An Optimal Algorithm for Stochastic Vertex Cover*

Jan van den Brand
vdbrand@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia, USA

Inge Li Gørtz
inge@dtu.dk
Technical University of Denmark
Kongens Lyngby, Copenhagen
Denmark

Chirag Pabbaraju
cpabbara@stanford.edu
Stanford University
Stanford, California, USA

Debmalya Panigrahi
debmalya@cs.duke.edu
Duke University
Durham, North Carolina, USA

Clifford Stein
cliff@ieor.columbia.edu
Columbia University
New York City, New York, USA

Miltiadis Stouras
miltiadis.stouras@epfl.ch
EPFL
Lausanne, Switzerland

Ola Svensson
ola.svensson@epfl.ch
EPFL
Lausanne, Switzerland

Ali Vakilian
vakilian@vt.edu
Virginia Tech
Blacksburg, Virginia, USA

Abstract

The goal in the stochastic vertex cover problem is to obtain an approximately minimum vertex cover for a graph $G^\star$ that is realized by sampling each edge independently with some probability $p \in (0, 1]$ in a base graph $G = (V, E)$. The algorithm is given the base graph G and the probability p as inputs, but its only access to the realized graph $G^\star$ is through queries on individual edges in G that reveal the existence (or not) of the queried edge in $G^\star$. In this paper, we resolve the central open question for this problem: to find a $(1+\varepsilon)$-approximate vertex cover using only $O_\varepsilon(n/p)$ edge queries. Prior to our work, there were two incomparable state-of-the-art results for this problem: a $(3/2+\varepsilon)$-approximation using $O_\varepsilon(n/p)$ queries (Derakhshan, Durvasula, and Haghtalab, 2023) and a $(1+\varepsilon)$-approximation using $O_\varepsilon((n/p)\cdot \mathrm{RS}(n))$ queries (Derakhshan, Saneian, and Xun, 2025), where $\mathrm{RS}(n)$ is known to be at least $2^{\Omega\left(\frac{\log n}{\log\log n}\right)}$ and could be as large as $\frac{n}{2^{\Theta(\log^* n)}}$. Our improved upper bound of $O_\varepsilon(n/p)$ matches the known lower bound of $\Omega(n/p)$ for any constant-factor approximation algorithm for this problem (Behnezhad, Blum, and Derakhshan, 2022). A key tool in our result is a new concentration bound for the size of minimum vertex cover on random graphs, which might be of independent interest.

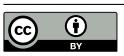
CCS Concepts

• **Theory of computation → Stochastic approximation**.

Keywords

Stochastic Vertex Cover, Approximation Algorithms

*A full version of this paper is available at https://arxiv.org/abs/2603.27795.



This work is licensed under a Creative Commons Attribution 4.0 International License.

STOC '26, Salt Lake City, UT, USA

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2536-4/2026/06
https://doi.org/10.1145/3798129.3800863

ACM Reference Format:
Jan van den Brand, Inge Li Gørtz, Chirag Pabbaraju, Debmalya Panigrahi, Clifford Stein, Miltiadis Stouras, Ola Svensson, and Ali Vakilian. 2026. An Optimal Algorithm for Stochastic Vertex Cover. In *Proceedings of the 58th Annual ACM Symposium on Theory of Computing (STOC '26), June 22–26, 2026, Salt Lake City, UT, USA*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3798129.3800863

1 Introduction

In the *stochastic vertex cover* problem, we are given a *base graph* $G = (V, E)$ and a sampling probability $p \in (0, 1]$. The *realized graph* $G^\star$ is a random graph generated by sampling each edge in G independently with probability p. The algorithm does not have access to $G^\star$ directly, but can query individual edges $e \in E$ to learn whether they appear in $G^\star$. The goal of the algorithm is to output a near-optimal vertex cover of $G^\star$ while querying as few edges in G as possible.

The stochastic setting has been widely considered in graph algorithms. Perhaps the most extensive literature exists for the stochastic matching problem, where the goal is to query a sparse subgraph H of the base graph G such that the realized edges in H contain an approximately maximum matching of the realized graph $G^\star$. For this problem, the first result was obtained by Blum et al. [13], who gave a 1/2-approximation algorithm using $n \cdot \mathrm{poly}(1/p)$ queries. This result has subsequently been improved in an extensive line of work [2–4, 7–10, 17], eventually culminating in an (almost tight) result that gives a $(1-\varepsilon)$-approximation using $n\cdot\mathrm{poly}(1/p)$ queries, for any fixed small $\varepsilon > 0$ [5]. Besides the maximum matching problem, stochastic optimization on graphs has also been considered for other classical problems such as minimum spanning tree and shortest paths [11, 14, 19, 21], as well as for more general frameworks in combinatorial optimization such as covering and packing problems [1, 12, 15, 20, 22].

In this paper, we consider the stochastic vertex cover problem. This problem was introduced by Behnezhad, Blum, and Derakhshan [6], who gave, for any $\varepsilon > 0$, a $(2+\varepsilon)$-approximation (polynomial-time) algorithm using $O\left(\frac{n}{\varepsilon^3 p}\right)$ queries. They also show

a simple lower bound that $\Omega(n/p)$ queries are necessary for any constant-factor approximation. The latter result is information-theoretic and rules out algorithms with fewer queries, even allowing an arbitrarily large running time. These results raised the question of whether there exist (exponential-time) algorithms that obtain a better-than-2 approximation, while still querying only $O(n/p)$ edges. This question was answered in the affirmative by Derakhshan, Durvasula, and Haghtalab [16], who obtained an approximation factor of $\left(\frac{3}{2}+\varepsilon\right)$, while querying $O\left(\frac{n}{\varepsilon p}\right)$ edges. This result, which helped delineate the information complexity of the problem by breaching the polynomial-time solvability barrier, led to the natural question: can we obtain a near-optimal $(1+\varepsilon)$-approximation algorithm using $O_\varepsilon(n/p)$ queries?

Interestingly, Behnezhad, Blum, and Derakshan had previously addressed this question for *bipartite* graphs: they obtained an $(1+\varepsilon)$-approximation using $O_{\varepsilon,p}(n)$ queries, although the dependence on $1/p$ and $1/\varepsilon$ were triple-exponential [6]. The first algorithm to obtain a $(1+\varepsilon)$-approximation for general graphs was obtained recently by Derakhshan, Saneian, and Xun [18], but their algorithm uses (super-linear) $O\left(\frac{n}{p}\cdot \mathrm{RS}(n)\right)$ queries. Here, RS refers to Ruzsa-Szemerédi Graphs and $\mathrm{RS}(n)$ is the largest β such that there exists an n-vertex graph whose edges can be partitioned into β induced, edge-disjoint matchings of size $\Theta(n)$. The value of $\mathrm{RS}(n)$ is known to be at least $2^{\Omega\left(\frac{\log n}{\log\log n}\right)}$ and could be as large as $\frac{n}{2^{\Theta(\log^* n)}}$.

Although the number of queries in this last result is super-linear, it applies to a more general setting. Previously, [16] had observed that under a regime permitting "mild" correlation between edge realizations, surpassing the $\frac{3}{2}$ factor requires $\Omega\left(n\cdot \mathrm{RS}(n)\right)$ queries. Since the result of [18] applies to this regime as well, their bound is tight in this mildly correlated setting. This left open the question of whether one can obtain a tight bound of $O_\varepsilon(n/p)$ in the original setting of independent edge sampling, or whether this dependence on the parameter $\mathrm{RS}(n)$ was fundamental to the problem even with full independence.

1.1 Our Result

In this paper, we give a $(1+\varepsilon)$-approximation algorithm for the stochastic vertex cover problem using $O_\varepsilon(n/p)$ queries, for any small $\varepsilon > 0$. By the lower bound of [6], our algorithm is *optimal*, up to the dependence on ε which is $1/\varepsilon^5$. To the best of our knowledge, the only previously known $(1+\varepsilon)$-approximation with linearly many (in n) queries was for bipartite graphs, but a linear dependence on $1/p$ (or polynomial dependence on $1/\varepsilon$) was not known even in this special case. Our result also shows that the dependence on $\mathrm{RS}(n)$ in [18] was an artifact of the correlations between edges, and in this sense, is not fundamental to the stochastic vertex cover problem itself.

Theorem 1.1. *For any $\varepsilon \in (0, c)$, where $c > 0$ is a small enough constant, there is a deterministic algorithm for the stochastic vertex cover problem that achieves an approximation factor of $1+\varepsilon$ using $O\left(\frac{n}{\varepsilon^5 p}\right)$ edge queries.*

Similar to all previous algorithms for stochastic vertex cover, our approximation factor is with respect to the *expected* size of the minimum vertex cover in the realized graph $G^\star$, and the set of edges queried by our algorithm is *non-adaptive*, i.e., independent of the realized graph $G^\star$.

1.2 Our Techniques

Description of the Algorithm. Our algorithm is simple, and in some sense, canonical among non-adaptive algorithms. Since the set of edges queried by the algorithm is non-adaptive, the output must deterministically include a valid vertex cover on the remaining (non-queried) edges in G to ensure feasibility. Call this deterministic vertex set P. Since P is always part of the output, it is wasteful to query any edge incident to P. So, we query the edges that are not incident to P, i.e., that are in the induced graph $G[V \setminus P]$. These queries reveal the realized graph $G^\star[V \setminus P]$; we compute its minimum vertex cover and add these vertices to P in the output. Finally, to choose P optimally, we solve an optimization problem that minimizes the expected size of the vertex cover output by the above algorithm, under the constraint that the number of edges in $G[V\setminus P]$ is at most $O_\varepsilon(n/p)$, our desired query bound. This optimal choice of P is denoted $\hat{P}$. We call this the Vertex-Cover algorithm and formally describe it in Section 2.3.

An Adaptive Algorithm as an Analysis Tool. While our algorithm is simple, its analysis is quite subtle. In the rest of this section, we give an outline of the main ideas in the analysis. Since our algorithm is defined via an optimization problem, it is difficult to directly compare its cost to opt, the expected size of an optimal vertex cover. Instead, we first define a surrogate algorithm that *adaptively* chooses a set $\mathrm{SEED}(G^\star)$ such that $G[V \setminus \mathrm{SEED}(G^\star)]$ has $O_\varepsilon(n/p)$ edges. Later, we will compare this adaptive strategy to our non-adaptive Vertex-Cover algorithm. The set $\mathrm{SEED}(G^\star)$ has two parts: a non-adaptive part $\mathrm{SEED}_{\mathrm{NA}}$ and an adaptive part $\mathrm{SEED}_{\mathrm{A}}(G^\star)$. First, we describe the choice of $\mathrm{SEED}_{\mathrm{NA}}$. Let $\mathrm{MVC}(H)$ denote a minimum vertex cover of any graph H. We partition the vertices into three groups, L, M and S, according to their probability to appear in $\mathrm{MVC}(G^\star)$. Vertices in L have "large" probability (at least $1-2\varepsilon$), those in M have "moderate" probability (between ε and $1-2\varepsilon$) and the ones in S have "small" probability (at most ε). Observe that we can safely include all vertices in L to $\mathrm{SEED}_{\mathrm{NA}}$. That is because $(1-2\varepsilon)\cdot|L| \geq \mathrm{opt}$, from which it follows that $\mathbb{E}[|L \setminus \mathrm{MVC}(G^\star)|] \leq O(\varepsilon)\cdot \mathrm{opt}$. Next, we can discard all vertices in S; these vertices will not appear in $\mathrm{SEED}(G^\star)$ for any $G^\star$. Using the fact that the vertices in S are infrequent in the optimal solution, we show that there are only $O_\varepsilon(n/p)$ edges that are entirely within S or between S and M, therefore these edges can be queried (notice that the edges between S and L will be covered by vertices in L). This allows us to focus on the subgraph $G[M]$ in the remaining discussion.

Deciding the Adaptive Part of $\mathrm{SEED}(G^\star)$. The remaining vertices, namely the set M, appear in $\mathrm{MVC}(G^\star)$ with probabilities ranging from ε to $1-2\varepsilon$. We cannot afford to add all vertices in M to $\mathrm{SEED}_{\mathrm{NA}}$, but we cannot discard all these vertices either. Instead, we use a greedy algorithm to select vertices in M to add to $\mathrm{SEED}_{\mathrm{NA}}$. A natural strategy would be to choose vertices with high degree in G. Indeed, if the degree of a vertex is at most $O_\varepsilon(1/p)$, we can query all its incident edges, and hence, it is redundant to add such a vertex to $\mathrm{SEED}(G^\star)$. But, in general, high-degree vertices in M may only

appear in $\mathrm{MVC}(G^\star)$ with probability ε, and as such, they might be numerous compared to opt. Since we cannot afford to add all these vertices to $\mathrm{SEED}_{\mathrm{NA}}$, we ask: which high-degree vertices should we prefer?

The answer to the above question lies at the heart of our analysis. Observe that if a high-degree vertex v is *not* in $\mathrm{MVC}(G^\star)$, then *all the neighbors of* v must be in $\mathrm{MVC}(G^\star)$. Using this observation, we define a deterministic procedure (called the Vertex-Seed algorithm) that outputs a sequence of vertices Q, where the i-th vertex v_i has the following property: with probability at least some constant δ over the choice of $G^\star$, the vertex v_i is not in $\mathrm{MVC}(G^\star)$ *and* has a large neighborhood in G among vertices whose status (whether in $\mathrm{MVC}(G^\star)$ or not) has not been "decided" by previous vertices in Q. The intuition is that such a vertex *reveals* a large number of previously undecided vertices to be in $\mathrm{MVC}(G^\star)$, and therefore, allows $|Q|$ to be bounded against opt. We add the vertices in Q to $\mathrm{SEED}_{\mathrm{NA}}$, and show that the expected size of $|Q \setminus \mathrm{MVC}(G^\star)|$ is at most $O(\varepsilon^2) \cdot \mathrm{opt}$ (a bound of $O(\varepsilon) \cdot \mathrm{opt}$ would suffice for now, but later, we will need the sharper bound of $O(\varepsilon^2) \cdot \mathrm{opt}$). Furthermore, we add the neighbors of vertices in $Q \setminus \mathrm{MVC}(G^\star)$ to the adaptive set $\mathrm{SEED}_{\mathrm{A}}(G^\star)$; since these vertices must be in $\mathrm{MVC}(G^\star)$, this does not affect the approximation bound.

Finally, we consider the vertices A that reveal a large number of previously undecided vertices to be in $\mathrm{MVC}(G^\star)$ for a specific realization $G^\star$, but are not in Q because they do not meet the probability threshold δ over the different realizations of $G^\star$. We add the vertices in A to the adaptive set $\mathrm{SEED}_{\mathrm{A}}(G^\star)$ as well, and show that the expected size of $A \setminus \mathrm{MVC}(G^\star)$ is also $O(\varepsilon) \cdot \mathrm{opt}$. This completes the description of the set $\mathrm{SEED}(G^\star)$. This last step ensures that the vertices outside $\mathrm{SEED}(G^\star)$ each have only $O_\varepsilon(1/p)$ edges in $G[M]$ that need to be queried.

Using $\mathrm{SEED}(G^*)$ *to Analyze Our Algorithm.* So far, we have described the adaptive set $\mathrm{SEED}(G^\star)$, and outlined intuition for two facts: (1) that $\mathrm{SEED}(G^\star)$ contains at most $(1 + O(\varepsilon)) \cdot \mathrm{opt}$ vertices in expectation, and (2) that there are at most $O_\varepsilon(n/p)$ edges in G that are not covered by $\mathrm{SEED}(G^\star)$. We remark that although $\mathrm{SEED}(G^\star)$ satisfies the conditions of the optimization problem in the Vertex-Cover algorithm, it does not immediately give an adaptive algorithm with $O_\varepsilon(n/p)$ queries. This is because the computation of the set $\mathrm{SEED}(G^\star)$ can require more than $O_\varepsilon(n/p)$ queries. So, the reader should view the definition of $\mathrm{SEED}(G^\star)$ strictly as an analysis tool, and not an alternative adaptive algorithm. Intuitively, we want to compare $\mathrm{SEED}(G^\star)$ to $\hat{P}$, the vertices chosen non-adaptively in the Vertex-Cover algorithm. If $\mathrm{SEED}(G^\star)$ were non-adaptive, this comparison is immediate since it would be a valid choice of S in the optimization problem defining the Vertex-Cover algorithm. But, in general, $\mathrm{SEED}(G^\star)$ can vary based on the realization of $G^\star$, and therefore, the expected size of $\mathrm{SEED}(G^\star)$ might be smaller than $|\hat{P}|$.

Dealing with the Adaptivity of $\mathrm{SEED}(G^\star)$. Note that $\mathrm{SEED}(G^\star)$ contains two parts: a non-adaptive set $\mathrm{SEED}_{\mathrm{NA}}$ and an adaptive set $\mathrm{SEED}_{\mathrm{A}}(G^\star)$. The set $\mathrm{SEED}_{\mathrm{A}}(G^\star)$ only depends on two random quantities: the identity of the set $Q \cap \mathrm{MVC}(G^\star)$ and the realizations of edges incident to vertices in $Q \setminus \mathrm{MVC}(G^\star)$. Importantly, our choice to include Q in $\mathrm{SEED}_{\mathrm{NA}}$ makes $\mathrm{SEED}(G^\star)$'s extension to a valid vertex cover (i.e. $\mathrm{MVC}(G^\star[V \setminus \mathrm{SEED}(G^\star)])$) independent of the realizations of the edges incident to $Q \cap \mathrm{MVC}(G^\star)$. This allows us to fix the realization of these edges, and analyze $\mathrm{SEED}(G^\star)$ over the remaining randomness in $G^\star$. This limits the adaptivity of $\mathrm{SEED}(G^\star)$, as it now only depends on the set $Q \cap \mathrm{MVC}(G^\star)$ which can take at most $2^{|Q|} = 2^{O(\varepsilon^2)\cdot \mathrm{opt}}$ values. In addition, we show that the size of the $\mathrm{SEED}(G^\star)$'s extension, namely $|\mathrm{MVC}(G^\star[V \setminus \mathrm{SEED}(G^\star)])|$, sharply concentrates. We do so by proving a general theorem on the convergence of $|\mathrm{MVC}(G^\star)|$ on a randomly generated graph $G^\star \sim G_p$:

Theorem 1.2. *Let* $Z = |\mathrm{MVC}(G^\star)|$, $\mathrm{opt} = \mathbb{E}_{G^\star \sim G_p}[|\mathrm{MVC}(G^\star)|]$. *Then for any* $t \geq 0$,

$$\Pr[|Z - \mathrm{opt}| \geq t] \leq 2\exp\left(-\frac{t^2}{4C \cdot \mathrm{opt} + 2t/3}\right), \tag{1}$$

where $C < 8$ *is a constant.*

Note that this theorem is not specific to our construction and establishes a general concentration result for minimum vertex cover. The tail bound proven is much sharper than what one can obtain from standard techniques (e.g., vertex exposure martingales) and we believe it might be of independent combinatorial interest.[1]

Finally, we use a union bound over the $2^{O(\varepsilon^2)\cdot \mathrm{opt}}$ different realizations of $Q \cap \mathrm{MVC}(G^\star)$, for each of which the tail bound on the size of $\mathrm{MVC}(G^\star[V \setminus \mathrm{SEED}_{\mathrm{NA}}])$ applies. Using this, we can now claim that the advantage of adaptivity in defining $\mathrm{SEED}(G^\star)$ is negligible. Formally, we show that, for any fixed realization of the edges incident to Q, the set $\mathrm{SEED}(G^\star)$ can take at most $2^{O(\varepsilon)\cdot \mathrm{opt}}$ different forms for the graphs $G^\star$ that are consistent with the fixed realization of edges incident to Q. Out of those forms, the one that minimizes the expected size of $\mathrm{SEED}(G^\star) \cup \mathrm{MVC}(G^\star[V \setminus \mathrm{SEED}(G^\star)])$ is at most $(1 + O(\varepsilon))$ worse than adaptively selecting the best of these forms for each graph $G^\star$. Since the latter holds for any realization of the edges incident to Q, averaging over their randomness gives us that there exists a deterministic set P that produces a solution of expected size $(1 + O(\varepsilon))\mathrm{opt}$. This, in turn, establishes that our algorithm, which optimizes over all non-adaptive choices of P, is a $(1 + O(\varepsilon))$-approximation algorithm.

2 The Vertex-Cover Algorithm

In this section, we formally define the stochastic vertex cover problem, and establish notation that we will use throughout the paper. Then, we give a formal description of our Vertex-Cover algorithm.

2.1 Notation and Terminology

Throughout, let $G = (V, E)$ denote the base graph, with $n = |V|$. Fix an edge-realization parameter $p \in (0, 1]$; each edge $e \in E$ is realized independently with probability p. All results extend to the heterogeneous model with edge-wise probabilities $(p_e)_{e\in E}$ by replacing p in the statements with $p := \min_{e\in E} p_e$. For clarity of exposition, throughout the paper we present the homogeneous case $p_e = p$ for all $e \in E$.

[1] A strengthened version of Talagrand's inequality for c-Lipschitz, s-certifiable functions also yields a slightly weaker concentration bound, which still suffices to derive our results.

Let $G^\star$ be the random realized subgraph obtained by including each $e \in E$ independently with probability p; we write $G^\star \sim G_p$. For a set of edges $F \subseteq E$, we use $G^\star \setminus F^\star$ to denote the distribution obtained by realizing all edges in $E \setminus F$ as above while leaving the edges in F unresolved (to be realized later). An *edge query* reveals whether a particular $e \in E$ is present in $G^\star$.

For a graph $H = (V, E)$ and $v \in V$, let $N_H(v)$ be the set of neighbors of v in H; for $S \subseteq V$, let $N_H(S) = \{u \in V \setminus S : \exists s \in S \text{ with } (u, s) \in E\}$. Let $\mathrm{MVC}(\cdot)$ be the mapping that assigns to every realized graph $G^\star$ an arbitrary but fixed minimum vertex cover $\mathrm{MVC}(G^\star) = S \subseteq V$. Let $\mathrm{MVC}(G^\star)$ denote the resulting (random) minimum vertex cover on the realized graph, and define $\mathrm{opt} = \mathbb{E}_{G^\star \sim G_p}\left[\left|\mathrm{MVC}(G^\star)\right|\right]$. For each $v \in V$, set $c_v = \Pr\left[v \in \mathrm{MVC}(G^\star)\right]$, so that $\sum_{v \in V} c_v = \mathrm{opt}$. For an edge $e = (u, v) \in E$ define the probability that edge e is covered as $c_e = \Pr[u \in \mathrm{MVC}(G^\star) \text{ or } v \in \mathrm{MVC}(G^\star)]$.

2.2 The Stochastic Vertex Cover Problem

Given a base graph $G = (V, E)$ and a probability parameter p, in the stochastic vertex cover problem, the goal is to output a feasible vertex cover of the realized graph $G^\star$, in which each edge of G is realized in $G^\star$ independently with probability p, while querying as few edges in G as possible. The algorithm is allowed unlimited access to the base graph G as well as unlimited computation time. For an $\alpha > 0$, we say a (randomized) solution C is an α-approximate stochastic vertex cover, if any edge in $G^\star$ has at least one of its endpoint in C, and $\mathbb{E}[|C|] \le \alpha \cdot \mathbb{E}[\left|\mathrm{MVC}(G^\star)\right|]$.

In this paper, we give a non-adaptive $(1 + \varepsilon)$-approximation algorithm for the stochastic vertex cover problem, i.e., it queries a fixed set of edges chosen in advance, independent of all query outcomes.

2.3 Description of the Vertex-Cover Algorithm

First, we choose $P \subseteq V$ minimizing $|P| + \mathbb{E}\left[\left|\mathrm{MVC}(G^\star[V \setminus P])\right|\right]$ under the constraint that the induced subgraph $G[V \setminus P]$ contains at most $O(n/(\varepsilon^5 p))$ edges. Let that set be $\hat{P}$. We then query all edges in the induced subgraph $G[V \setminus \hat{P}]$ and compute the minimum vertex cover H of the now known, realized graph $G^\star[V \setminus \hat{P}]$. Finally, we return the set $\hat{P} \cup H$ as our vertex cover. Recall, as in previous papers, that we are not concerned with computational efficiency, but only the query complexity. The pseudocode for the algorithm is given in Figure 1.

It is immediate that this algorithm correctly produces a valid vertex cover:

Claim 2.1. *The output of the* Vertex-Cover *algorithm is a vertex cover for* $G^\star$.

Proof. Each edge either has an endpoint in $\hat{P}$ or is an edge in the induced subgraph $G^\star[V \setminus \hat{P}]$. □

3 Analysis of the Vertex-Cover Algorithm

In this section we analyse the Vertex-Cover algorithm using our surrogate algorithm. In the first two subsections we describe the Vertex-Seed algorithm and bound the size of the set Q chosen by the algorithm. Next we describe how to choose the adaptive set

Algorithm: Vertex-Cover

(1) Let $\hat{P}$ be the optimal solution to

$$\min_{P \subseteq V} \quad |P| + \mathbb{E}\left[\left|\mathrm{MVC}(G^\star[V \setminus P])\right|\right] \qquad \text{s.t.} \quad G[V \setminus P] \text{ has at most } 2 \cdot \frac{10^3\, n}{\varepsilon^5 p} \text{ edges.} \tag{2}$$

(2) Query all the edges in $G[V \setminus \hat{P}]$ to get its realization $G^\star[V \setminus \hat{P}]$.

(3) Let $H = \mathrm{MVC}\left(G^\star[V \setminus \hat{P}]\right)$.

(4) Return $\hat{P} \cup H$.

Figure 1: The Vertex-Cover algorithm.

$\mathrm{SEED}(G^\star)$ and prove that it contains at most $(1+O(\varepsilon)) \cdot \mathrm{opt}$ vertices in expectation. Finally, we prove that there are at most $O_\varepsilon(n/p)$ edges in G that are not covered by $\mathrm{SEED}(G^\star)$, and analyze the performance of the Vertex-Cover algorithm.

Throughout the analysis, we assume that $\mathrm{opt} \ge c \cdot \log(1/\varepsilon)/\varepsilon^2$ for some constant c; this assumption is without loss of generality: if $\mathrm{opt} = O(\log(1/\varepsilon)/\varepsilon^2)$, then the base graph G contains $O(n/(p\varepsilon^3))$ edges and the Vertex-Cover algorithm can select $\hat{P} = \emptyset$, query all of G and obtain an exact solution.

3.1 The Vertex-Seed Algorithm

In this section, we describe the Vertex-Seed algorithm, a deterministic algorithm that returns a sequence of vertices $Q = (v_1, v_2, \ldots, v_k)$ in the base graph G depending only on a vertex cover function $\mathrm{VC}(G^\star)$ that maps every realization $G^\star$ to a fixed feasible vertex cover of $G^\star$. The intuition is that each vertex v_i in Q corresponds to a query of the type "is $v_i \in \mathrm{VC}(G^\star)$?" If the answer is negative, i.e. $v_i \notin \mathrm{VC}(G^\star)$, then all neighbors of v_i in $G^\star$ are necessarily in $\mathrm{VC}(G^\star)$. Our goal in selecting Q is to keep $|Q|$ small while revealing a large number of vertices in $\mathrm{VC}(G^\star)$ by virtue of being neighbors of vertices in $Q \setminus \mathrm{VC}(G^\star)$. The Vertex-Seed algorithm describes a greedy procedure for incrementally constructing Q with this purpose. As input, in addition to the base graph G, the Vertex-Seed algorithm takes two parameters, δ and γ, which will be defined later.

Before defining the algorithm, we introduce some notation:

- For a sequence of vertices $Q_i = (v_1, v_2, \ldots, v_i)$ and a fixed realization $G^\star$, define

$$\mathsf{decided}(Q_i, G^\star) = \left\{v \in V \setminus Q_i : N_{G^\star}(v) \cap (Q_i \setminus \mathrm{VC}(G^\star)) \neq \emptyset\right\}. \tag{3}$$

  In words, a vertex v is in $\mathsf{decided}(Q_i, G^\star)$ if it has a neighbor that is in Q_i but not in $\mathrm{VC}(G^\star)$. Note that a vertex $v \in \mathsf{decided}(Q_i, G^\star)$ necessarily belongs to $\mathrm{VC}(G^\star)$, by virtue of feasibility of $\mathrm{VC}(G^\star)$. We further let $\mathsf{undecided}(Q_i, G^\star) = (V \setminus Q_i) \setminus \mathsf{decided}(Q_i, G^\star)$. Thus, the sets $\mathsf{decided}(Q_i, G^\star)$ and $\mathsf{undecided}(Q_i, G^\star)$ induce a partition of the vertices in $V \setminus Q_i$.

h

Algorithm: Vertex-Seed

Initialize Q to be the empty sequence.

While there is a vertex v such that

$$\Pr_{G^\star}[v \in \text{revealing}(Q, G^\star)] \geq \delta,$$

append v to Q.

Figure 2: The Vertex-Seed algorithm.

- For a sequence of vertices $Q_i = (v_1, v_2, \ldots, v_i)$ and a fixed realization $G^\star$, define

$$\text{revealing}(Q_i, G^\star) = \left\{ v \in V \setminus Q_i : v \notin \text{VC}(G^\star) \text{ and } |N_G(v) \cap \text{undecided}(Q_i, G^\star)| \geq \frac{1}{p \cdot \gamma} \right\}. \quad (4)$$

  In words, a vertex v is in $\text{revealing}(Q_i, G^\star)$ if it is not in $\text{VC}(G^\star)$ *and* has at least $\frac{1}{p \cdot \gamma}$ neighbors in the base graph G that are in $\text{undecided}(Q_i, G^\star)$. Intuitively, a vertex $v \in \text{revealing}(Q_i, G^\star)$ is a good candidate for extending Q_i to Q_{i+1} since it is likely to move a large number of vertices from $\text{undecided}(Q_i, G^\star)$ to $\text{decided}(Q_{i+1}, G^\star)$.

The Vertex-Seed algorithm starts with Q being empty and then constructs it iteratively by adding vertices that have a high probability of being in $\text{revealing}(Q, G^\star)$. The pseudocode is given in Figure 2.

We use the notation $Q_i = (v_1, \ldots, v_i)$ to denote the set of vertices in Q after i iterations of the Vertex-Seed algorithm: Q_0 denotes the initial empty set, and Q_k denotes the final set if the algorithm terminates after k iterations. Note that the sequence of computed Q_i's is deterministic, and when we measure probability within the loop (namely $\Pr_{G^\star}[v \in \text{revealing}(Q_i, G^\star)]$), it is only over the draw of $G^\star$. Furthermore, note that the algorithm necessarily terminates after at most n iterations.

3.2 Bounding the Number of Vertices in Q

Observe that the Vertex-Seed algorithm is deterministic and hence, the number of vertices in Q is also deterministic. The next lemma bounds this number.

Lemma 3.1. *Let $Q = (v_1, \ldots, v_k)$ be the sequence of vertices constructed by the* Vertex-Seed *algorithm. For any $n \geq 4\log(2/\delta)$, we have that $k \leq \frac{10\gamma}{\delta} \cdot n$.*

First, we give the intuition behind the lemma, before we formally prove it. Suppose a vertex $v \in \text{revealing}(Q, G^\star)$ is added to Q by the Vertex-Seed algorithm. Then, it has $\frac{1}{p \cdot \gamma}$ neighbors in G that are currently undecided. Informally, we should expect that $1/\gamma$ of these neighbors realize in $G^\star$ and, therefore, are moved from $\text{undecided}(Q, G^\star)$ to $\text{decided}(Q, G^\star)$ as a result of adding v to Q. Since there are only n vertices in G, the number of times this can happen is $\gamma \cdot n$. Finally, since the event $v \in \text{revealing}(Q, G^\star)$ holds with probability δ for every vertex added to Q, we should expect that Q can only contain $\frac{\gamma \cdot n}{\delta}$ vertices.

The argument above is not formal because the expected number of vertices moved from $\text{undecided}(Q, G^\star)$ to $\text{decided}(Q, G^\star)$ by a vertex $v \in \text{revealing}(Q, G^\star)$ is not necessarily $1/\gamma$, since the neighborhood of v in $G^\star$ is not independent of the event $v \in \text{revealing}(Q, G^\star)$. Nevertheless, we show that this intuition holds, and can be made formal, in the proof below.

Proof of Lemma 3.1. For $i = 1, \ldots, k$, let X_i be the (random) indicator variable that is 1 if $v_i \in \text{revealing}(Q_{i-1}, G^\star)$ and 0 otherwise. Note that $\mathbb{E}[X_i] \geq \delta$ for all i by definition of the Vertex-Seed algorithm. Let $X = \sum_{i=1}^{k} X_i$. Thus, $\mathbb{E}[X] = \sum_{i=1}^{k} \mathbb{E}[X_i] \geq k \cdot \delta$.

Let $u = \frac{10\gamma}{\delta} \cdot n$. Recall that we want to show that $k \leq u$. We consider two cases, $X \leq u \cdot \delta/2$ and $X > u \cdot \delta/2$, and write

$$k \cdot \delta \leq \mathbb{E}[X] \leq u \cdot \delta/2 + k \cdot \Pr[X > u \cdot \delta/2]. \quad (5)$$

To complete the proof, we will show that $\Pr[X > u \cdot \delta/2] \leq \delta/2$, which implies $k \leq u$ as required.

Now let $\mathcal{G}$ be the subset of realizations of $G^\star$ for which $X \geq u \cdot \delta/2$. In other words, we want to show that $\Pr[G^\star \in \mathcal{G}] \leq \delta/2$. We further partition $\mathcal{G}$ as follows: for each $S \subseteq Q$ such that $|S| \geq u \cdot \delta/2$, we let $\mathcal{G}_S \subseteq \mathcal{G}$ be the realizations of $G^\star$ for which $\{v_i \in Q \mid v_i \in \text{revealing}(Q_{i-1}, G^\star)\} = S$. These chosen $\mathcal{G}_S$ partition $\mathcal{G}$ since, by definition of $\mathcal{G}$, every realization $G^\star \in \mathcal{G}$ has

$$X = |\{v_i \in Q \mid v_i \in \text{revealing}(Q_{i-1}, G^\star)\}| \geq u \cdot \delta/2.$$

We thus have that $\mathcal{G}$ is partitioned into at most $\sum_{\ell=u\delta/2}^{k} \binom{k}{\ell} \leq 2^k$ many sets $\mathcal{G}_S$.

We proceed to bound $\Pr[G^\star \in \mathcal{G}_S]$ for a fixed set S. To do this, the following viewpoint for generating a realization $G^\star$ will be helpful. There is a random string $R = (r_1, r_2, \ldots)$ of bits where each r_i is a random bit that is 1 with probability p and 0 otherwise. The realization $G^\star$ is generated as follows:

- For each $v_i \in S$ (in the order it was added to Q), the existence of an edge $e = (v_i, v)$ in $G^\star$ is determined by the next random bit in R if the following properties are satisfied by v:
  - v is a neighbor of v_i in the base graph G,
  - v is not in $\{v_1, \ldots, v_{i-1}\}$, and
  - there is no edge (v_j, v) realized so far in $G^\star$ such that v_j is a vertex in $\{v_1, \ldots, v_{i-1}\} \cap S$.
- Any remaining edge not considered above is simply realized with probability p independently.

The above procedure simply realizes each edge with probability p independently but distinguishes (as a function of S) certain random bits with R.

Now, we claim that if any $G^\star$ realized this way ends up belonging to $\mathcal{G}_S$, then we must have used at least $5n/p$ bits from R. To see this, consider the realization of edges adjacent to some fixed vertex $v_i \in S$ in the process above. Since the realized graph $G^\star \subseteq \mathcal{G}_S$, it must be the case that $v_i \in \text{revealing}(Q_{i-1}, G^\star)$, which means that v_i has at least $1/(p \cdot \gamma)$ neighbors in G that are in $\text{undecided}(Q_{i-1}, G^\star)$. Since $\text{undecided}(Q_{i-1}, G^\star) \subseteq V \setminus Q_{i-1}$, all these neighbors are not in Q_{i-1}. Moreover, since each vertex $v_i \in S$ is in $\text{revealing}(Q_{i-1}, G^\star)$, we have that none of the vertices in S are in $\text{VC}(G^\star)$. Therefore, if a vertex v is in $\text{undecided}(Q_{i-1}, G^\star)$, its neighborhood in $G^\star$ is disjoint from $Q_{i-1} \cap S$. As a result, v_i has at least $1/(p \cdot \gamma)$ neighbors v in G, such that $v \notin Q_{i-1}$, and furthermore, the neighborhood of v in $G^\star$ is disjoint from $Q_{i-1} \cap S$.

For any such neighbor v of v_i, the process described above uses a new bit from R to confirm the presence of the edge (v_i, v). In total, the number of potential edges determined using the random bits from R is hence at least

$$\frac{|S|}{p \cdot \gamma} \geq \frac{u \cdot \delta/2}{p \cdot \gamma} = \frac{5n}{p}, \text{ since } u = \frac{10\gamma}{\delta} \cdot n.$$

Observe also that if an edge (v_i, v) is confirmed to be in $G^\star$ using a bit from R, then no other edge (v_j, v) for $j > i$ is determined using a bit from R. This means that the number of edges confirmed to be in $G^\star$ using bits from R can be at most the number of distinct vertices in G, which is n. In summary, we conclude that, if the realized graph $G^\star \in \mathcal{G}_S$, then it must be the case that we used at least $5n/p$ bits from R, and at most n of these bits realized edges in $G^\star$. So, consider the first $5n/p$ bits in R. As the expectation of the number of ones in the independent Bernoulli trials $r_1, \ldots, r_{5n/p}$ is $5n$, we have by a standard Chernoff bound that

$$\Pr[G^\star \in \mathcal{G}_S] \leq \Pr\left[\sum_{j=1}^{5n/p} r_j \leq \left(1 - \frac{4}{5}\right) 5n\right] \leq \exp\left(\frac{-5n \cdot (4/5)^2}{3}\right) \leq e^{-n}.$$

Finally, by a union bound over the partitioning of $\mathcal{G}$,

$$\Pr[G^\star \in \mathcal{G}] \leq 2^k \cdot e^{-n} \leq (2/e)^n \leq \delta/2,$$

where the last inequality holds for $n \geq 4\log(2/\delta)$. So, by Eq. (5), $k \leq \frac{10\gamma}{\delta} \cdot n$ as required. □

3.3 Selection of the Vertex Set SEED($G^\star$)

We can use the Vertex-Seed algorithm directly on the base graph G and add the vertices in Q to the set SEED($G^\star$). But, this results in a set Q whose size is independent of the expected size of MVC($G^\star$), which translates to an additive error in the approximation bound of the algorithm. Instead, we use the Vertex-Seed algorithm in a more nuanced fashion that avoids this additive loss.

We partition vertices V in G into three sets:

- The set $L := \{v \in V : \Pr_{G^\star}[v \in \text{MVC}(G^\star)] \geq 1 - 2\varepsilon\}$ of vertices that have *large* probability of being in MVC($G^\star$),
- the set $M := \{v \in V : \varepsilon < \Pr_{G^\star}[v \in \text{MVC}(G^\star)] < 1 - 2\varepsilon\}$ of vertices that have *moderate* probability of being in MVC($G^\star$), and
- the set $S := \{v \in V : \Pr_{G^\star}[v \in \text{MVC}(G^\star)] \leq \varepsilon\}$ of vertices that have *small* probability of being in MVC($G^\star$).

Recall that opt denotes the expected size of the minimum vertex cover, i.e., opt $:= \mathbb{E}[\text{MVC}(G^\star)]$. In the next two claims, we bound the number of vertices in L and M, in terms of opt. Note that L and M are deterministic sets, but MVC($G^\star$) is random.

Claim 3.2. *For any $\varepsilon \leq 1/4$, $\mathbb{E}[L \setminus \text{MVC}(G^\star)]$ is at most $4\varepsilon \cdot$ opt.*

Proof. Note that for any vertex $v \in L$, we have $\Pr[v \in L \setminus \text{MVC}(G^\star)] \leq 2\varepsilon$. Therefore,

$$\mathbb{E}[|L \setminus \text{MVC}(G^\star)|] \leq 2\varepsilon \cdot |L|.$$

Using this bound, we get

$$|L| \leq \mathbb{E}[|L \setminus \text{MVC}(G^\star)|] + \mathbb{E}[|\text{MVC}(G^\star)|] \leq 2\varepsilon \cdot |L| + \text{opt},$$

which implies that $|L| \leq \text{opt}/(1 - 2\varepsilon)$. Therefore,

$$\mathbb{E}[|L \setminus \text{MVC}(G^\star)|] \leq 2\varepsilon \cdot |L| \leq \frac{2\varepsilon}{1 - 2\varepsilon} \cdot \text{opt} \leq 4\varepsilon \cdot \text{opt}, \text{ for } \varepsilon \leq 1/4.$$

□

Claim 3.3. *The number of vertices in M is at most* opt$/\varepsilon$.

Proof. Every vertex in M is in MVC($G^\star$) with probability at least ε, and opt is at least the expected number of vertices in M that are in MVC($G^\star$). The claim follows. □

Now, we run the Vertex-Seed algorithm on the induced graph $G[M]$ and designate the output set Q. We also use $G^\star[M]$ to denote the induced subgraph on M of the realized graph $G^\star$. In the Vertex-Seed algorithm, we use the following parameters:

- The parameter δ, which is used as the probability threshold for a vertex to be added to Q, is set to $\delta := \varepsilon^2$.
- The parameter γ, which decides the threshold on the number of undecided neighbors in the base graph G for a vertex to be deemed revealing, is set to $\gamma := \varepsilon^3 \cdot \delta/10^3 = \varepsilon^5/10^3$.

Furthermore, the vertex cover VC($G^\star[M]$) used in the Vertex-Seed algorithm is set to MVC($G^\star$) $\cap M$. Clearly, this is a feasible vertex cover of $G^\star[M]$.

Based on this choice of parameters, we can bound the number of vertices in Q. Recall that Vertex-Seed is a deterministic algorithm and it is run on a deterministic graph $G[M]$. Hence, Q is deterministic.

Claim 3.4. *The size of the set Q output by the* Vertex-Seed *algorithm when run on $G[M]$ is at most $(\varepsilon^2/100) \cdot$ opt.*

Proof. By Lemma 3.1 and our choice of parameters, we get $|Q| \leq (\varepsilon^3/100) \cdot |M|$. Now, the claim follows from the bound on M in Claim 3.3. □

We are now ready to define the vertex set SEED($G^\star$) for any fixed realized graph $G^\star$. We define SEED($G^\star$) as the union of four sets described below. The first two sets, L and Q, do not depend on the realized graph $G^\star$, whereas the last two sets depend on $G^\star$.

- the set L of vertices that have large probability of being in MVC($G^\star$),
- the set $Q \subseteq M$ of vertices returned by the Vertex-Seed algorithm on $G[M]$,
- the set of vertices decided$(Q, G^\star[M])$, and
- the set of vertices

$$A := \left\{v \in M \setminus Q : |N_{G[M]}(v) \cap \text{undecided}(Q, G^\star[M])| \geq \frac{1}{p \cdot \gamma}\right\}.$$

Next we prove that in expectation, SEED($G^\star$) contains a small number of vertices that are not in MVC($G^\star$). To do this we first prove the following bound on the size of A.

Claim 3.5. *We have $\mathbb{E}[|A \setminus \text{MVC}(G^\star)|] \leq \varepsilon \cdot$ opt.*

Proof. Consider any vertex $v \in M \setminus Q$. Upon sampling $G^\star \sim G_p$, observe that if $v \in A \setminus \text{MVC}(G^\star)$, then $v \in \text{revealing}(Q, G^\star[M])$. By definition of the termination of the Vertex-Seed algorithm, we therefore have that

$$\Pr_{G^\star}[v \in A \setminus \text{MVC}(G^\star)] \leq \Pr_{G^\star}[v \in \text{revealing}(Q, G^\star[M])] < \delta.$$

Finally, by the bound on M in Claim 3.3, this implies

$$\mathbb{E}[|A \setminus \mathrm{MVC}(G^\star)|] < \delta \cdot |M| \le \varepsilon \cdot \mathrm{opt}. \quad \square$$

We can now bound the expected size of the set $\mathrm{SEED}(G^\star)$.

Lemma 3.6. *We have*

$$\mathbb{E}[|\mathrm{SEED}(G^\star) \setminus \mathrm{MVC}(G^\star)|] \le O(\varepsilon) \cdot \mathbb{E}[\mathrm{MVC}(G^\star)]$$

for any $\varepsilon < 1/4$.

Proof. Note first, that $\mathrm{decided}(Q, G^\star[M]) \subseteq \mathrm{MVC}(G^\star)$, since by the definition of decided, for each vertex $v \in \mathrm{decided}(Q, G^\star[M])$, there is a vertex $v_i \notin \mathrm{MVC}(G^\star)$ that is adjacent to v in the realization $G^\star$. Thus,

$$\mathbb{E}[|\mathrm{SEED}(G^\star) \setminus \mathrm{MVC}(G^\star)|] = \mathbb{E}[|L \setminus \mathrm{MVC}(G^\star)|] + \mathbb{E}[|Q \setminus \mathrm{MVC}(G^\star)|] + \mathbb{E}[|A \setminus \mathrm{MVC}(G^\star)|].$$

By Claim 3.2, 3.4, and 3.5, we have

$$\begin{aligned}\mathbb{E}[|\mathrm{SEED}(G^\star) \setminus \mathrm{MVC}(G^\star)|] &\le 4\varepsilon \cdot \mathrm{opt} + (\varepsilon^2/100) \cdot \mathrm{opt} + \varepsilon \cdot \mathrm{opt} \\ &\le O(\varepsilon) \cdot \mathbb{E}[|\mathrm{MVC}(G^\star)|]. \quad \square\end{aligned}$$

3.4 Using Vertex-Seed to Analyze Vertex-Cover

In this section we prove our main result about the performance of Vertex-Cover algorithm, which we formally state in the next theorem.

Theorem 3.7. *The output of the* Vertex-Cover *algorithm has an expected size of at most* $(1 + O(\varepsilon)) \cdot \mathrm{opt}$.

To analyze the Vertex-Cover algorithm, we first define an auxiliary problem (Problem (6)), which strengthens the constraints of Problem (2) by requiring that the solution S includes the vertices in Q, where Q is the deterministic output of the Vertex-Seed algorithm on $G[M]$. For the remaining of the section we will use $\hat{P}$ to denote the optimal solution to Problem (6).

$$\begin{aligned}\min_{P \subseteq V} \quad & |P| \;+\; \mathbb{E}\Big[\,\big|\mathrm{MVC}\big(G^\star[V \setminus P]\big)\big|\,\Big] \\ \text{s.t.} \quad & Q \subseteq P, \\ & G[V \setminus P] \;\text{ has at most }\; 2 \cdot \frac{10^3\, n}{\varepsilon^5 p} \;\text{ edges.}\end{aligned} \tag{6}$$

Our analysis compares the cost of $\hat{P}$ to the expected cost of a strategy that adaptively picks $S = \mathrm{SEED}(G^\star)$ for every realization $G^\star$. We begin by proving that for any fixed realization $G^\star$, the set $\mathrm{SEED}(G^\star)$ is a feasible solution to (6) as $Q \subseteq \mathrm{SEED}(G^\star)$, by definition, and by showing that the induced graph $G[V \setminus \mathrm{SEED}(G^\star)]$ is sparse (Lemma 3.8). Moreover, Lemma 3.6 implies that

$$\mathbb{E}[|\mathrm{SEED}(G^\star)| + |\mathrm{MVC}\big(G^\star[V \setminus \mathrm{SEED}(G^\star)]\big)|] = (1 + O(\varepsilon)) \cdot \mathrm{opt},$$

therefore adaptively picking $\mathrm{SEED}(G^\star)$ as a solution for each $G^\star$, would achieve an expected objective value of $(1 + O(\varepsilon)) \cdot \mathrm{opt}$ in Problem (6). Our goal is to prove that the optimal static solution $\hat{P}$ performs essentially as well.

For any realization $G^\star$, the set $\mathrm{SEED}(G^\star)$ depends only on two quantities: (a) on the realization of edges incident to Q and (b) on the intersection of Q with $\mathrm{MVC}(G^\star)$. Let F be the set of edges, of the base graph G, which have at least one endpoint in Q and let also $F^\star \subseteq F$ denote a fixed realization of them. For convenience, we slightly abuse notation and index each possible SEED set by the tuple $(Q_{VC}, F^\star)$ that determines it. For the sake of completeness, we re-state the formal definitions of decided, undecided and SEED in this new notation. For a fixed $F^\star \subseteq F$ and a fixed $Q_{VC} \subseteq Q$, we have

- $\mathrm{decided}(Q_{VC}, F^\star) = \{u \in M \setminus Q : \exists (u, v) \in F^\star \text{ s.t. } v \in (Q \setminus Q_{VC})\}$
- $\mathrm{undecided}(Q_{VC}, F^\star) = M \setminus (Q \cup \mathrm{decided}(Q_{VC}, F^\star))$
- $A(Q_{VC}, F^\star) = \{u \in M \setminus Q : \; |N_{G[M]}(u) \cap \mathrm{undecided}(Q_{VC}, F^\star)| \ge 1/(p \cdot \gamma)\}$
- $\mathrm{SEED}(Q_{VC}, F^\star) = L \cup Q \cup \mathrm{decided}(F^\star, Q_{VC}) \cup A(Q_{VC}, F^\star)$.

Notice that for any realization $G^\star$, for which $F^\star$ is the realization of the edges in F, we have that $\mathrm{decided}(Q \cap \mathrm{MVC}(G^\star), F^\star) = \mathrm{decided}(Q, G^\star[M])$ and $\mathrm{SEED}(G^\star) = \mathrm{SEED}(Q \cap \mathrm{MVC}(G^\star), F^\star)$.

A key observation is that the objective of Problem 6 is independent of how the edges in F realize, because every feasible solution S contains the set Q, and thus the edges in F are not part of the induced graph $G[V \setminus S]$. We can therefore fix a realization $F^\star$ of these edges and compare $\hat{P}$ and $\mathrm{SEED}(G^\star)$ over the randomness outside of F. For a fixed realization $F^\star$, SEED only depends on Q_{VC}, which ranges over subsets of Q and can, thus, take at most $2^{|Q|}$ values.

By employing the concentration of the minimum vertex cover (Theorem 1.2) and taking a union bound over the $2^{|Q|}$ possible values of SEED, we are able to show that the value of SEED that has the minimum expected cost (for the fixed $F^\star$ and over the randomness outside of F) is almost as good as adaptively selecting $\mathrm{SEED}(G^\star)$. Since each value of SEED is feasible for Problem 6 and the objective only depends on the randomness outside F, we can conclude that $\hat{P}$ cannot be worse than the best fixed SEED.

Finally, since the above holds for any fixed $F^\star$, we conclude the proof taking the expectation over the randomness in F and showing that the expected cost of $\hat{P}$ almost as good as the expected cost of $\mathrm{SEED}(G^\star)$.

We will now formalize the proof outlined above. First, we begin by proving that for any $F^\star \subseteq F$ and any $Q_{VC} \subseteq Q$, the induced graph $G[V \setminus \mathrm{SEED}(Q_{VC}, F^\star)]$ is sparse. This proves that $\mathrm{SEED}(Q_{VC}, F^\star)$ is a feasible solution to Problem (6).

Lemma 3.8. *For any* $F^\star \subseteq F$ *and any* $Q_{VC} \subseteq Q$, *the graph* $G[V \setminus \mathrm{SEED}(Q_{VC}, F^\star)]$ *has at most* $\frac{2 \cdot 10^3 \cdot n}{p\varepsilon^5}$ *many edges.*

Proof of Lemma 3.8. Since $\mathrm{SEED}(Q_{VC}, F^\star)$ contains L, all vertices in $V \setminus \mathrm{SEED}(Q_{VC}, F^\star)$ lie in $(M \cup S) \setminus \mathrm{SEED}(Q_{VC}, F^\star)$. Moreover, since $\mathrm{SEED}(Q_{VC}, F^\star)$ also contains $\mathrm{decided}(Q_{VC}, F^\star)$,

$$M \setminus \mathrm{SEED}(Q_{VC}, F^\star) \;\subseteq\; \mathrm{undecided}(Q_{VC}, F^\star).$$

Now, every vertex $u \in M \setminus \mathrm{SEED}(Q_{VC}, F^\star)$ has at most $1/(p \cdot \gamma)$ neighbors inside $\mathrm{undecided}(Q_{VC}, F^\star)$ (since otherwise, such a vertex would be included in $A(Q_{VC}, F^\star)$). Hence the number of edges in $G[V \setminus \mathrm{SEED}(Q_{VC}, F^\star)]$ that have both endpoints in M is at most

$$\frac{|M|}{p\gamma} \;\le\; \frac{10^3\, n}{p\varepsilon^5},$$

since $\gamma = \varepsilon^5/10^3$.

The remaining edges in $G[V \setminus \mathrm{SEED}(Q_{VC}, F^\star)]$, are either between S and M or entirely within S. Fix any such edge $e = (u, v)$

with $u \in S$ and $v \in (M \cup S) \setminus \mathrm{SEED}(Q_{VC}, F^\star)$, and let $\mathrm{MVC}(G^\star)$ be the minimum vertex cover of a realization $G^\star$. Let

$$\begin{aligned} c_e &:= \Pr[e \text{ is covered by } \mathrm{MVC}(G^\star)] \\ &\le \Pr[u \in \mathrm{MVC}(G^\star)] + \Pr[v \in \mathrm{MVC}(G^\star)] \\ &\le \varepsilon + 1 - 2\varepsilon = 1 - \varepsilon. \end{aligned}$$

As proven in [16], there can be at most $O(n/(\varepsilon \cdot p))$ such edges (u, v) in G. We also provide a concise proof here for completeness. Let $W = \{e \in E : c_e \le 1 - \varepsilon\}$, and let

$$X = |\{e \in E : e \text{ is } \textit{not} \text{ covered by } \mathrm{MVC}(G^\star)\}|.$$

By linearity of expectation,

$$\mathbb{E}[X] \;=\; \sum_{e \in E} (1 - c_e) \;\ge\; \sum_{e \in W} (1 - c_e) \;\ge\; \varepsilon\,|W|. \tag{7}$$

To upper bound $\mathbb{E}[X]$, observe that every uncovered edge under $\mathrm{MVC}(G^\star)$ must be absent from $G^\star$ (otherwise $\mathrm{MVC}(G^\star)$ would not be a vertex cover of $G^\star$). The probability that a vertex-induced subgraph of G has more than n/p unrealized edges is at most $(1-p)^{n/p} \le e^{-n}$. Also, there are at most 2^n choices for the vertex induced subgraph $H = G[V \setminus \mathrm{MVC}(G^\star)]$. Therefore,

$$\Pr[H \text{ has more than } n/p \text{ edges}] \le (2/e)^n .$$

Using the above, we can upper bound $\mathbb{E}[X]$ as

$$\mathbb{E}[X] \;\le\; \frac{n}{p} + n^2 \left(\frac{2}{e}\right)^n \le \frac{n}{p} + 6. \tag{8}$$

Combining (7) and (8) yields the deterministic bound

$$|W| \;\le\; \frac{\mathbb{E}[X]}{\varepsilon} \;\le\; \frac{n}{p\,\varepsilon} + \frac{6}{\varepsilon} \le 2\,\frac{n}{p\,\varepsilon} \le 2\,\frac{n}{p\varepsilon^5}.$$

Therefore, the total remaining edges are at most $2 \cdot 10^3\ n/(p\varepsilon^5)$ which concludes the proof of the lemma. □

For convenience, we define $g(A)$, for any set $A \subseteq V$, to be the random variable corresponding to the size of a vertex cover of $G^\star$ that includes the set A and covers the induced graph $G^\star[V \setminus A]$ optimally, i.e.,

$$g(A) = |A| + |\mathrm{MVC}(G^\star[V \setminus A])|.$$

Notice that the objective values of the optimization problems (2) and (6) are equal to $\mathbb{E}[g(S)]$. Before we continue with the analysis, we will prove the following tail bound for $g(S)$, by using the concentration of the minimum vertex cover (Theorem 1.2).

Lemma 3.9. *For any set $S \subseteq V$ and any $\varepsilon \in [0, 1]$, the following holds*

$$\Pr\left[\ |g(S) - \mathbb{E}[g(S)]| > \varepsilon\,\mathbb{E}[g(S)]\ \right] \le 2\,e^{-\varepsilon^2\,\mathrm{opt}/66}.$$

Proof of Lemma 3.9. Let $X := |\mathrm{MVC}(G^\star[V \setminus S])|$. Since $|S|$ is deterministic,

$$\begin{aligned} &\Pr[\ |g(S) - \mathbb{E}[g(S)]| > \varepsilon\,\mathbb{E}[g(S)]\ ] \\ &\qquad = \Pr[\ |X - \mathbb{E}[X]| > \varepsilon\,\mathbb{E}[X] + \varepsilon\,|S|\ ]. \end{aligned} \tag{$*$}$$

Case (a): $\mathbb{E}[X] \ge \mathrm{opt}/2$. From ($*$),

$$\begin{aligned} \Pr[\ |X - \mathbb{E}[X]| > \varepsilon\,\mathbb{E}[X] + \varepsilon\,|S|\ ] &\le \Pr[\ |X - \mathbb{E}[X]| > \varepsilon\,\mathbb{E}[X]\ ] \\ &\le 2e^{-\varepsilon^2\,\mathbb{E}[X]/33} \;\le\; e^{-\varepsilon^2\,\mathrm{opt}/66}, \end{aligned}$$

where we used the tail bound of the minimum vertex cover (Theorem 1.2) for X and the fact that $\mathbb{E}[X] \ge \mathrm{opt}/2$.

Case (b): $\mathbb{E}[X] \le \mathrm{opt}/2$.

Since $g(S) = |S| + X$ and $S \cup \mathrm{MVC}(G^\star[V \setminus S])$ is a vertex cover of $G^\star$, we have $g(S) \ge |\mathrm{MVC}(G^\star)|$ for every realization, hence $\mathbb{E}[g(S)] \ge \mathrm{opt}$ and thus $|S| + \mathbb{E}[X] = \mathbb{E}[g(S)] \ge \mathrm{opt}$. Therefore, from ($*$),

$$\begin{aligned} \Pr[\ |X - \mathbb{E}[X]| > \varepsilon\,\mathbb{E}[X] + \varepsilon\,|S|\ ] &\le \Pr[\ |X - \mathbb{E}[X]| > \varepsilon\,\mathrm{opt}\ ] \\ &= \Pr\left[\ |X - \mathbb{E}[X]| > \tfrac{\varepsilon\,\mathrm{opt}}{\mathbb{E}[X]} \cdot \mathbb{E}[X]\ \right] \\ &\le 2e^{-\varepsilon^2\,\mathrm{opt}^2/(33 \cdot \mathbb{E}[X])} \\ &\le 2e^{-\varepsilon^2\,\mathrm{opt}/66}, \end{aligned}$$

where we again used concentration for X and the bound $\mathbb{E}[X] \le \mathrm{opt}/2$.

Combining the two cases yields the stated inequality. □

We are now ready to prove that, for any fixed realization $F^\star$, $\hat{P}$ is almost as good as adaptively selecting the best set among $\{\mathrm{SEED}(Q_{VC}, F^\star) \;:\; Q_{VC} \subseteq Q\}$. We state this formally in the next lemma.

Lemma 3.10. *Let $\hat{P}$ be an optimal solution to (6). Then, for any realization $F^\star$,*

$$\mathbb{E}_{G^\star \setminus F^\star}\left[g\left(\hat{P}\right)\right] \;\le\; (1+O(\varepsilon))\ \mathbb{E}_{G^\star \setminus F^\star}\left[\min_{Q_{VC} \subseteq Q} g(\mathrm{SEED}(Q_{VC}, F^\star))\right].$$

Proof of Lemma 3.10. As we previously discussed, because $Q \subseteq S$, edges in F never appear in $G[V \setminus S]$, therefore the objective of (6) depends only on randomness outside F and equals $\mathbb{E}_{G^\star \setminus F^\star}[g(S)]$. For any fixed $F^\star$ and any $Q_{VC} \subseteq Q$, the set $S = \mathrm{SEED}(Q_{VC}, F^\star)$ is feasible for (6): (a) it contains Q by definition, and (b) by Lemma 3.8, the graph $G[V \setminus \mathrm{SEED}(Q_{VC}, F^\star)]$ has $O(n/(\varepsilon^5 p))$ edges. Hence, by the optimality of $\hat{P}$,

$$\mathbb{E}_{G^\star \setminus F^\star}\left[g\left(\hat{P}\right)\right] \;\le\; \min_{Q_{VC} \subseteq Q}\ \mathbb{E}_{G^\star \setminus F^\star}\left[g(\mathrm{SEED}(Q_{VC}, F^\star))\right].$$

It remains to relate the expected cost of the best SEED to the expected cost of adaptively selecting the best SEED for every realization, i.e. the minimum expectation and the expected minimum. Define

$$\mu := \min_{Q_{VC} \subseteq Q}\ \mathbb{E}_{G^\star \setminus F^\star}\left[g(\mathrm{SEED}(Q_{VC}, F^\star))\right].$$

By a union bound and Lemma 3.9 applied to each fixed Q_{VC},

$$\Pr\left[\min_{Q_{VC} \subseteq Q} g(\mathrm{SEED}(Q_{VC}, F^\star)) < (1-\varepsilon)\mu\right] \;\le\; 2^{|Q|+1}\,e^{-\varepsilon^2\,\mathrm{opt}/66}.$$

Together with Claim 3.4 (which gives $|Q| \le \varepsilon^2 \mathrm{opt}/100$), we obtain

$$\Pr\left[\min_{Q_{VC} \subseteq Q} g(\mathrm{SEED}(Q_{VC}, F^\star)) < (1-\varepsilon)\mu\right] \;\le\; 2\left(\frac{2}{e}\right)^{\varepsilon^2 \mathrm{opt}/100}.$$

Therefore,

$$\begin{aligned}
&\mathbb{E}_{G^\star\setminus F^\star}\left[\min_{Q_{VC}\subseteq Q} g\big(\mathrm{SEED}(Q_{VC},F^\star)\big)\right]\\
&\geq \Pr\left[\min_{Q_{VC}\subseteq Q} g\big(\mathrm{SEED}(Q_{VC},F^\star)\big)\geq (1-\varepsilon)\,\mu\right](1-\varepsilon)\,\mu\\
&\geq \left(1-2\left(\tfrac{2}{e}\right)^{\varepsilon^2\mathrm{opt}/100}\right)(1-\varepsilon)\,\mu,
\end{aligned}$$

which rearranges to

$$\begin{aligned}
&\min_{Q_{VC}\subseteq Q}\ \mathbb{E}_{G^\star\setminus F^\star}\left[g\big(\mathrm{SEED}(Q_{VC},F^\star)\big)\right]\\
&\leq \frac{1}{\left(1-2\,(2/e)^{\varepsilon^2\mathrm{opt}/100}\right)(1-\varepsilon)}\,\mathbb{E}_{G^\star\setminus F^\star}\left[\min_{Q_{VC}\subseteq Q} g\big(\mathrm{SEED}(Q_{VC},F^\star)\big)\right].
\end{aligned}$$

Using $\mathrm{opt}\geq C\log(1/\varepsilon)/\varepsilon^2$ to absorb the $2\cdot(2/e)^{\varepsilon^2\mathrm{opt}/2}$ term into $O(\varepsilon)$ yields the claimed $(1+O(\varepsilon))$ factor. As we have discussed, in the case of $\mathrm{opt}=O(\log(1/\varepsilon)/\varepsilon^2)$ the problem becomes trivial, in the sense that it is feasible to query the whole base graph G. □

Finally, averaging the above result over the randomness of $F^\star$ yields our final bound.

Lemma 3.11. *Let $\hat{P}$ be an optimal solution to (6). Then*

$$\mathbb{E}\left[g\left(\hat{P}\right)\right]\ \leq\ (1+O(\varepsilon))\cdot\mathrm{opt}.$$

Proof of Lemma 3.11. From the law of total expectation, with respect to the edges F, we have that

$$\mathbb{E}\left[g\left(\hat{P}\right)\right]\ =\ \mathbb{E}_{F^\star}\left[\mathbb{E}_{G^\star\setminus F^\star}\left[g\left(\hat{P}\right)\right]\right].$$

By averaging the result of Lemma 3.10 over the randomness in $F^\star$, we get

$$\mathbb{E}\left[g\left(\hat{P}\right)\right]\ \leq\ (1+O(\varepsilon))\cdot\mathbb{E}_{F^\star}\left[\mathbb{E}_{G^\star\setminus F^\star}\left[\min_{Q_{VC}\subseteq Q} g\big(\mathrm{SEED}(Q_{VC},F^\star)\big)\right]\right].$$

Observe that the expectation in the right hand side above, is simply taken over the all randomness in $G^\star$. We can upper bound this term by picking $F^\star$ to be the realization of edges in F and $Q_{VC}=Q\cap\mathrm{MVC}(G^\star)$, which yields

$$\begin{aligned}
&\mathbb{E}\left[\min_{Q_{VC}\subseteq Q} g\big(\mathrm{SEED}(Q_{VC},F^\star)\big)\right]\ \leq\ \mathbb{E}\left[g\big(\mathrm{SEED}(G^\star)\big)\right]\\
&=\ \mathbb{E}\left[\left|\mathrm{SEED}(G^\star)\cup\mathrm{MVC}\big(G^\star[V\setminus\mathrm{SEED}(G^\star)]\big)\right|\right]\\
&\leq\ \mathbb{E}\left[\left|\mathrm{SEED}(G^\star)\cup\mathrm{MVC}(G^\star)\right|\right].
\end{aligned}$$

The proof is concluded by employing Lemma 3.6,

$$\begin{aligned}
&\mathbb{E}\left[\left|\mathrm{SEED}(G^\star)\cup\mathrm{MVC}(G^\star)\right|\right]\\
&\leq\ \mathbb{E}\left[\left|\mathrm{MVC}(G^\star)\right|\right]\ +\ \mathbb{E}\left[\left|\mathrm{SEED}(G^\star)\setminus\mathrm{MVC}(G^\star)\right|\right]\\
&\leq\ (1+O(\varepsilon))\cdot\mathbb{E}\left[\left|\mathrm{MVC}(G^\star)\right|\right].
\end{aligned}$$

□

Finally, Theorem 3.7 follows from Lemma 3.11 and the fact that Problem (6) is a more constrained version of Problem (2).

Acknowledgments

The authors thank the anonymous reviewer for pointing out the alternative approach based on Talagrand's inequality for proving the concentration bound for the size of a minimum vertex cover.

This project started at Dagstuhl Seminar 25181, "Learned Predictions for Data Structures and Running Time".

Miltiadis Stouras and Ola Svensson are supported by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number MB22.00054. Jan van den Brand is supported by NSF Award CCF-2338816 and CCF-2504994. Inge Li Gørtz is supported by Danish Research Council grant DFF-8021-002498. Chirag Pabbaraju is supported by Gregory Valiant's and Moses Charikar's Simons Investigator Awards, and a Google PhD Fellowship. Cliff Stein is supported in part by NSF grant CCF-2218677, ONR grant ONR-13533312, and by the Wai T. Chang Chair in Industrial Engineering and Operations Research. Debmalya Panigrahi is supported in part by NSF grants CCF-2329230 and CCF-1955703.

References

[1] Arash Asadpour, Hamid Nazerzadeh, and Amin Saberi. 2008. Stochastic submodular maximization. In *International Workshop on Internet and Network Economics*. Springer, 477–489. doi:10.1007/978-3-540-92185-1_53

[2] Sepehr Assadi and Aaron Bernstein. 2019. Towards a Unified Theory of Sparsification for Matching Problems. In *2nd Symposium on Simplicity in Algorithms (SOSA 2019)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 11–1. doi:10.4230/OASIcs.SOSA.2019.11

[3] Sepehr Assadi, Sanjeev Khanna, and Yang Li. 2016. The Stochastic Matching Problem with (Very) Few Queries. In *Proceedings of the 2016 ACM Conference on Economics and Computation*. 43–60. doi:10.1145/2940716.2940769

[4] Sepehr Assadi, Sanjeev Khanna, and Yang Li. 2017. The stochastic matching problem: Beating half with a non-adaptive algorithm. In *Proceedings of the 2017 ACM Conference on Economics and Computation*. 99–116. doi:10.1145/3033274.3085146

[5] Amir Azarmehr, Soheil Behnezhad, Alma Ghafari, and Ronitt Rubinfeld. 2025. Stochastic matching via in-n-out local computation algorithms. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*. 1055–1066. doi:10.1145/3717823.3718279

[6] Soheil Behnezhad, Avrim Blum, and Mahsa Derakhshan. 2022. Stochastic vertex cover with few queries. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 1808–1846. doi:10.1137/1.9781611977073.73

[7] Soheil Behnezhad and Mahsa Derakhshan. 2020. Stochastic Weighted Matching:(1-ϵ) Approximation. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 1392–1403. doi:10.48550/arXiv.2004.08703

[8] Soheil Behnezhad, Mahsa Derakhshan, and MohammadTaghi Hajiaghayi. 2020. Stochastic matching with few queries:(1-ε) approximation. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*. 1111–1124. doi:10.48550/arXiv.2002.11880

[9] Soheil Behnezhad, Alireza Farhadi, MohammadTaghi Hajiaghayi, and Nima Reyhani. 2019. Stochastic matching with few queries: New algorithms and tools. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2855–2874. doi:10.48550/arXiv.1811.03224

[10] Soheil Behnezhad and Nima Reyhani. 2018. Almost optimal stochastic weighted matching with few queries. In *Proceedings of the 2018 ACM Conference on Economics and Computation*. 235–249. doi:10.1145/3219166.3219226

[11] Dimitri P Bertsekas and John N Tsitsiklis. 1991. An analysis of stochastic shortest path problems. *Mathematics of Operations Research* 16, 3 (1991), 580–595. doi:10.1287/moor.16.3.580

[12] Anand Bhalgat, Ashish Goel, and Sanjeev Khanna. 2011. Improved approximation results for stochastic knapsack problems. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, 1647–1665. doi:10.1137/1.9781611973082.127

[13] Avrim Blum, John P Dickerson, Nika Haghtalab, Ariel D Procaccia, Tuomas Sandholm, and Ankit Sharma. 2015. Ignorance is almost bliss: Near-optimal stochastic matching with few queries. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*. 325–342. doi:10.1145/2764468.2764479

[14] Avrim Blum, Anupam Gupta, Ariel Procaccia, and Ankit Sharma. 2013. Harnessing the power of two crossmatches. In *Proceedings of the fourteenth ACM conference on Electronic commerce*. 123–140. doi:10.1145/2482540.2482569

[15] Brian C Dean, Michel X Goemans, and Jan Vondrák. 2008. Approximating the stochastic knapsack problem: The benefit of adaptivity. *Mathematics of Operations*

Research 33, 4 (2008), 945–964. doi:10.1287/moor.1080.0330

[16] Mahsa Derakhshan, Naveen Durvasula, and Nika Haghtalab. 2023. Stochastic minimum vertex cover in general graphs: A 3/2-approximation. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*. 242–253. doi:10.1145/3564246.3585230

[17] Mahsa Derakhshan and Mohammad Saneian. 2025. Query Efficient Weighted Stochastic Matching. In *52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 67–1. doi:10.4230/LIPIcs.ICALP.2025.67

[18] Mahsa Derakhshan, Mohammad Saneian, and Zhiyang Xun. 2025. Query complexity of stochastic minimum vertex cover. In *16th Innovations in Theoretical Computer Science Conference (ITCS 2025)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 41–1. doi:10.4230/LIPIcs.ITCS.2025.41

[19] Michel X Goemans and Jan Vondrák. 2006. Covering minimum spanning trees of random subgraphs. *Random Structures & Algorithms* 29, 3 (2006), 257–276. doi:10.1002/rsa.20115

[20] Daniel Golovin and Andreas Krause. 2011. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research* 42 (2011), 427–486. https://www.jair.org/index.php/jair/article/view/10731

[21] Jan Vondrák. 2007. Shortest-path metric approximation for random subgraphs. *Random Structures & Algorithms* 30, 1-2 (2007), 95–104. doi:10.1002/rsa.20150

[22] Yutaro Yamaguchi and Takanori Maehara. 2018. Stochastic packing integer programs with few queries. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 293–310. doi:10.1007/s10107-019-01388-x

Received 2025-11-04; accepted 2026-02-01