom{n}{l}c^l\sum_{l'=\lceil n/4\rceil}^{\lfloor n/2\rfloor}\binom{n/2}{l'}c^{l'}}\right) = O^*\left((1+c)^{\frac{3}{4}\cdot n}\right)$$

time to achieve constant success probability. This proves Theorem 10.

Discussion

When $c \geq \frac{2^{8/3}}{3} - 1 \approx 1.11653$, then the running time is $O^*\left((1+c)^{\frac{3}{4}\cdot n}\right)$. When $c \leq \frac{2^{8/3}}{3} - 1$, then the running time is dominated by the term $\binom{n}{n/4} \approx 1.7548^n$. When $c \leq 1.0872$, then the following algorithm is faster.

▶ **Theorem 11.** *If $c \leq 1.0872$, there is a bounded-error quantum algorithm, which determines whether $(U, \mathcal{F})$ has a k-cover (resp., a k-partition or a k-packing) in*

$$O^*\left(\left(\min_{0.1303\leq\alpha\leq 0.25}\left((1+c)^{3/4}c^{\alpha/2}\left(1-4\cdot\alpha\right)^{\frac{4\cdot\alpha-1}{8}}\left(4\cdot\alpha\right)^{-\frac{\alpha}{2}},\ \alpha^{-\alpha}\cdot(1-\alpha)^{\alpha-1}\right)\right)^n\right)$$

time.

The first step is to use the algorithm from Theorem 6 with some $0.1303 \leq \alpha < 0.25$ and store the result in QRAM. Our algorithm is identical to the one presented in Theorem 10 except that we cannot directly look up whether a subset $X \subseteq U$ has a k-cover in QRAM for $|X| \leq n/4$. Instead, we use the following approach.

▶ **Lemma 12.** *If $c \leq 1.0872$ and $0.1303 \leq \alpha < 0.25$, after running the algorithm from Theorem 6, there exists a bounded-error quantum algorithm that checks for a k-cover of a subset $X \subseteq U$ where $|X| \leq n/4$ in $O^*\left(\sqrt{\binom{n/4}{\alpha\cdot n}\cdot c^{\alpha\cdot n}}\right)$ time.*

Proof. We use Grover's search to divide the elements of X into two sets (XL, XR) where $|XL| \geq \alpha \cdot n$ and k into $k_{XL} + k_{XR}$. We use Grover's search on the branching algorithm A to enumerate candidate sets $S_1 \in e(XL, r(\mathcal{F}, XL))$; if $|XL| - |S_1| > \alpha \cdot n$, then this branch is unsuccessful; otherwise, look up whether $XL \setminus S_1$ has a $(k_{XL} - 1)$-cover and XR has a k_{XR}-cover in QRAM. The correctness of this approach follows from Lemma 9.

As the function $f(x) = \binom{n/4}{x} c^x$ is strictly decreasing for $x > \frac{c}{c+1} \cdot \frac{n}{4}$ and

$$\alpha \cdot n \geq 0.1303 \cdot n > \frac{1.0872 \,/\, 4}{1.0872 + 1} \cdot n \geq \frac{c}{c+1} \cdot \frac{n}{4},$$

the running time of this algorithm is

$$O^*\left(\sqrt{\sum_{i=\lceil \alpha \cdot n \rceil}^{\lfloor n/4 \rfloor} \binom{n/4}{i} c^i}\right) = O^*\left(\sqrt{\binom{n/4}{\alpha \cdot n} c^{\alpha \cdot n}}\right).$$

This proves the lemma. ◀

Our algorithm is the same as the one in Theorem 8, except when we check for a $(k_{LL} - 1)$-cover for $LL \setminus S_1$ (resp., on LR, RL, RR) we use Lemma 12 instead of a direct lookup in QRAM.

Running the algorithm from Theorem 6 takes

$$O^*\left(\binom{n}{\alpha \cdot n}\right) = O^*\left(\alpha^{-\alpha n} \cdot (1-\alpha)^{(\alpha-1)n}\right)$$

time. The steps using Grover's search take

$$O^*\left(\sqrt{\sum_{l=\lceil n/2 \rceil}^{n} \binom{n}{l} c^l \sum_{l'=\lceil n/4 \rceil}^{\lfloor n/2 \rfloor} \binom{n/2}{l'} c^{l'}} \sqrt{\binom{0.25 \cdot n}{\alpha \cdot n} c^{\alpha \cdot n}}\right)$$
$$= O^*\left(\left((1+c)^{3/4} c^{\alpha/2} \left(\frac{1}{1-4\cdot\alpha}\right)^{\frac{1-4\cdot\alpha}{8}} \left(\frac{1}{4\cdot\alpha}\right)^{\frac{\alpha}{2}}\right)^n\right) \text{ time.}$$

The running time of the first part increases with α while the running time of the second part decreases with α. We can therefore optimise the running time by balancing the value of α for a given c. To determine when this approach is faster than $O^*\left(\binom{n}{n/4}\right)$ we compute the value of c when α is balanced at 0.25:

$$(1+c)^{3/4} c^{1/8} = \frac{4}{3^{3/4}}$$
$$\implies \qquad c \approx 1.08724 \ .$$

Therefore, this algorithm outperforms the algorithms of Theorem 8 and Theorem 10 when $c \leq 1.08723$.

Combining Theorem 8, Theorem 10, and Theorem 11, we obtain the following corollary.

▶ **Corollary 13.** *There is a bounded-error quantum algorithm, which determines whether $(U, \mathcal{F})$ has a k-cover (resp., a k-partition or a k-packing) in $O^*\left((2+c)^{n/2}\right)$ time, where $n = |U|$.*

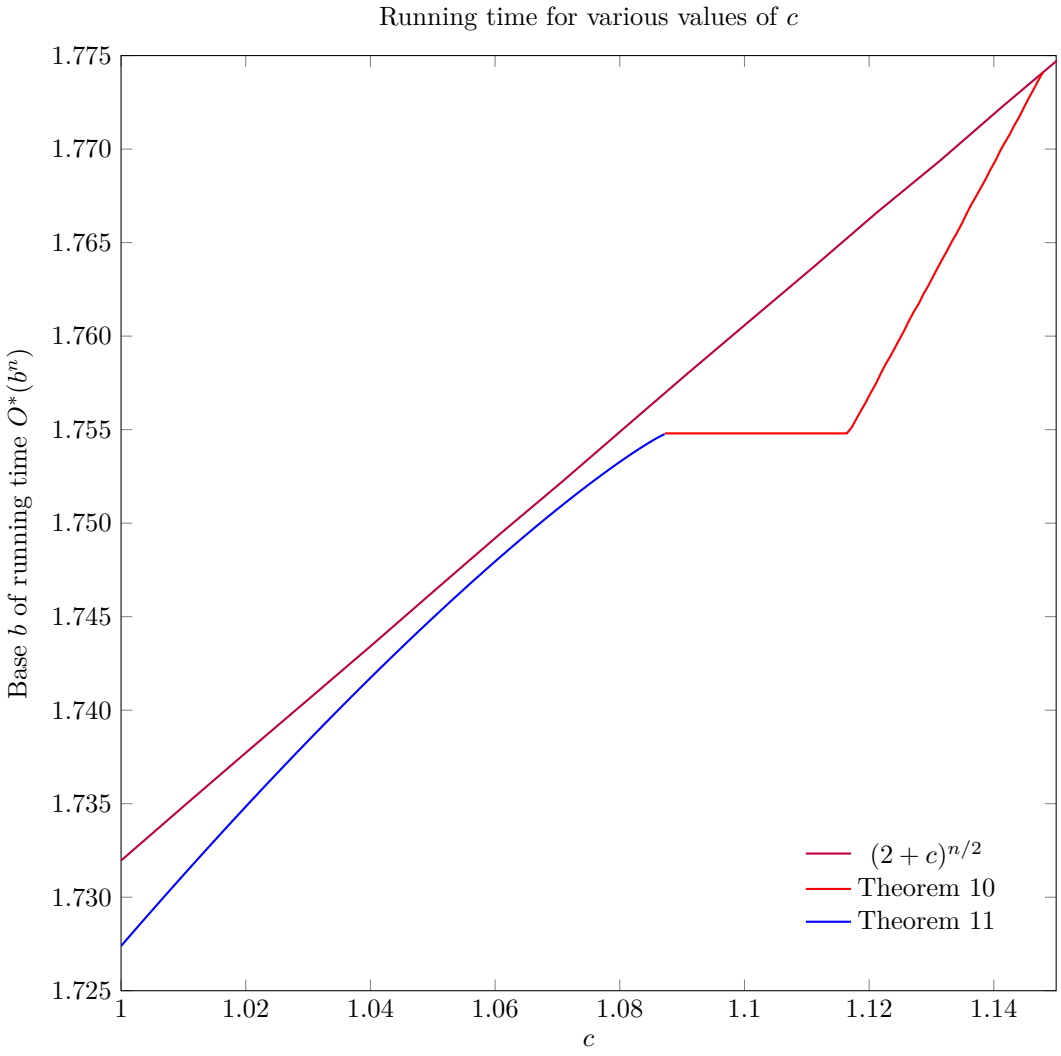

Figure 1 Visual presentation of running times of Theorem 10 and Theorem 11.

When $c \leq 1.147899$, then the running time provided by the best among Theorem 10 and Theorem 11 is slightly faster. At $c = 1$, the running time of Theorem 11 matches the running time of the Set Cover algorithm of [1] when $|\mathcal{F}|$ is subexponential in n.

When $c < -\frac{2}{3} + \sqrt[3]{\frac{47}{54} - \frac{\sqrt{93}}{18}} + \sqrt[3]{\frac{47}{54} + \frac{\sqrt{93}}{18}} \approx 1.147899$, then choosing the best algorithm among Theorem 10 and Theorem 11 gives an algorithm running in $O^*(b^n)$ time for some $b < \sqrt{c+2}$. This can be seen visually in Figure 1 and proved rigorously, e.g., by interleaving a stepwise function between the function $(c+2)^{n/2}$ and the function from Theorem 11 when $c \leq 1.08$:

$$\mathsf{steps}(c,n) = \begin{cases} 3^{n/2} & \text{if } c \leq 1.0123 \\ 1.735595^n & \text{if } 1.0123 \leq c \leq 1.0221 \\ \dots & \\ 1.7533^n & \text{if } 1.0742 \leq c \leq 1.08. \end{cases}$$

c	α	running time
1.0	0.236159	$O^*(1.7274^n)$
1.01	0.238036	$O^*(1.7312^n)$
1.02	0.239858	$O^*(1.7349^n)$
1.03	0.241622	$O^*(1.7384^n)$
1.04	0.243320	$O^*(1.7418^n)$
1.05	0.244946	$O^*(1.7450^n)$
1.06	0.246488	$O^*(1.7480^n)$
1.07	0.247928	$O^*(1.7508^n)$
1.08	0.249227	$O^*(1.7533^n)$

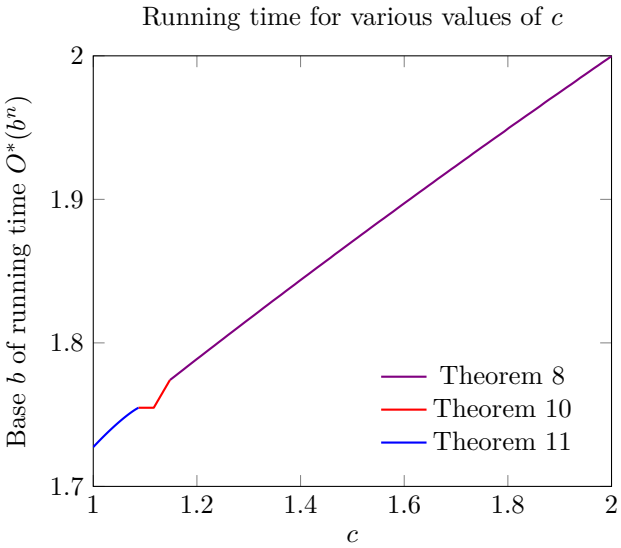

Figure 2 Running time for various values of c.

When $1.08 \leq c \leq 1.147899$, then Theorem 10 gives an algorithm with running time $O^*(d^n)$ for some $d < \sqrt{c+2}$. We list precise running times for certain values of c in Figure 2.

6 Applications

We can now use Corollary 13 in combination with simple branching algorithms enumerating various vertex sets in graphs, in particular algorithms enumerating maximal independent sets (equivalently, maximal cliques in the complement graph) in $O^*(3^{n/3})$ time [17] (we note that the algorithm of [24] is not a *simple* branching algorithm), maximal bicliques in $O^*(3^{n/3})$ time, maximal induced matchings in $O^*(10^{n/5})$ time [23], maximal induced forests in $O(1.8527^n)$ time [19], and minimal k-hitting sets in $O^*((2-1/k)^n)$ time [15]. The last result is used to enumerate maximal $\mathcal{H}$-free subgraphs, which have no induced subgraph isomorphic to any graph from the family $\mathcal{H}$ of graphs, all of which have at most k vertices. Some well-known $\mathcal{H}$-free graph classes are

- *cluster* graphs with $\mathcal{H} = \{P_3\}$, where P_k denotes the path on k vertices,
- *triangle-free* graphs with $\mathcal{H} = \{K_3\}$, where K_k denotes the complete graph on k vertices,
- *cographs* with $\mathcal{H} = \{P_4\}$ [12],
- *claw-free* graphs with $\mathcal{H} = \{K_{1,3}\}$, where $K_{k,\ell}$ denotes the complete bipartite graph with partite sets of size k and ℓ,
- *trivially-perfect* graphs with $\mathcal{H} = \{P_4, C_4\}$ [21], where C_k denotes the cycle on k vertices,
- *threshold* graphs with $\mathcal{H} = \{P_4, C_4, \overline{C_4}\}$ [11], where $\overline{G}$ denotes the complement of G,
- *split* graphs with $\mathcal{H} = \{C_4, \overline{C_4}, C_5\}$ [14], and
- *line* graphs, where $\mathcal{H}$ is a set of 9 graphs on at most 6 vertices each [3].

▶ **Theorem 14.** *There are bounded-error quantum algorithms, which, for a graph on n vertices and integer k, determine whether there is a k-packing, k-partitioning, or k-covering of G*

- *with maximal independent sets, maximal cliques, or maximal bicliques in $O(1.8554^n)$ time,*
- *with maximal cluster graphs or maximal triangle-free graphs in $O(1.9149^n)$ time,*

- *with maximal cographs, maximal claw-free graphs, maximal trivially-perfect graphs, or maximal threshold graphs in* $O(1.9365^n)$ *time,*
- *with maximal split graphs in* $O(1.9494^n)$ *time,*
- *with maximal line graphs in* $O(1.9579^n)$ *time, and*
- *with maximal induced forests in* $O(1.9629^n)$ *time.*

There are bounded-error quantum algorithms, which, for a graph on n *vertices and integer* k*, determine whether there is a* k*-packing or* k*-partitioning of* G

- *with maximal induced matchings in* $O(1.8934^n)$ *time*[2]*.*

In particular, this leads to a bounded-error quantum algorithm computing the chromatic number of an input graph in $O(1.8554^n)$ time; a graph can be covered with k maximal independent sets iff it has chromatic number at most k. In Section 7, we expedite this algorithm by exploiting fast algorithms for coloring a graph with a small number of colors and this will enable us to partition the vertex set in a more balanced way in the divide-and-conquer steps.

Even though there is a simple branching algorithm that enumerates all minimal dominating sets in $O(1.7159^n)$ time [16], this algorithm cannot be readily used for computing the domatic number using Theorem 8. This is because when we consider a subset X of the vertex set of $G = (V, E)$, we need to enumerate vertex subsets from X that are minimal dominating sets for G, and not $G[X]$. In Section 8, we prove that such minimal dominating sets can be enumerated in $O^*\left((2-\varepsilon)^{|X|}\right)$ time if $|X|$ is linear in $|V|$.

7 Faster Computation of the Chromatic Number

Assume the vertex set of input graph $G = (V, E)$ can be partitioned into independent sets $C = (I_1, I_2, \ldots, I_\chi)$ where χ is the chromatic number of G. Denote by n the number of vertices of G.

Using the algorithm from Theorem 8, we can compute the chromatic number of G in $O(1.8554^n)$ time. The family of subsets $\mathcal{F}$ corresponds to the independent sets of G and the family $e(X, r(\mathcal{F}, X))$ corresponds to the maximal independent sets of $G[X]$, which can be enumerated by a simple branching algorithm A in $O(3^{|X|/3})$ time [17]. We now prove a stronger result.

A proof of Lemma 17 along with a more detailed complexity analysis can be found in the full version of the paper [20].

▶ **Theorem 15.** *There is a bounded-error quantum algorithm, which for a graph on* n *vertices, computes the chromatic number in* $O(1.7956^n)$ *time.*

Assume, w.l.o.g., that $|I_1| \geq |I_2| \geq |I_3| \geq |I_4| \geq \cdots \geq |I_\chi|$.

In the first step, we use the algorithm from Theorem 6 with $\alpha = 0.27$ and store the result in QRAM. That is, for each α-small subset $X \subset V$, we find the chromatic number of X by finding the smallest k such that there exists a k-partition of X into independent sets. This takes $O^*(\binom{n}{0.27 \cdot n}) = O^*(1.79187^n)$ time.

We then consider a few different possibilities for the large sets in C and present an algorithm which finds a partition into a smallest number of independent sets for each of these cases. These possibilities cover all configurations of C so one result is guaranteed to find the chromatic number by detecting the partition C above, or an equivalent partition. The cases are as follows:

[2] Note that it is NP-hard to determine, for a graph $G = (V, E)$ and vertex subset $X \subseteq V$, whether there is a superset $Y \supseteq X$ that induces a 1-regular subgraph of G. This can be seen by a simple reduction from Independent Set.

1. G contains a vertex subset of size at least $0.48 \cdot n$ that is the union of at most five sets in C.
   We check for k-coloring for all $1 \leq k \leq n$ and take the minimum valid value of k as the chromatic number. To check for a certain k, we use Grover's search over all $Q \subset V$ where $|Q| \leq 0.52 \cdot n$ and check whether $G - Q$ is 5-colorable. If it is 5-colorable, we compute its chromatic number by checking its k'-colorability for $k' < 5$. We then solve the instance $(Q, r(\mathcal{F}, Q))$ to find a k_Q-coloring, where $k_Q = k - \chi(G - Q)$, using Grover's search to divide Q into (QL, QR) where $|QL| \geq 0.27 \cdot n$ and $k_{QL} + k_{QR} = k_Q$. For $(QL, r(\mathcal{F}, QL))$ (and similarly for QR) we use Grover's search on the branching algorithm A to enumerate candidate sets $S_1 \in e(QL, r(\mathcal{F}, QL))$; if $|QL| - |S_1| > 0.27 \cdot n$, then this branch is unsuccessful; otherwise, find $\chi(QL \setminus S_1)$ in QRAM.
   The running time of this case with quantum 5-coloring in $O^*(1.4695^n)$ [26] is $O^*(1.7831^n)$.
2. G contains a vertex subset of size at least $0.48 \cdot n$ that is the union of six sets in C, but does not contain a vertex subset of size at least $0.48 \cdot n$ that is the union of at most five sets in C.
   In this case we have

$$\begin{aligned} & |I_1| + |I_2| + \cdots + |I_5| < 0.48 \cdot n \\ \implies & 5 \cdot |I_6| \leq 5 \cdot |I_5| < 0.48 \cdot n \\ \implies & |I_1| + |I_2| + \cdots + |I_6| < 0.576 \cdot n, \end{aligned}$$

   which, along with the condition $0.48 \cdot n \leq |\bigcup_{i=1}^{6} I_i|$ gives

$$0.424 \cdot n < |V \setminus \bigcup_{i=1}^{6} I_i| \leq 0.52 \cdot n.$$

   For this case we check for k-coloring for all $1 \leq k \leq n$ and take the minimum valid value of k as the chromatic number. To check for a certain k, we use Grover's search over all $Q \subset V$ where $0.424 \cdot n < |Q| \leq 0.52 \cdot n$ and check whether $G - Q$ is 6-colorable. If so, we check whether the instance $(Q, r(\mathcal{F}, Q))$ is $(k-6)$-colorable in the same way as in item 1.
   The running time of this case with quantum 6-coloring in $O^*(1.5261^n)$ [26] is $O^*(1.7937^n)$.
3. G does not contain a vertex subset of size at least $0.48 \cdot n$ that is the union of at most six sets in C. That is $|\bigcup_{i=1}^{6} I_i| < 0.48 \cdot n$.
   Let $T = \bigcup_{i=1}^{q} I_i$ where q is the maximum index such that $|T| < \frac{n}{2}$. As $|\bigcup_{i=1}^{6} I_i| < 0.48 \cdot n$, we have $q \geq 6$. We consider two possibilities for the size of T and present an algorithm which finds a partition into a smallest number of independent sets for each case. As a valid coloring is computed in each case, the smallest partition gives the chromatic number if $|\bigcup_{i=1}^{6} I_i| < 0.48 \cdot n$.

   3.1. Consider the case where $|T| < \frac{6 \cdot n}{13}$:
   Let $L = \bigcup_{i=1}^{q+1} I_i$ and $R = \bigcup_{q+2}^{\chi} I_i$. We have

$$\begin{aligned} & |I_1| + |I_2| + \cdots + |I_q| < \frac{6 \cdot n}{13} \\ \implies & |I_1| + |I_2| + \cdots + |I_{q+1}| < \frac{7 \cdot n}{13} \\ \iff & |L| < \frac{7 \cdot n}{13} \end{aligned}$$

   and $0.5 \cdot n \leq |L|$ from the definition of q.

Assume L is not 7-colorable. There must be more than 7 independent sets in the construction of of L, which means

$$\begin{aligned} & q+1>7 \iff q \geq 7 \\ \implies & 7 \cdot |I_{q+1}| < \frac{6 \cdot n}{13} \\ \implies & |L| < \frac{48 \cdot n}{91} \end{aligned}$$

By contraposition, $|L| \geq \frac{48 \cdot n}{91}$ implies that L is 7-colorable.

▶ **Lemma 16.** *After running the algorithm presented in Theorem 6, there exists a bounded-error quantum algorithm to check the 7-colorability of a subset $X \subseteq U$ where $|X| \leq \frac{7}{3} \cdot \alpha \cdot n$ in $O^*(1.5622^{|X|})$ time.*

Proof. Assume that X is 7-colorable and there is a partition $D = (J_1, J_2, \ldots, J_7)$ of X into independent sets. Assume, w.l.o.g., that $|J_1| \geq |J_2| \geq \cdots \geq |J_7|$. Let $TL = J_1 \cup J_2 \cup J_3$ and $TR = J_4 \cup J_5 \cup J_6 \cup J_7$. We have

$$\begin{aligned} & \frac{|TL|}{3} \geq |J_3| \geq |J_4| \geq \frac{|TR|}{4} \\ \implies & |TL| + \frac{4}{3} \cdot |TL| \geq |X| \\ \iff & |TL| \geq \frac{3}{7} \cdot |X|. \end{aligned}$$

Consider $TR \setminus J_4$,

$$\begin{aligned} & |TR| \leq \frac{4}{7} \cdot |X| \text{ and } |J_4| \geq \frac{1}{4} \cdot |TR| \\ \implies & |TR \setminus J_4| \leq \frac{3}{7} \cdot |X| \leq \alpha \cdot n. \end{aligned}$$

So there exists a subset $S_1 \in e(TR, r(\mathcal{F}, TR))$ such that $|TR \setminus S_1| \leq \alpha \cdot n$.
We use Grover's search to divide X into two sets (XL, XR) where $|XL| \geq \frac{3 \cdot |X|}{7}$. We check XL for 3-colorability using a fast quantum algorithm. If it is 3-colorable, we use Grover's search on the branching algorithm A to enumerate candidate sets $S_1 \in e(XR, r(\mathcal{F}, XR))$; if $|XR| - |S_1| > \alpha \cdot n$, then this branch is unsuccessful; otherwise, check whether $\chi(XR \setminus S_1) \leq 3$ in QRAM. If X is 7-colorable and $(XL, XR) = (TL, TR)$, we indeed detect 7-colorability.
The running time of this algorithm with quantum 3-coloring in $O^*(1.1528^n)$ time [18] is $O^*(1.5622^{|X|})$. ◀

We check for k-coloring for all $1 \leq k \leq n$ and take the minimum valid value of k as the chromatic number. To check for a certain k, we use Grover's search to divide V into two sets (L, R) where $0.5 \cdot n \leq |L| < \frac{7 \cdot n}{13}$ and k into $k_L + k_R$. When $|L| < \frac{48 \cdot n}{91}$, we solve the subinstances $(L, r(\mathcal{F}, L))$ and $(R, r(\mathcal{F}, R))$ as in Theorem 8. When $|L| \geq \frac{48 \cdot n}{91}$, we solve the subinstance $(R, r(\mathcal{F}, R))$ as in Theorem 8 and run the algorithm in Lemma 16 on L; if L isn't 7-colorable, then this branch is unsuccessful; otherwise the subinstance $(L, r(\mathcal{F}, L))$ is k_L-colorable for $k_L \geq 7$. This algorithm finds the chromatic number of G when (L, R) are equal to the ones we constructed from C. The running time of this algorithm is $O^*(1.7956^n)$.

3.2. Otherwise, consider the case where $\frac{6 \cdot n}{13} \leq |T| < \frac{n}{2}$:
Let $L = T = \bigcup_{i=1}^{q} I_i$ and $R = \bigcup_{q+1}^{\chi} I_i$. We partition R into independent sets $C' = (I'_1, I'_2, \ldots, I'_{\chi'})$ where $I'_i = I_{q+i}$ for all $1 \leq i \leq \chi'$. By the definition of C, this is an optimal partition of R. As $q \geq 6$, we get $\frac{|L|}{6} \geq |I_q| \geq |I_{q+i}| = |I'_i|$ for all $1 \leq i \leq \chi'$.

▶ **Lemma 17.** *Assume we have a partition $D = (J_1, J_2, \ldots, J_m)$ of a set X where $t \geq |J_1| \geq |J_2| \geq \cdots \geq |J_m|$. Define $p = \left\lceil \frac{|X|}{t} \right\rceil$ and $r = \frac{|X|}{p}$. For any non-negative integer a, $1 \leq a \leq p-2$, there exists an integer k such that $a \cdot r \leq |\bigcup_{i=1}^{k} J_i| \leq (a+1) \cdot r$.*
Let $T_a = \bigcup_{i=1}^{q'} I'_i$ and $T_b = \bigcup_{i=1}^{q'+1} I'_i$ where q' is the maximum index such that $|T_a| < \frac{|R|}{2}$. Note that $\frac{|R|}{2} \leq |T_b|$. When we apply Lemma 17 with $X = R$, $D = C'$ and $t = \frac{|L|}{6}$, we get

$$\frac{|X|}{t} = \frac{6 \cdot |R|}{|L|} \leq \frac{6 \cdot \frac{7 \cdot n}{13}}{\frac{6 \cdot n}{13}} = 7$$
$$\text{and} \quad \frac{6 \cdot |R|}{|L|} > \frac{6 \cdot \frac{n}{2}}{\frac{n}{2}} = 6$$
$$\implies \qquad p = \left\lceil \frac{|X|}{t} \right\rceil = 7.$$

If we let $a = 3$, the lemma states that there exists a k such that $\frac{3 \cdot |R|}{7} \leq |\bigcup_{i=1}^{k} I'_i| \leq \frac{4 \cdot |R|}{7}$. As q' and $q'+1$ differ by 1, it is not possible that $|\bigcup_{i=1}^{q'} I'_i| < \frac{3 \cdot |R|}{7} \leq |\bigcup_{i=1}^{k} I'_i| \leq \frac{4 \cdot |R|}{7} < |\bigcup_{i=1}^{q'+1} I'_i|$. So, either $\frac{3 \cdot |R|}{7} \leq |T_a| < \frac{|R|}{2}$ or $\frac{|R|}{2} \leq |T_b| \leq \frac{4 \cdot |R|}{7}$. We let $TL = T_a$ or $TL = T_b$ such that $\frac{3 \cdot |R|}{7} \leq |TL| \leq \frac{4 \cdot |R|}{7}$ and $TR = R \setminus TL$. Note that in both cases, removing the independent set $I'_{q'+1}$ from either TL or TR would leave both subsets with a size $\leq \frac{|R|}{2}$. Therefore, there exists a subset $S_1 \in e(TL, r(\mathcal{F}, TL))$ (resp. for TR) such that $|TL \setminus S_1| \leq \frac{|R|}{2} \leq 0.27$ (and similarly for TR).

We check for k-coloring for all $1 \leq k \leq n$ and take the minimum valid value of k as the chromatic number. To check for a certain k, we use Grover's search to divide V into two sets (L, R) where $\frac{6 \cdot n}{13} \leq |L| < 0.5 \cdot n$ and k into $k_L + k_R$. We solve the subinstance $(L, r(\mathcal{F}, L))$ as in Theorem 8. We also solve the subinstance $(R, r(\mathcal{F}, R))$ as in Theorem 8, except that R is divided into (RL, RR) with $\frac{3 \cdot |R|}{7} \leq |RL| \leq \frac{4 \cdot |R|}{7}$ instead of $|RL| > 0.5 \cdot |R|$. This algorithm finds the chromatic number of G when (L, R) are equal to the ones we constructed from C and $(RL, RR) = (TL, TR)$. The running time of this algorithm is $O^*(1.7956^n)$.

We observe that the overall running time of the algorithm is

$$O^*\left(\sqrt{\binom{n}{\frac{7}{13} \cdot n}\binom{\frac{7}{13} \cdot n}{\frac{4}{13} \cdot n} 3^{\frac{1}{3} \cdot \frac{4}{13} \cdot n}}\right)$$
$$= O^*\left(\left(\frac{\sqrt{13}}{2^{7/13} \cdot 3^{23/78}}\right)^n\right)$$
$$= O^*(1.7956^n)$$

We reach this worst case when $\chi(G) = 13$ and $|I_1| = |I_2| = |I_3| = \cdots = |I_{13}|$. An example of such a graph is the disjoint union of $\frac{n}{13}$ complete graphs on 13 vertices. If the universe is partitioned into 2, one part must have at least 7 independent sets which is then partitioned into two parts where one part has at least 4 independent sets. Hence we cannot improve on the running time with a different twice-partitioning strategy.

We also note that the current best known quantum algorithms for checking 13-colorability, 7-colorability and 4-colorability [26] do not improve our running time. When iterating through $S_1 \in (X, r(\mathcal{F}, X))$, we only need to consider S_1 with $|S_1| \geq |X| - \alpha \cdot n$ so we can use an improved upper bound [10] when we only need to consider $|S_1| > |X| / 3$. However, this is only the case when $|X| > \frac{3}{2} \cdot \alpha \cdot n$; so it does not affect our overall running time as $|X| = \frac{4}{13} \cdot n$ in the worst case.

The pseudocode for the algorithm in this section can be found in [20].

8 Enumeration of Minimal Subset Dominating Sets

In this section, we prove that the number of minimal dominating sets of a graph that are subsets of some linear-sized subset of vertices X is at most $O^*\left((2-\varepsilon)^{|X|}\right)$. Moreover, they can be enumerated by a simple branching algorithm whose running time is within a polynomial factor of this bound.

▶ **Theorem 18.** *There is a simple branching algorithm, which, given any graph $G = (V, E)$ and any subset of vertices $X \subseteq V$ with $|X| \geq d \cdot |V|$ for some $d > 0$, enumerates all minimal dominating sets of G that are subsets of X in $O^*\left((2-\varepsilon_d)^{|X|}\right)$ time, for some $\varepsilon_d > 0$.*

The theorem will follow from a slightly more general theorem about minimal set covers of a set system. From a graph $G = (V, E)$ and a vertex subset $X \subseteq V$, we obtain a set system $(U, \mathcal{F})$ where $U = V$ and a set $S_x \in \mathcal{F}$ for each $x \in X$ that contains the closed neighborhood of vertex x in the graph G: $S_x = N_G[x]$. Then, there is a 1-to-1 correspondence between inclusion-wise minimal dominating sets in G that are subsets of X and inclusion-wise minimal set covers of $(U, \mathcal{F})$.

▶ **Theorem 19.** *There is a simple branching algorithm, which, given any set system $(U, \mathcal{F})$ with $|U| \leq r \cdot |\mathcal{F}|$ for some $r > 0$, enumerates all minimal set covers of $(U, \mathcal{F})$ in $O^*\left(2^{(1-\varepsilon_r)\cdot|\mathcal{F}|}\right)$ time, for some $\varepsilon_r > 0$.*

Fomin et al. [16] proved Theorem 18 for $X = V$ and Theorem 19 for $r = 1$. In particular, their algorithm enumerates all minimal dominating sets of a graph on n vertices in $O(1.7159^n)$ time.

Proof. Let $r > 0$. Let $\varepsilon = \frac{3(r+1)-\log(2^{3(r+1)}-1)}{3(r+1)^2}$. Consider the measure

$$\mu(U, \mathcal{F}) = (1 - (r+1)\varepsilon)|\mathcal{F}| + \varepsilon|U|$$

which associates a weight of $1 - (r+1)\varepsilon > 0$ to each set and a weight of $\varepsilon > 0$ to each element of a set system $(U, \mathcal{F})$. We will show that every set system $(U, \mathcal{F})$ has at most $2^{\mu(U,\mathcal{F})}$ minimal set covers by induction on $|\mathcal{F}|$. For a set system with $|U| \leq r \cdot |\mathcal{F}|$, this number is $2^{(1-(r+1)\varepsilon)|\mathcal{F}|+\varepsilon|U|} \leq 2^{(1-\varepsilon)\cdot|\mathcal{F}|}$. The proof can easily be turned into a simple branching algorithm enumerating all minimal set covers whose search tree is the induction tree of this proof.

The statement trivially holds when $|\mathcal{F}| = 0$ since such an instance has at most 1 minimal set cover. For the induction, we consider three cases.

1. $\mathcal{F}$ contains a set S with $|S| \geq 3(r+1)$. Any minimal set cover either contains S or not. Those that do not contain S are also minimal set covers of $(U, \mathcal{F} \setminus \{S\})$ and those that contain S are made up of S and a minimal set cover of $(U \setminus S, \mathcal{F} \setminus \{S\})$. For the induction, we would like that

$$\begin{aligned} & 2^{\mu(U,\mathcal{F}\setminus\{S\})} + 2^{\mu(U\setminus S,\mathcal{F}\setminus\{S\})} \leq 2^{\mu(U,\mathcal{F})} \\ \iff \quad & 2^{\mu(U,\mathcal{F})-(1-(r+1)\varepsilon)} + 2^{\mu(U,\mathcal{F})-(1-(r+1)\varepsilon)-|S|\cdot\varepsilon} \leq 2^{\mu(U,\mathcal{F})} \\ \iff \quad & 2^{-1+(r+1)\varepsilon} + 2^{-1+(r+1)\varepsilon-|S|\cdot\varepsilon} \leq 1 \\ \impliedby \quad & 2^{-1+(r+1)\varepsilon} + 2^{-1-2(r+1)\varepsilon} \leq 1, \end{aligned}$$

   and this inequality holds because $\varepsilon \leq \frac{1}{r+1} \log\left(\frac{1+\sqrt{5}}{2}\right)$.
2. There is an element $u \in U$ with frequency at most $3(r+1)$, i.e., u occurs in at most $3(r+1)$ sets of $\mathcal{F}$. Denote the sets that contain u by $\mathcal{S} = \{S \in \mathcal{F} : u \in S\}$. Since u needs to be covered, each set cover contains at least one set from $\mathcal{S}$. We use induction on all $2^{|\mathcal{S}|} - 1$ choices of including at least one set from $\mathcal{S}$ into the set covers and excluding the remaining sets from $\mathcal{S}$; each such choice leads to a set cover instance where we remove all sets in $\mathcal{S}$, and we remove all elements covered by the sets that are included in the set covers. Each such choice reduces the measure μ by more than $|\mathcal{S}| \cdot (1-(r+1)\varepsilon)$. Now, we would therefore like that

$$\begin{aligned} & (2^{|\mathcal{S}|}-1) \cdot 2^{\mu(U,\mathcal{F})-|\mathcal{S}|\cdot(1-(r+1)\varepsilon)} \leq 2^{\mu(U,\mathcal{F})} \\ \iff \quad & (2^{|\mathcal{S}|}-1) \cdot 2^{-|\mathcal{S}|\cdot(1-(r+1)\varepsilon)} \leq 1 \\ \iff \quad & 2^{-|\mathcal{S}|\cdot(1-(r+1)\varepsilon)} \leq (2^{|\mathcal{S}|}-1)^{-1} \\ \iff \quad & -|\mathcal{S}| \cdot (1-(r+1)\varepsilon) \leq -\log(2^{|\mathcal{S}|}-1) \\ \iff \quad & (r+1)\varepsilon - 1 \leq \frac{-\log(2^{|\mathcal{S}|}-1)}{|\mathcal{S}|} \\ \iff \quad & \varepsilon \leq \frac{|\mathcal{S}| - \log(2^{|\mathcal{S}|}-1)}{(r+1)|\mathcal{S}|} \end{aligned}$$

   Note that $\frac{|\mathcal{S}|-\log(2^{|\mathcal{S}|}-1)}{(r+1)|\mathcal{S}|}$ decreases when $|\mathcal{S}|$ increases, and for the maximum possible value of $|\mathcal{S}|$, which is $3(r+1)$, the inequality holds with equality for the value of ε given in the beginning of the proof.
3. It remains to consider the case where all sets have size less than $3(r+1)$ and all elements have frequency more than $3(r+1)$. Since the sum of set sizes equals the sum of element frequencies, we have that $|\mathcal{F}| \geq |U|$. Here, we use the result of Fomin et al. [16] who proved that the number of minimal set covers is at most $1.7159^{|\mathcal{F}|}$. For the induction, we would like that

$$\begin{aligned} & 1.7159^{|\mathcal{F}|} \leq 2^{\mu(U,\mathcal{F})} \\ \iff \quad & 2^{|\mathcal{F}|\log 1.7159} \leq 2^{(1-(r+1)\varepsilon)|\mathcal{F}|+\varepsilon|U|} \\ \impliedby \quad & 0.779|\mathcal{F}| \leq (1-(r+1)\varepsilon)|\mathcal{F}| \\ \iff \quad & \varepsilon \leq \frac{0.221}{r+1} \end{aligned}$$

   and this inequality holds for the value of ε given in the beginning of the proof.

This concludes the proof of the theorem. ◀

Theorem 8 now lets us conclude the following.

▶ **Corollary 20.** *There is a bounded-error quantum algorithm which computes the domatic number of any graph on n vertices in $O((2-\varepsilon)^n)$ time for some constant $\varepsilon > 0$.*

References

1 Andris Ambainis, Kaspars Balodis, Janis Iraids, Martins Kokainis, Krisjanis Prusis, and Jevgenijs 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.

2 Andris Ambainis and Martins Kokainis. Quantum algorithm for tree size estimation, with applications to backtracking and 2-player games. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2017)*, pages 989–1002. ACM, 2017. doi:10.1145/3055399.3055444.

3 Lowell W. Beineke. Characterizations of derived graphs. *Journal of Combinatorial Theory, Series B*, 9:129–135, 1970. doi:10.1016/S0021-9800(70)80019-9.

4 Andreas Björklund and Thore Husfeldt. Inclusion–exclusion algorithms for counting set partitions. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006)*, pages 575–582. IEEE Computer Society, 2006. doi:10.1109/FOCS.2006.41.

5 Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Fourier meets Möbius: fast subset convolution. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC 2007)*, pages 67–74. ACM, 2007. doi:10.1145/1250790.1250801.

6 Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Trimmed Moebius inversion and graphs of bounded degree. *Theory of Computing Systems*, 47(3):637–654, 2010. doi:10.1007/S00224-009-9185-7.

7 Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Covering and packing in linear space. *Information Processing Letters*, 111(21-22):1033–1036, 2011. doi:10.1016/J.IPL.2011.08.002.

8 Andreas Björklund, Thore Husfeldt, and Mikko Koivisto. Set partitioning via inclusion-exclusion. *SIAM Journal on Computing*, 39(2):546–563, 2009. doi:10.1137/070683933.

9 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\%3C493::AID-PROP493\%3E3.0.CO;2-P.

10 Jesper Makholm Byskov. Enumerating maximal independent sets with applications to graph colouring. *Operations Research Letters*, 32(6):547–556, 2004. doi:10.1016/j.orl.2004.03.002.

11 V. Chvátal and P. L. Hammer. Set-packing problem and threshold graphs. Technical report, University of Waterloo Research Report, 1973. CORR 73-21.

12 D. G. Corneil, H. Lerchs, and L. Stewart Burlingham. Complement reducible graphs. *Discrete Applied Mathematics*, 3(3):163–174, 1981. doi:10.1016/0166-218X(81)90013-5.

13 Christoph Dürr and Peter Høyer. A quantum algorithm for finding the minimum. *CoRR*, quant-ph/9607014, 1996. arXiv:quant-ph/9607014.

14 Stephan Földes and Peter L. Hammer. Split graphs. In *Proceedings of the 8th Southeastern Conference on Combinatorics, Graph Theory and Computing*, pages 311–315, 1977.

15 Fedor V. Fomin, Serge Gaspers, Daniel Lokshtanov, and Saket Saurabh. Exact algorithms via monotone local search. *Journal of the ACM*, 66(2):8:1–8:23, 2019. doi:10.1145/3284176.

16 Fedor V. Fomin, Fabrizio Grandoni, Artem V. Pyatkin, and Alexey A. Stepanov. Combinatorial bounds via measure and conquer: Bounding minimal dominating sets and applications. *ACM Transactions on Algorithms*, 5(1):9:1–9:17, 2008. doi:10.1145/1435375.1435384.

17 Fedor V. Fomin and Dieter Kratsch. *Exact Exponential Algorithms*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2010. doi:10.1007/978-3-642-16533-7.

18 Martin Fürer. Solving NP-complete problems with quantum search. In *Proceedings of the 8th Latin American Symposium on Theoretical Informatics (LATIN 2008)*, volume 4957 of *Lecture Notes in Computer Science*, pages 784–792. Springer, 2008. doi:10.1007/978-3-540-78773-0_67.

19 Serge Gaspers and Edward J. Lee. Exact algorithms via multivariate subroutines. In *Proceedings of the 44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, volume 80 of *LIPIcs*, pages 69:1–69:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.69.

20 Serge Gaspers and Jerry Zirui Li. Quantum algorithms for graph coloring and other partitioning, covering, and packing problems. *CoRR*, abs/2311.08042, 2023. doi:10.48550/arXiv.2311.08042.

21 Martin C. Golumbic. Trivially perfect graphs. *Discrete Mathematics*, 24(1):105–107, 1978. doi:10.1016/0012-365X(78)90178-4.

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

23 Sushmita Gupta, Venkatesh Raman, and Saket Saurabh. Maximum r-regular induced subgraph problem: Fast exponential algorithms and combinatorial bounds. *SIAM Journal on Discrete Mathematics*, 26(4):1758–1780, 2012. doi:10.1137/09077850X.

24 David S. Johnson, Mihalis Yannakakis, and Christos H. Papadimitriou. On generating all maximal independent sets. *Information Processing Letters*, 27(3):119–123, 1988. doi:10.1016/0020-0190(88)90065-8.

25 Mikko Koivisto. An $O^*(2^n)$ algorithm for graph coloring and other partitioning problems via inclusion–exclusion. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006)*, pages 583–590. IEEE Computer Society, 2006. doi:10.1109/FOCS.2006.11.

26 Kazuya Shimizu and Ryuhei Mori. Exponential-time quantum algorithms for graph coloring problems. *Algorithmica*, 84(12):3603–3621, 2022. doi:10.1007/S00453-022-00976-2.

27 F. Yates. The design and analysis of factorial experiments. *Imperial Bureau of Soil Science, Harpenden, England*, Technical Communication No. 35, 1937.