

Lecture 3

September 22, 2020

Instructor: Sepehr Assadi

Scribe: Sepehr Assadi

Disclaimer: These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.

In this lecture, we will primarily focus on the following paper:

- “Sepehr Assadi, Yu Chen, Sanjeev Khanna, Sublinear Algorithms for $(\Delta + 1)$ Vertex Coloring. In SODA 2019.”

1 The $(\Delta + 1)$ Vertex Coloring Problem

A proper c -coloring of an undirected graph is any assignment of colors from the palette $\{1, 2, \dots, c\}$ to the vertices of the graph such that no two adjacent vertices receive the same color. Graph coloring is another highly fundamental problem in TCS and graph theory with a wide range of applications. One of the most basic and applicable forms of graph coloring problems is $(\Delta + 1)$ coloring of graphs with maximum degree Δ as every graph admits such a coloring¹: we can color the vertices greedily and by the pigeonhole principle we never run out of color for each vertex as number of its neighbors is less than the available colors².

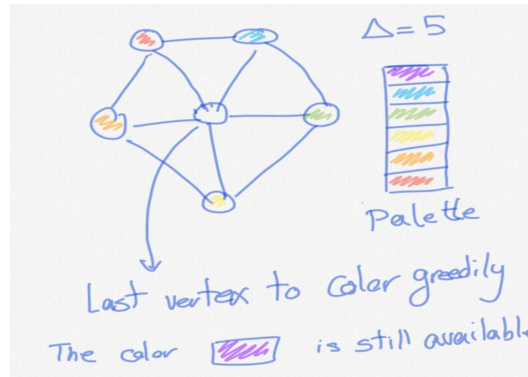


Figure 1: An illustration of the greedy $(\Delta + 1)$ coloring algorithm: By the time we get to color a vertex v , there is still at least one color available to v .

In this lecture, we study the $(\Delta + 1)$ coloring problem in single-pass streams. Despite its utter simplicity, the greedy algorithm for $(\Delta + 1)$ coloring is not suitable for streaming implementation as it requires visiting all neighbors of each vertex at the same time. As a result, $(\Delta + 1)$ coloring appeared to be more or less a “hard” problem that cannot be solved via single-pass semi-streaming algorithm. However, in 2019, Assadi, Chen, and Khanna [2] proved that this problem in fact admits a single-pass semi-streaming algorithm. Formally,

Theorem 1 ([2]). *There is a randomized semi-streaming algorithm that makes a single pass over the edges of any graph $G = (V, E)$ with maximum degree Δ and with high probability, outputs a $(\Delta + 1)$ coloring of G . Moreover, the algorithm does not need the prior knowledge of Δ , runs in polynomial time, and never outputs a wrong coloring of G (rather, outputs FAIL when it cannot find a proper coloring).*

¹And this is tight for cliques and odd-cycles and in fact only for these graphs: by Brook’s theorem, any connected graph other than a clique or an odd-cycle admits a Δ -coloring.

²In fact, $(\Delta + 1)$ coloring has an “ultra greedy” property: any partial solution, i.e., a partial coloring of vertices, can be extended to a proper coloring without modification.

We will prove a slightly simpler version of this theorem in this lecture: the algorithm we design will have exponential time and also assumes the prior knowledge of Δ .³

Notation and Preliminaries. For any integer $t \geq 1$, we use $[t] := \{1, \dots, t\}$. As before, for any graph $G = (V, E)$ and vertex $v \in V$, $N(v)$ denotes the neighborset of v and $\deg(v)$ is the degree of v . We will use the following variant of Chernoff bound.

Proposition 2 (Chernoff bound). Suppose X_1, \dots, X_n are independent random variables in $[0, 1]$ and $X = \sum_i X_i$. Then, for any $t \geq 1$,

$$\Pr(|X - \mathbb{E}[X]| \geq t \cdot \mathbb{E}[X]) \leq 2 \cdot \exp\left(-\frac{t \cdot \mathbb{E}[X]}{3}\right).$$

2 Warm Up: A Semi-Streaming (2Δ) -Coloring Algorithm

We start by presenting a very simple semi-streaming algorithm for (2Δ) -coloring (which is an algorithmically easier problem than $(\Delta + 1)$ coloring). The algorithm simply samples $O(\log n)$ “potential” colors for each vertex at the beginning of the stream and throughout the stream store any edge that can ever become monochromatic under these potential colors; at the end, it tries to find a coloring of the vertices greedily using these potential colors. Formally,

Algorithm 1. A single-pass semi-streaming algorithm for (2Δ) -coloring.

- (i) For every vertex $v \in V$, sample a set $L(v)$ of $s = (4 \log n)$ colors from $\{1, \dots, 2\Delta\}$ uniformly at random and independently.
- (ii) During the stream, store any edge $e = (u, v)$ in a subgraph H if $L(u) \cap L(v) \neq \emptyset$.
- (iii) At the end of the stream:
 - (a) Go over vertices v in V in some arbitrary order;
 - (b) For any vertex v , if $L(v) = \emptyset$ output FAIL; otherwise, color v from any color c in $L(v)$ and remove c from $L(u)$ for any neighbor u of v in H .

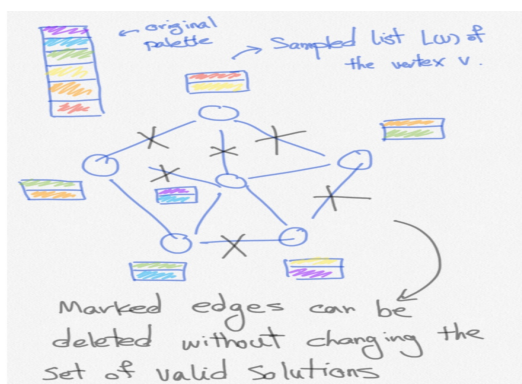


Figure 2: An illustration of Algorithm 1: Many of the edges of the graph can be discarded without the risk of outputting a wrong coloring.

³The latter assumption is quite easy to remove but removing the former one requires some work. However, it was shown in [2] that the entire algorithm can be made to work even in $\tilde{O}(n^{3/2})$ time; for moderately dense graphs, this is even faster than reading the entire input once—in other words, there is an algorithm faster than the textbook greedy algorithm for this problem!

We first prove that the space complexity of [Algorithm 1](#) is $\tilde{O}(n)$ with high probability. The space of this algorithm is $O(n \cdot s) = O(n \log n)$ for storing the colors plus the size of H which we bound in the following.

Lemma 3. *The number of edges in the subgraph H of [Algorithm 1](#) is $O(n \log^2(n))$ with high probability.*

Proof. We prove that the maximum degree of H is $O(\log^2(n))$ with high probability which implies the lemma immediately. Fix any vertex $v \in V$ and its list of colors $L(v)$ and for any neighbor u of v in G , let $X_u \in \{0, 1\}$ be an indicator random variable which is 1 iff v is neighbor to u in H . For u to be neighbor of v , $L(u) \cap L(v)$ should be non-empty. Thus,

$$\Pr(X_u = 1) \leq \sum_{c \in L(v)} \Pr(c \in L(u)) = \frac{s^2}{2\Delta},$$

where the first inequality is by union bound and the equality is because $|L(v)| = |L(u)| = s$ and conditioned on $L(v)$, the set $L(u)$ is still a random s -subset of $\{1, \dots, 2\Delta\}$. Moreover, $\deg_H(v) = \sum_{u \in N_G(v)} X_u$ and thus

$$\mathbb{E}[\deg_H(v)] \leq \Delta \cdot \frac{s^2}{2\Delta} = 8 \log^2(n).$$

As $\deg_H(v)$ is sum of independent⁴ 0/1-random variables X_u for $u \in N_G(v)$, we can apply Chernoff bound ([Proposition 2](#)) and obtain that

$$\Pr(\deg_H(v) \geq 16 \log^2(n)) \leq 2 \cdot \exp\left(-\frac{16 \log^2(n)}{3}\right) < \frac{1}{n^5}.$$

A union bound over all n vertices now finalizes the proof. \square

We now prove the correctness of [Algorithm 1](#). An immediate observation is that this algorithm never outputs a wrong coloring of G . More formally, any proper coloring of H such that color of each vertex v is chosen from $L(v)$ —this is called a *list-coloring* of H from lists $L(v)$ for $v \in V$ —is also a proper (2Δ) -coloring of G . This is simply because each $L(v) \subseteq \{1, \dots, 2\Delta\}$ and that any edge (u, v) of G that can ever be monochromatic under the list-coloring of H also belongs to H by definition. The above argument suggests that as long as the algorithm does not output FAIL, it will give a correct answer. We now bound the probability of it outputting FAIL, which finalizes the proof.

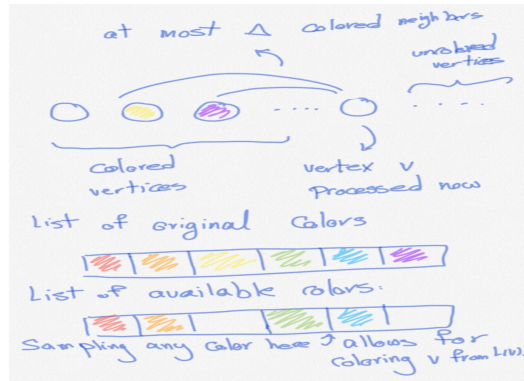


Figure 3: An illustration of the proof of [Lemma 4](#): By the time we get to process v , there are still $\geq \Delta$ colors whose inclusion in $L(v)$ allows for coloring v .

Lemma 4. *With high probability, [Algorithm 1](#) does not output FAIL.*

⁴ X_u 's are independent as we already conditioned on $L(v)$, and after that, $L(u)$'s for $u \in N_G(v)$ are chosen independently.

Proof. Suppose we are at the step of coloring a vertex $v \in V$ in the greedy algorithm at the end of [Algorithm 1](#). At this point, we condition on the choice of all lists $L(u)$ for vertices u which have already been colored and say that a color $c \in \{1, \dots, 2\Delta\}$ is *good*, if it is not used to color any neighbor of v . Clearly, if $L(v)$ contains any good color, we will be able to color v and no need to output FAIL at this point. By the independence of the choice of lists, we can think of $L(v)$ being sampled only now. As there are at most Δ neighbors for v ,

$$\Pr(L(v) \text{ contains no good color}) \leq \frac{\Delta^s}{2\Delta} = \left(\frac{1}{2}\right)^{4 \log n} = \frac{1}{n^4}.$$

A union bound over all $v \in V$ implies that the algorithm does not output FAIL with high probability. \square

Remark. The space complexity of [Algorithm 1](#) is bounded by $\tilde{O}(n)$ with high probability but not in the worst case; with some non-zero probability, it may even use $\Omega(n^2)$ space. Typically in the streaming model, we rather bound the worst-case space complexity of algorithms (similar to the communication cost of protocols in communication complexity). The transition however is standard: we can simply run the algorithm as it is and whenever it attempted to use a space beyond the bound of [Lemma 3](#) we just terminate the algorithm and output FAIL right away. This fixes the space complexity of the algorithm to a worst case bound, while increasing the error probability by a negligible factor.

3 The Semi-Streaming $(\Delta + 1)$ Coloring Algorithm

[Algorithm 1](#) suggests a natural approach for the $(\Delta + 1)$ coloring problem as well: what will happen if we sample the colors of each vertex from a set of $(\Delta + 1)$ colors instead of (2Δ) ones?

It is quite easy to verify that a simple extension of [Lemma 3](#) still holds to bound the space of this new algorithm. Alas, the last step of the algorithm for greedy list-coloring of H from lists $L(v)$ of $v \in V$ no longer works as before. For instance, suppose G was a $(\Delta + 1)$ -clique. By the time we get to color the last vertex of the clique, all of its Δ neighbors should have necessarily been colored with distinct colors (or the algorithm may have failed already). But since we only have $(\Delta + 1)$ colors to begin with, this leaves out just a single good color for v ; the probability that this color is chosen in $L(v)$ is then only $\Theta(\log n / \Delta)$ and thus with probability $1 - o(1)$ the algorithm has to output FAIL.

The $(\Delta + 1)$ -clique example on the surface suggest that [Algorithm 1](#) is doomed for $(\Delta + 1)$ coloring. However, there is still a silver lining in this example: as we prove in the lemma below, even though in this example H is no longer *greedily* list-colorable, it still admits proper list-coloring using a different argument.

Lemma 5. *Suppose for every vertex v of a $(\Delta + 1)$ -clique G , we sample a list $L(v)$ of colors of size $s = 8 \ln n$ independently and uniformly at random from the colors $\{1, \dots, \Delta + 1\}$. Then, with high probability, G is list-colorable from lists $L(v)$ of $v \in V$ (note that here $n = \Delta + 1$).*

Proof. As we saw, there is no hope of list-coloring G greedily in this lemma. We instead use a “global” argument that allows us to color all vertices at the same time by exploiting their dependencies more carefully.

Define the following bipartite graph $\mathcal{G} := (\mathcal{L}, \mathcal{R}, \mathcal{E})$ which we call the *palette graph* (see [Figure 4](#)):

1. The set of vertices on the left \mathcal{L} is all vertices of the $(\Delta + 1)$ -clique;
2. The set of vertices on the right \mathcal{R} is the colors $\{1, \dots, \Delta + 1\}$;
3. There is an edge between a vertex $v \in \mathcal{L}$ to a color $c \in \mathcal{R}$ iff $c \in L(v)$.

We claim that there is a list-coloring of G from lists L iff \mathcal{G} admits a perfect matching⁵: this is simply because there is a one-to-one mappings between list-colorings of G and perfect matchings in \mathcal{G} by considering any edge (v, c) as part of the matchings of \mathcal{G} as the assignment of color c from $L(v)$ to v in G .

⁵The reader may want to consult Lecture 2 for the definitions of perfect matchings and basic tools such as Hall’s theorem that will be used repeatedly in this lecture as well.

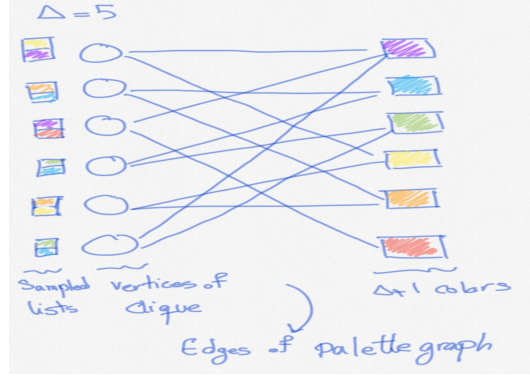


Figure 4: An illustration of the palette graph in Lemma 5: Finding a perfect matching in this palette graph allows us to color all vertices the same way in our clique.

It thus remains to prove that \mathcal{G} admits a perfect matching with high probability. Considering \mathcal{G} is a *random* graph with degree $\Theta(\log n)$, standard results in random graph theory already implies the existence of such a perfect matching [3]. For completeness and as a warm up, we prove this claim below.

Claim 6. *The bipartite graph $\mathcal{G} := (\mathcal{L}, \mathcal{R}, \mathcal{E})$ has a perfect matching with high probability.*

Proof. Recall that by Hall's theorem (see Lecture 2), for this bipartite graph to not have a perfect matching, there should exist a set S in \mathcal{L} , and T in \mathcal{R} , such that $|T| = |S| - 1$ and there are no edges between S and $\bar{T} = \mathcal{R} \setminus T$ (this is true iff $|N(S)| < |S|$ for some S which is the more familiar statement of Hall's theorem—note that as $|\mathcal{L}| = |\mathcal{R}|$, we can simply consider the sets in \mathcal{L} instead of both sides).

Let us now fix such a choice of $S \subseteq \mathcal{L}$ and $T \subseteq \mathcal{R}$ and see what is the probability of this event happening:

$$\Pr(N_{\mathcal{G}}(S) \subseteq T) \leq \left(\frac{|T|}{n}\right)^{|S| \cdot s} = \left(\frac{|S| - 1}{n}\right)^{|S| \cdot s} = \left(1 - \frac{n - |S| + 1}{n}\right)^{|S| \cdot s} \leq \exp\left(-\frac{(n - |S| + 1) \cdot |S|}{n} \cdot s\right).$$

We can bound the above quantity differently for $|S| \leq n/2$ and $|S| > n/2$ and obtain that:

$$\begin{aligned} \text{when } |S| \leq n/2: \quad & \Pr(N_{\mathcal{G}}(S) \subseteq T) \leq \exp(-|S| \cdot 4 \ln n); \\ \text{when } |S| > n/2: \quad & \Pr(N_{\mathcal{G}}(S) \subseteq T) \leq \exp(-(n - |S| + 1) \cdot 4 \ln n); \end{aligned}$$

Finally, we do a union bound over all choices of S, T , by grouping them based on the size of S :

$$\begin{aligned} \Pr(\mathcal{G} \text{ has no perfect matching}) &\leq \sum_{k=1}^n \sum_{\substack{S \subseteq \mathcal{L} \\ |S|=k}} \sum_{\substack{T \subseteq \mathcal{R} \\ |T|=k-1}} \Pr(N_{\mathcal{G}}(S) \subseteq T) \\ &\leq \sum_{k=1}^{n/2} \binom{n}{k}^2 \exp(-4k \cdot \ln n) + \sum_{k=n/2+1}^n \binom{n}{k-1}^2 \exp(-4(n-k+1) \cdot \ln n) \\ &\quad \text{(by monotonicity of } \binom{a}{b} \text{ for } b \leq a/2 \text{ and } b \geq a/2 \text{ individually)} \\ &= \sum_{k=1}^{n/2} \binom{n}{k}^2 \exp(-4k \cdot \ln n) + \sum_{k=n/2+1}^n \binom{n}{n-k+1}^2 \exp(-4(n-k+1) \cdot \ln n) \\ &\quad \text{(as } \binom{a}{b} = \binom{a}{a-b}) \\ &\leq \sum_{k=1}^{n/2} \exp(-2k \cdot \ln n) + \sum_{k=n/2+1}^n \exp(-2(n-k+1) \cdot \ln n) \quad \text{(as } \binom{a}{b} \leq a^b) \\ &= 2 \sum_{k=1}^{n/2} \frac{1}{n^{2k}} \leq \frac{3}{n^2}. \quad \square \text{ Claim 6} \end{aligned}$$

We are now done as we can color the vertices of G according to the perfect matching of \mathcal{G} and find a list-coloring of G from the lists $\{L(v) \mid v \in G\}$. \square

The natural question at this point is that whether [Lemma 5](#) and $(\Delta+1)$ -cliques were just lucky exceptions, or rather [Algorithm 1](#) can be made to work for any input graph, as long as we change the post-processing step of the algorithm to a more sophisticated approach beyond greedy list-coloring? The *palette sparsification theorem* of [\[2\]](#) proves that, perhaps surprisingly, the correct answer is the latter one.

Theorem 7 (Palette Sparsification Theorem [\[2\]](#)). *Let $G = (V, E)$ be any graph with n vertices and maximum degree Δ . Suppose for every vertex $v \in V$, we sample a list $L(v)$ of $\Theta(\log n)$ colors independently and uniformly at random from the colors $\{1, \dots, \Delta + 1\}$. Then, with high probability, G can be properly colored by coloring every $v \in V$ from the list $L(v)$; in other words, G is list-colorable from lists $\{L(v) \mid v \in V\}$.*

We will go over the proof of this theorem in the next section. For now, let us see the resulting semi-streaming algorithm for $(\Delta + 1)$ coloring.

Algorithm 2. A single-pass semi-streaming algorithm for $(\Delta + 1)$ -coloring.

- (i) For every vertex $v \in V$, sample a set $L(v)$ of $s = \alpha \cdot \log n$ colors from $\{1, \dots, \Delta + 1\}$ uniformly at random and independently, where α is the hidden constant in the Oh-notation of [Theorem 7](#).
- (ii) During the stream, store any edge $e = (u, v)$ in a subgraph H if $L(u) \cap L(v) \neq \emptyset$.
- (iii) At the end of the stream, find a list-coloring of H from the lists $L(v)$ for $v \in V$ or output FAIL if no list-coloring exists.

We can bound the space complexity of this algorithm with $O(n \log^2(n))$ using the same exact argument as in [Lemma 3](#), and thus we omit the proof. The fact that this algorithm never outputs a wrong coloring is also exactly as before for [Algorithm 1](#). Finally, the Palette Sparsification Theorem ([Theorem 7](#)) ensures that the algorithm does not output FAIL with high probability, ensuring its correctness.

We should note that [Algorithm 2](#), as it is, takes exponential time as the list-coloring problem it aims to solve at the end is in general NP-hard. As our main focus in the streaming model is the space of the algorithm, this is not a total deal breaker. That being said, considering the list-coloring instances that arise in this algorithm have additional structures (as we shall see in the proof of the Palette Sparsification Theorem), one can in fact design a very fast algorithm for this list-coloring also; see [\[2\]](#) for this part.

Remark. As is clear from [Algorithm 2](#), almost the entirety of the semi-streaming algorithm for $(\Delta + 1)$ coloring relies on the Palette Sparsification Theorem and the rest can be handled pretty easily. It turns out that the surprising power of the Palette Sparsification Theorem is not limited to the streaming setting and one can use this theorem to design various other sublinear algorithms for $(\Delta + 1)$ coloring and beyond. We refer the interested reader to [\[2, 1\]](#) for further details on this theorem and its extensions and applications.

4 Palette Sparsification Theorem

We now present a proof of the palette sparsification theorem. The general strategy of the proof is as follows: the greedy list-coloring approach to coloring G from the sampled lists $\{L(v) \mid v \in V\}$ works fine for coloring vertices of degree $(1 - \Omega(1)) \cdot \Delta$ using the same proof as [Lemma 4](#); as we shall see, this proof can be extended even to subgraphs of G that are “locally sparse”, i.e., with “sufficient” number of non-edges in the subgraph. At the same time, to color the subgraphs of G that are “almost a clique”, we can use a similar strategy as the random graph theory approach of [Lemma 5](#). Thus for a complete proof, we are going to decompose the

graph into sparse and dense subgraphs and apply each of these ideas separately to each part, while showing that the resulting partial colorings of each part can be extended to the entire graph.

Remark. This approach of coloring sparse and dense parts of the graph separately has a rich history in the graph theory literature starting from the pioneering work of Reed [7]. The very high level idea is typically as follows: the sparse vertices do not have much structure but can be colored greedily while the dense part typically cannot be colored greedily but has a “nice” structure that one can exploit in a more global argument^a.

^aThe reason this global argument is not applied to the entire graph is the lack of any structure for sparse vertices which makes the arguments tedious or downright infeasible to carry out there.

4.1 A Sparse-Dense Decomposition

We shall use the following decomposition which is a simple extension of the classical graph decomposition of Reed in [7] and numerous follow-ups (see Chapter 15 of the excellent book of Molloy and Reed on graph coloring [6] for more details).

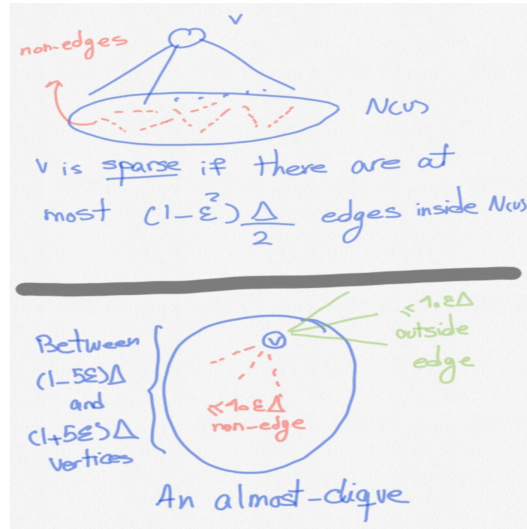


Figure 5: An illustration of the decomposition in Proposition 8.

Proposition 8. Let $\varepsilon < 1/50$ and $G = (V, E)$ be any arbitrary graph. We can decompose the vertices of G into the sets $V_{\text{sparse}}, C_1, \dots, C_k$ with the following properties:

- (i) For every vertex $v \in V_{\text{sparse}}$, the total number of edges between the neighbors of v is $(1 - \varepsilon^2) \cdot \Delta^2/2$;
- (ii) Any set of vertices C_i , called an almost-clique⁶, has the following properties:
 - (a) $(1 - 5\varepsilon)\Delta \leq |C_i| \leq (1 + 5\varepsilon)\Delta$;
 - (b) Any vertex $v \in C_i$ has at most $10\varepsilon\Delta$ neighbors outside of C_i ;
 - (c) Any vertex $v \in C_i$ has at most $10\varepsilon\Delta$ non-neighbors inside of C_i .

Proof. Let D denote the set of all vertices with more than $(1 - \varepsilon^2) \cdot \Delta^2/2$ edges among their neighbors. For any vertex $v \in D$, define the following set S_v :

⁶The reason behind the name should be clear from the properties; each C_i can be turned into a $(\Delta + 1)$ -clique by changing $\approx \varepsilon$ fraction of its vertices and edges.

1. Let $S_v \leftarrow N(v) \cup \{v\}$ initially;
2. While there is a vertex $u \in S_v$ with $|N(u) \cap S_v| \leq (1 - 4\varepsilon)\Delta$, delete u from S_v ;
3. While there is a vertex $w \in V \setminus S_v$ with $|N(w) \cap S_v| \geq (1 - 4\varepsilon)\Delta$, insert w to S_v .

Considering there is no alternation between deleting and inserting vertices to S_v , this process gives a unique set S_v . We now list the following properties of the sets S_v for all $v \in D$:

- (i) every vertex $u \in S_v$ has at least $(1 - 4\varepsilon)\Delta$ neighbors in S_v ;

Proof: by the definition of the set S_v itself.

- (ii) every vertex $u \in S_v$ has at most $4\varepsilon\Delta$ neighbors outside of S_v ;

Proof: by part (i) and the fact that degree of each vertex is at most Δ .

- (iii) $|N(v) \cap S_v| \geq (1 - 3\varepsilon)\Delta$ and at least $(1 - 3\varepsilon)\Delta$ vertices in $N(v)$ have $\leq 3\varepsilon\Delta$ neighbors outside of S_v .

Proof: we define the non-degree of $u \in N(v)$ as the number of vertices in $N(v)$ that are not neighbor to u . As the average degree of vertices in $N(v)$ is $\geq (1 - \varepsilon^2)\Delta$ and $|N(v)| \leq \Delta$, the average non-degree is $\leq \varepsilon^2\Delta$. By Markov bound, number of vertices $U \subseteq N(v)$ with non-degree $> \varepsilon\Delta$ is less than $\varepsilon\Delta$. As such, each vertex of $N(v) \setminus U$ has more than $|N(v)| - \varepsilon\Delta - \varepsilon\Delta > (1 - 3\varepsilon)\Delta$ neighbors inside $N(v) \setminus U$ (note that $|N(v)| > (1 - 2\varepsilon^2)\Delta$ as otherwise v cannot belong to D even if its neighborhood is a clique). Finally, it can be seen that $S_v \supseteq N(v) \setminus U$ as none of the vertices of $N(v) \setminus U$ would ever be deleted in the definition of S_v , implying the property.

- (iv) $|S_v \setminus N(v)| \leq 5\varepsilon\Delta$ and the total number of edges going out of S_v is $\leq 4\varepsilon\Delta^2$.

Proof: consider the set S_v at the end of the step (2) of its definition. At this point, the total number of edges going from S_v to outside vertices by parts (iii) and (ii) is $\leq (1 - 3\varepsilon)\Delta \cdot 3\varepsilon\Delta + 3\varepsilon\Delta \cdot 4\varepsilon\Delta < 4\varepsilon\Delta^2$. At the same time, whenever we add a new vertex w to S_v in the step (3), we reduce the number of outgoing edges by $\geq (1 - 4\varepsilon)\Delta$ due to the inclusion of w , and increase it by $\leq 4\varepsilon\Delta$ for the new contribution of the outside edges of w . As such, the total number of times we can include a new vertex w to S_v at step (3) is $(4\varepsilon\Delta^2)/(1 - 8\varepsilon)\Delta \leq 5\varepsilon\Delta$. This means that $|S_v \setminus N(v)| \leq 5\varepsilon\Delta$.

- (v) If $S_v \cap S_u \neq \emptyset$, then u belongs to S_v and v belongs to S_u ;

Proof: let w be any vertex in $S_v \cap S_u$ and note that by part (i), $N(w)$ intersects both S_v and S_u in at least $(1 - 4\varepsilon)\Delta$ vertices. By part (iv), this also implies that

$$|N(v) \cap N(w)| \geq (1 - 4\varepsilon - 5\varepsilon)\Delta > 4\Delta/5 \quad \text{and} \quad |N(u) \cap N(w)| > (1 - 4\varepsilon - 5\varepsilon)\Delta > 4\Delta/5.$$

As such $|N(u) \cap N(v)| \geq 3\Delta/5$. Now, consider the set of edges between $N(v) \cap N(u)$ and $N(u) \setminus N(v)$. On one hand, all these edges are from $N(v)$ to outside $N(v)$ and there can be only $\varepsilon^2\Delta^2$ such edges since $v \in D$. On the other hand, all these edges are inside $N(u)$ and there are at most $\varepsilon^2\Delta^2$ non-edges inside $N(u)$ since $u \in D$. This implies that

$$|N(v) \cap N(u)| \cdot |N(u) \setminus N(v)| - \varepsilon^2\Delta^2 \leq |E(N(v) \cap N(u), N(u) \setminus N(v))| \leq \varepsilon^2\Delta^2.$$

Combining with the (loose) lower bound of $|N(u) \cap N(v)| \geq 3\Delta/5$, we get $|N(u) \setminus N(v)| < 4\varepsilon^2\Delta$. This gives a much stronger lower bound of $|N(v) \cap N(u)| > (1 - \varepsilon)\Delta$. Finally, this means that

$$|N(v) \cap S_u| > (1 - 4\varepsilon)\Delta \quad \text{and} \quad |N(u) \cap S_v| > (1 - 4\varepsilon)\Delta,$$

implying that $v \in S_u$ and $u \in S_v$ by steps (2)/(3) of the procedure.

- (vi) v belongs to S_v ;

Proof: By part (iii), $|N(v) \cap S_v| > (1 - 3\varepsilon)\Delta$ and thus v never gets deleted from S_v in step (2).

Let us now construct the decomposition. We go over vertices of D in some arbitrary order and pick $v \in D$ and corresponding S_v to form an almost-clique C ; we then remove all of S_v from D . Properties (v) and (vi) ensures that the collection of S_v -sets picked in this procedure partitions D and that each almost-clique is corresponding to some set S_v . This ensures that for any set $C (= S_v \text{ for some } v \in D)$:

- $(1 - 3\varepsilon)\Delta \leq |C| \leq (1 + 5\varepsilon)\Delta$: by properties (iii) and (iv) of $S_v = C$;
- each vertex $u \in C$ has at most $4\varepsilon\Delta$ outside neighbors: by property (ii) of $S_v = C$;
- each vertex $u \in C$ has at most $9\varepsilon\Delta$ inside non-neighbors: by property (i) and (iv) of $S_v = C$.

As a result each C is truly an almost-clique as desired in [Proposition 8](#). Finally, any remaining vertex after this step does not belong to D and thus can be safely placed in V_{sparse} , concluding the proof. \square

Throughout the proof of [Theorem 7](#), we pick $\varepsilon := 10^{-10}$, chosen so that any constant arising in the proofs times $\varepsilon < 1$ and we do not mention this explicitly each time. We then fix a decomposition of the input graph $G = (V, E)$ with this parameter ε given by [Proposition 8](#) and let V_{sparse} and C_1, \dots, C_k refer to the sets of this decomposition. We assume the number of sampled colors is $O(\varepsilon^{-4} \cdot \log n)$ (which is $\Theta(\log n)$ considering $\varepsilon = \Theta(1)$). We also assume that $\Delta = \Omega(\varepsilon^{-4} \cdot \log n)$ as otherwise sampling the colors in [Theorem 7](#) will collect all available colors of each vertex and the theorem become vacuously true⁷.

Remark. The decomposition given in [Proposition 8](#) is somewhat different, in terms of the constructions but not the properties, than the one used in the original proof of [\[2\]](#), which was motivated by the HSS network decomposition of [\[4\]](#). The reason for the choice of [\[2\]](#) is that it is easier to obtain the decomposition using the existing result of [\[4\]](#) compared to the approach above, and the resulting decomposition is more suitable for the “algorithmic” palette sparsification that is needed to obtain fast algorithms for the list-coloring part compared to the exponential-time algorithms in this lecture.

4.2 Part One: Coloring Vertices in V_{sparse}

We first show that all vertices in V_{sparse} are list-colorable from the sampled lists with high probability. The proof of this part closely follows the classical approaches in graph theory [\[5\]](#) for proving that “locally sparse” vertices, as in V_{sparse} , can be colored with much less than Δ colors. The approach is to first show that we can color a constant fraction of vertices $v \in V_{\text{sparse}}$ using colors of $L(v)$ in a way that the remaining uncolored graph basically becomes a “low degree” graph and then apply the same argument as in [Lemma 4](#) to finalize the proof. We note that throughout this proof, we solely focus on the induced subgraph of G on V_{sparse} (notice that property of V_{sparse} vertices in [Proposition 8](#) continues to hold here as well as we may have only reduced the number of edges in their neighborhoods).

Part One-(A): Creating Excess Colors

Consider the following algorithm for a partial list-coloring of V_{sparse} :

Algorithm 3. An algorithm for partially list-coloring a subset of V_{sparse} from the sampled lists.

- (i) For any vertex $v \in V_{\text{sparse}}$, sample a color $c(v)$ uniformly at random from $L(v)$.
- (ii) If the color $c(v) \neq c(u)$ for any neighbor $u \in N(v)$, we **color** v with $c(v)$.

⁷We note that as stated, the hidden constant in $O(\log n)$ samples for this theorem is $\sim 10^{40}$ or so (!). However, we made no attempt to reduce the constants and simply used the simplest choices throughout to help clarify the (already complicated enough) proof. Indeed, one can reduce these constants dramatically without changing any part of the proof; that being said, the limit of this approach is still a large constant $\sim 10^3$ or so.

For any vertex $v \in V_{\text{sparse}}$, define:

- $A_1(v)$: the colors in $\{1, \dots, \Delta + 1\} - c(v)$ that are not used to color any neighbor of v in **Algorithm 3**; these colors are *available* to v (if included in $L(v)$), for the next step of coloring V_{sparse} .
- $\deg_1(v)$: the number of uncolored neighbors of v after **Algorithm 3**.

The intuition behind **Algorithm 3** is simply as follows: for any pairs of vertices in the neighborhood of a vertex $v \in V_{\text{sparse}}$ that are colored the same, $A_1(v)$ reduces by one while $\deg_1(v)$ reduces by two. As such, if we can have a “large” number of pairs of vertices in the neighborhood of v with the same color, then v becomes “low degree” compared to the number of available colors it has (and then we know how to handle low degree vertices similar to **Lemma 4** in the next part). Now considering the fact that there are many non-adjacent pairs of vertices in $N(v)$ as $v \in V_{\text{sparse}}$, we hope that **Algorithm 3** would be able to color enough of number of such pairs with the same color⁸. We now formalize this as follows.



Figure 6: An illustration of the intuition behind **Algorithm 3** and the proof of **Lemma 9**: the more pairs of vertices we can color the same in $N(v)$, the more colors we “save” for the next step.

Lemma 9. *After running **Algorithm 3**, with high probability, $|A_1(v)| \geq \deg_1(v) + 10^{-4} \cdot \varepsilon^2 \Delta$.*

Proof. For the purpose of this proof, let us assume that degree of v is exactly Δ : this can be achieved by adding a set of $\Delta - \deg(v)$ ‘dummy’ vertices to the neighborhood of v . Considering v belongs to V_{sparse} , by **Proposition 8**, the number of edges in the neighborhood of v is at most $(1 - \varepsilon^2) \cdot \Delta^2/2$, which implies that there are $\geq \binom{\Delta}{2} - (1 - \varepsilon^2)\Delta^2/2 = \varepsilon^2\Delta^2/2 - \Delta/2$ non-edges between neighbors of v (including the new dummy vertices). Let $t = \varepsilon^2\Delta^2/3$ and f_1, \dots, f_t denote t arbitrary non-edge in $N(v)$. Additionally, define the following random variable:

- X : number of colors that are sampled by *at least* two neighbors of v and are additionally retained by *all* these neighbors.

⁸Strictly speaking, the statement that “there are many non-adjacent pairs of vertices in $N(v)$ as $v \in V_{\text{sparse}}$ ” is not true as when $\deg(v) < (1 - \varepsilon)\Delta$, $N(v)$ can be a clique and v would still be in V_{sparse} . But in that case v is already low degree and there is nothing we need to do anyway.

Since any color counted in X is used more than once to color a neighbor of v , we have

$$|A_1(v)| \geq \deg_1(v) + X - 1, \quad (1)$$

where the -1 term is because of $c(v)$ itself. Let us first lower bound the expected value of X .

Claim 10. $\mathbb{E}[X] \geq e^{-6} \cdot \varepsilon^2 \cdot \Delta$.

Proof. We define Y as the number of colors that are sampled by the endpoints of exactly one of \bar{f}_i 's and are retained by both endpoints. Clearly, $X \geq Y$. We thus lower bound Y instead. For every non-edge $\bar{f}_i := (u_i, w_i)$, define the indicator random variable \bar{f}_i where $\bar{f}_i = 1$ if the following two conditions happen:

- $c(u_i) = c(w_i)$, and,
- for all $z \in N(v) \cup N(u_i) \cup N(w_i) \setminus \{u_i, w_i\}$, $c(z) \neq c(u_i)$ (this way, $c(u_i) = c(w_i)$ and is assigned to both of them for sure);

otherwise $\bar{f}_i = 0$. By definition, $Y = \sum_{i=1}^t Y_i$. We have,

$$\begin{aligned} \Pr[Y_i = 1] &= \Pr[c(u_i) = c(w_i) \wedge \forall z \in N(u_i) \cup N(w_i) \cup N(v) \setminus \{u_i, w_i\} : c(z) \neq c(u_i)] \\ &\geq \frac{1}{\Delta+1} \cdot \left(1 - \frac{1}{\Delta+1}\right)^{\Delta+(\Delta-1)+(\Delta-2)} \\ &\quad (\text{the color of each vertex is chosen independently and uniformly at random from } \Delta+1 \text{ colors}) \\ &\geq \frac{1}{\Delta+1} \cdot \exp\left(-\frac{4\Delta}{\Delta+1}\right) \geq \frac{e^{-4}}{\Delta+1}. \\ &\quad (1-x \geq e^{-4x/3} \text{ for sufficiently small } x \in (0, 1/10) \text{ and since } \Delta \text{ is } \omega(1)) \end{aligned}$$

By linearity of expectation, $\mathbb{E}[X] \geq \mathbb{E}[Y] \geq t \cdot \frac{e^{-4}}{\Delta+1} \geq e^{-6} \cdot \varepsilon^2 \Delta$ as $t = \varepsilon^2 \Delta^2 / 3$. \square **Claim 10**

The next step is to prove that X is concentrated. The proof of this part is a rather involved yet standard exercise in probabilistic arguments (see, e.g. [6, Chapter 10]) and can be skipped by the reader.

The proof uses Talagrand's inequality: A function $f(x_1, \dots, x_n)$ is called *c-Lipschitz* if changing any x_i can affect the value of f by at most c . Additionally, f is called *r-certifiable* iff whenever $f(x_1, \dots, x_n) \geq s$, there exists at most $r \cdot s$ variables $x_{i_1}, \dots, x_{i_{r \cdot s}}$ so that knowing the values of these variables certifies $f \geq s$.

Proposition 11 (Talagrand's inequality; cf. [6]). *Let X_1, \dots, X_n be n independent random variables and $f(X_1, \dots, X_n)$ be a c -Lipschitz and r -certifiable function. For any $t \geq 1$,*

$$\Pr\left(|f - \mathbb{E}[f]| > t + 30c\sqrt{r \cdot \mathbb{E}[f]}\right) \leq 4 \exp\left(-\frac{t^2}{8c^2 r \mathbb{E}[f]}\right).$$

Remark. Perhaps the most common concentration inequality used in probabilistic analysis is Chernoff bound. Despite its simplicity, Chernoff bound is applicable to a surprisingly large number of situation. Yet, there are still some limits to the power of this basic tool. In such cases, one may want to consider stronger tools such as Talagrand's, McDiarmid's, or Azuma's inequalities. At a very high level, the common theme of all these concentration inequalities is the following: if we have a random variable T which is a function of n independent trials X_1, \dots, X_n and T is not "very sensitive" to the outcome of each trial^a, then T is "more or less" concentrated around its expectation.

^aThis is captured by the notions of Lipschitz constant and certifiably in Talagrand's inequality in **Proposition 11**.

We still cannot apply **Proposition 11** directly to bound the deviation of X as X is not a certifiable function of the independent random variables in this context, i.e., $c(v)$ for $v \in V_{\text{sparse}}$. As such, we prove the concentration of X indirectly as follows. Define the following two additional variables:

- W : number of colors that are sampled by *at least* two neighbors of v , regardless of whether they are assigned to them or not.
- Z : number of colors that are sampled by *at least* two neighbors of v but are *not* assigned to *at least one* of them.

It is clear that $X = W - Z$. Also notice that both W and Z are functions of *independent* random variables that define the choices of random colors $c(v)$ for every $v \in V_{sparse}$. Both W and Z are $\Theta(1)$ -certifiable (for W point to two neighbors of v that sampled the color; for Z additionally point to one of the neighbors of this pair that also sampled the color, hence not allowing one of them to retain it). They are also both $\Theta(1)$ -Lipschitz: changing choice of one color for a vertex can only affect the two colors involved (the original one and the changed one). As such, we can apply Talagrand's inequality ([Proposition 11](#)) to obtain that (we only write the bound for W ; the same exact argument works also for Z):

$$\begin{aligned} \Pr(|W - \mathbb{E}[W]| \geq \mathbb{E}[X]/10) &\leq \exp\left(-\Theta(1) \cdot \frac{(\mathbb{E}[X] - \Theta(1)\sqrt{\Delta})^2}{\Delta}\right) && (W \leq \Delta/2 \text{ always}) \\ &\leq \exp(-\Theta(\varepsilon^4) \cdot \Delta) \ll n^{-10}. \\ &\text{(by [Claim 10](#) on expected value of } X \text{ and since we can assume } \Delta \text{ to be } \gg \varepsilon^{-4} \cdot \ln n) \end{aligned}$$

As such, we obtain that w.h.p. both W and Z are concentrated and thus also w.h.p.,

$$X = W - Z \geq \mathbb{E}[W] - \mathbb{E}[X]/10 - (Z + \mathbb{E}[X]/10) = \mathbb{E}[X] - \mathbb{E}[X]/5 = (4/5) \cdot \mathbb{E}[X].$$

Plugging in this and [Claim 10](#) in [Eq \(1\)](#) implies that $|A_1(v)| \geq \deg(v) + e^{-7} \cdot \varepsilon^2 \Delta$ with high probability. A union bound over all vertices finalizes the proof as $e^7 < 10^4$. \square

Part One-(B): Finalizing the Coloring of V_{sparse}

In the following, we condition on the high probability event of [Lemma 9](#). We can now use the same greedy algorithm in [Algorithm 1](#) to finalize the coloring of vertices in V_{sparse} .

Algorithm 4. An algorithm for finalizing the list-coloring of V_{sparse} from the sampled lists.

- (i) Go over uncolored vertices v in V_{sparse} in some arbitrary order;
- (ii) For any vertex v , if $L(v) = \emptyset$ output FAIL; otherwise, **color** v from any color c in $L(v)$ and remove the color c from $L(u)$ for any $u \in N(v)$.

Lemma 12. *With high probability, [Algorithm 4](#) does not output FAIL.*

Proof. Suppose we are at the step of coloring a vertex $v \in V_{sparse}$. Recall that at the beginning of [Algorithm 4](#), the list of available colors to v were $A_1(v)$ and by the time we get to process v , at most $\deg_1(v)$ other colors have been removed from $A_1(v)$, leaving us with at least $10^{-4} \cdot \varepsilon^2 \Delta$ choices by [Lemma 9](#).

Now, if $L(v)$ contains any of these colors, we will be able to color v and no need to output FAIL at this point. By the independence of the choice of lists, we can think of $L(v) - c(v)$ (where $c(v)$ is the color used in [Algorithm 3](#)) as being sampled only now. As such, by considering the $s - 1$ random colors remained in $L(v)$ at this point, we have,

$$\begin{aligned} \Pr(L(v) \text{ has no color for coloring } v) &\leq \left(1 - \frac{10^{-4} \cdot \varepsilon^2 \Delta}{\Delta + 1}\right)^{(s-1)} \leq \exp(-10^{-4} \cdot \varepsilon^{-2}(s-1)/2) \leq n^{-5}. \\ &\text{(by sampling } s \geq 10^{-5} \cdot \varepsilon^2 \log n \text{ colors in } L(v) \text{ which is } O(\log n) \text{ still)} \end{aligned}$$

A union bound over all $v \in V_{sparse}$ implies that the algorithm does not output FAIL with high probability. \square

[Lemmas 9](#) and [12](#) now imply that we can list-color all vertices in V_{sparse} from the lists $L(v)$ with high probability, finishing the first part of the proof of [Theorem 7](#).

4.3 Part Two: Coloring Almost-Cliques One by One

We now start the main part of the proof of [Theorem 7](#) which concerns coloring each almost-clique. Our approach would be to go over each almost-clique C one by one, assume that every vertex outside C is colored (even adversarially), and show that the randomness in the choice of $L(v)$ for $v \in C$ is still enough to list-color C in a way that it is also consistent with the coloring of outside vertices. The main idea is to use a generalization of the random graph theory approach used in [Lemma 5](#). However, this can be tricky when working with an almost-clique that contains “many” non-edges inside⁹. As a result, the first step of our approach here is a pre-processing step for almost-cliques with a “large” number of non-edges.

Throughout this section, we fix an almost-clique C in the decomposition of [Proposition 8](#); fix the choice of all lists $L(v)$ for $v \notin C$, and assume that the subgraph $G - C$ is already colored, even adversarially. We then show that with high probability, we can still color C from the sampled lists $L(v)$ for $v \in C$ in a way that it is consistent with the coloring of $G - C$. Having achieved this, we can go over almost-cliques of G one by one and use this approach to color them.

A note on sampling the lists. We are going to make one simplifying assumption: instead of each vertex sampling $\Theta(\log n)$ colors from $\{1, \dots, \Delta + 1\}$, we think of each vertex v as having $k := \Theta(\log n)$ lists $L_1(v), \dots, L_k(v)$ where in each one, we sample each color of $\{1, \dots, \Delta + 1\}$ with probability $q := \frac{1}{100\sqrt{\varepsilon}\Delta}$ and another list $L^*(v)$ wherein we sample each color with probability $p := \frac{\Theta(\log n)}{\Delta}$ (we specify the constant of Θ -notation in [Lemma 19](#)); we then think of

$$L(v) := L_1(v) \cup \dots \cup L_k(v) \cup L^*(v).$$

The latter process still samples lists of size $O(\log n)$ colors *with high probability* for each vertex and thus proving the result for the latter process also implies the former (with a polynomially small loss on the success probability and a constant factor larger list sizes). This assumption, while being without loss of generality, is quite helpful for us to maintain the (much) needed independence between the variables.

Part Two-(A): Pre-Processing the Almost-Clique C

We follow a similar high level strategy as Part One-(A), by trying to color *two* vertices in the almost-clique C with the same color, hence “buying” ourselves some extra colors for the main part of the argument. However, the way this coloring is obtained and its properties are entirely different from that of [Algorithm 3](#). In particular, this time we need to be able to exploit the case when we have only $\approx \varepsilon\Delta$ non-edge in an almost-clique compared to the case of Part One-(A) that worked with $\approx \varepsilon^2\Delta^2$ non-edges. On the other hand, the structure of almost-cliques allows us to do this more efficiently than the random coloring of [Algorithm 3](#).

The general idea of the algorithm is to find a *matching* among the *non-edges* of C such that both endpoints of this non-edge can be colored the same (from a color shared in the list of both endpoints which also does not appear in the neighborhood of either outside C), while using a different color for each of its edges, namely, a *colorful non-edge matching*. Such an approach would reduce the number of vertices of C at twice the rate of the number of used colors, hence preparing C for the last step of our algorithm¹⁰.

Let us briefly go over the intuition behind the algorithm first. Let t denote the number of non-edges in C and suppose $t \gtrsim \Delta$ (this is the regime we need the preprocessing for). For us to be able to include a non-edge $\bar{f} = (u, v)$ in \bar{M} we need $L(u) \cap L(v) \neq \emptyset$ —moreover, this intersection should be on a color c not used in the neighborhood of each vertex; considering out-degrees of u, v is $\approx \varepsilon\Delta$ by the decomposition of [Proposition 8](#),

⁹Among other challenges, such an almost-clique may even have more than $\Delta + 1$ vertices. In [Lemma 5](#), we modeled the problem as a matching problem—when number of vertices is $> \Delta + 1$ we certainly cannot hope to get a matching as some colors need to be assigned to at least two vertices; the vertices that are colored the same should also be an independent set in the original graph, a constraint that makes using the random graph theory approach rather infeasible.

¹⁰Again, for intuition, consider the case when $|C| = \Delta + 2$ and thus originally we cannot hope to get a matching of all vertices in C to the $\Delta + 1$ colors. However, such an almost-clique C necessarily has $\geq \Delta$ non-edges and so Part Two-(B) will be able to find a non-edge matching of size at least 1 inside C ; by coloring the endpoints of this non-edge we will have $\Delta + 2 - 2 = \Delta$ remaining vertices to be colored and $\Delta + 1 - 1 = \Delta$ available colors; thus, at least finding a matching of remaining vertices to the colors is not completely out of the question.

we still have that $L(u) \cap L(v)$ is non-empty with probability $\approx 1/\Delta$ (as we calculated in Lemma 3; here, we are omitting the $O(\log n)$ factor as it is not relevant). Thus, we expect $\approx t/\Delta$ non-edges to be readily available to be added to \bar{M} . Moreover, considering the non-edge degree of vertices in C is $\approx \varepsilon\Delta$ by the decomposition of Proposition 8, after considering the non-edges with intersecting lists, we expect that there is a non-edge *matching* of size $\approx t/\Delta$. Finally, we need this matching to be colorful as well but one expects that the randomness in the choices of lists and their intersection over the non-edges of the matching already takes care of that, leading us to a non-edge matching of size $\approx t/\Delta$, which is sufficient for us.

Turning this intuition into an actual proof is tricky however considering the non-trivial correlations in the events above, and in particular as (i) the randomness is over the vertices of the graph (or rather their lists), while we would like to work with the non-edges, (ii) including some non-edge into the matching prohibits inclusion of several others due to the matching constraint, and (iii) the set of colors that can be used across non-edges is correlated due to the coloring of vertices outside the almost-clique. As a result, we turn the above intuition to an actual proof in a rather indirect way (this is also the reason we partitioned the lists of each vertex into $\Theta(\log n)$ parts and focused on each one individually for this step). In the following, we define our algorithm for each of the lists $L_i(v)$ for $i \in [k]$ and show that each list can “succeed” with constant probability; considering we have $\Theta(\log n)$ lists, at least one of them succeeds with high probability.

Algorithm 5. An algorithm for finding a “colorful non-edge matching” in the almost-clique C using only the lists $\{L_i(v) \mid v \in V\}$.

1. Let $\bar{F} := \bar{f}_1, \dots, \bar{f}_t$ be the set of non-edges in C , i.e., pairs (u, v) in C with no edge between u and v .
2. Iterate over colors $c \in \{1, \dots, \Delta + 1\}$ in an arbitrary order:
 - (a) If there is $\bar{f} := (u, v) \in \bar{F}$ s.t. $c \in L_i(u) \cap L_i(v)$ and c is not assigned to any neighbor of u, v yet: add \bar{f} to \bar{M} , remove all non-edges incident on u, v from \bar{F} , and continue to the next color.

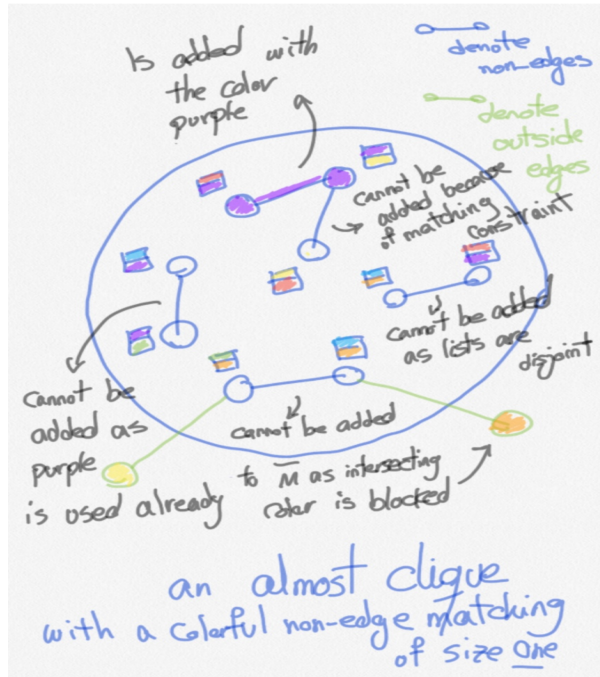


Figure 7: An illustration of a colorful non-edge matching in Algorithm 5: in this example, the matching contains a single non-edge marked in purple.

Let \bar{M} denote the non-edge matching returned by [Algorithm 5](#). We can use this non-edge matching to color endpoints of any $\bar{f} \in \bar{M}$ with the color c assigned to \bar{f} when we added it to \bar{M} ; as each color is given only for one non-edge and these colors both belong to the lists of the endpoints and at the same time have not been used in the neighborhood of either, this would be a consistent coloring with the rest of the graph. As such, after this step, if we define $\text{Remain}(C)$ as the set of remaining uncolored vertices in C and $\text{Colors}(C)$ as the set of colors not used inside C still, we will have:

$$|\text{Remain}(C)| = |C| - 2 \cdot |\bar{M}|, \quad |\text{Colors}(C)| = \Delta + 1 - |\bar{M}|. \quad (2)$$

We will only need [Algorithm 5](#) for almost-cliques with a “non-trivial number” of non-edges which is $\gtrsim \varepsilon\Delta$. In that case, the following lemma shows that we can find a colorful non-edge matching which is smaller than the number of non-edges by roughly $\approx \varepsilon\Delta$.

Lemma 13. *Suppose C has at least $t \geq 10^7 \cdot \varepsilon\Delta$ non-edges in \bar{F} . Then, [Algorithm 5](#) finds a colorful non-edge matching of size at least $\ell := 10^{-6} \cdot t/\varepsilon\Delta$ with probability at least $1/2$.*

Proof. In the following, we assume that the algorithm right away terminates if it finds a non-edge matching of size ℓ . For any color c when considered in Line (ii) of [Algorithm 5](#), we define:

- *Present(c)*: the set of non-edges in \bar{F} at the time of processing c such that c is not used in the neighborhood of either endpoints, i.e.,

$$\text{Present}(c) := \{(u, v) \in \bar{F} \mid \text{no neighbors of } u \text{ or } v \text{ is colored with } c\};$$

let $\text{pre}(c) := |\text{Present}(c)|$.

- We say that a color c is *successful* if we add a non-edge to \bar{M} when iterating c in [Algorithm 5](#); as such, $|\bar{M}|$ is equal to the number of successful colors.

We first show that for most colors, $\text{pre}(c)$ is sufficiently large. We say a color c is *heavy* iff $\text{pre}(c) \geq t/2$ and otherwise it is *light*. We first prove that at least half the colors are heavy.

Claim 14. *We have at least $\Delta/2$ heavy colors.*

Proof. For $\bar{f} = (u, v)$, let $B(\bar{f})$ be the set of colors used in the neighborhood of either u or v outside C . Since out-degree of any vertex in C is at most $10\varepsilon\Delta$ by [Proposition 8](#), $|B(\bar{f})| \leq 20\varepsilon\Delta$. As a result, at the beginning of the algorithm,

$$\sum_{\bar{f} \in \bar{F}} |B(\bar{f})| \leq t \cdot 20\varepsilon\Delta.$$

This means that an average color appears in $B(\bar{f})$ for at most $20\varepsilon t$ non-edges \bar{f} ; an application of Markov bound now implies that for at least $\Delta/2$ colors c , the color $c \in B(\bar{f})$ for at most $40\varepsilon t$ non-edges \bar{f} . This implies that at least $\Delta/2$ colors are “present” in at least $t - 80\varepsilon t \geq 0.9 \cdot t$ non-edges.

Additionally, whenever we add a non-edge (u, v) to \bar{M} , we remove all non-edge neighbors of u, v from \bar{F} ; considering the in-non-degree of any vertex in C is at most $10\varepsilon\Delta$ by [Proposition 8](#), we may delete up to $\ell \cdot 20\varepsilon\Delta = (0.1) \cdot t$ non-edges from \bar{F} ever. As such, for each of the $\Delta/2$ colors c found above, c can be used to color $\geq 0.9t - (0.1)t$ non-edges, thus $\text{pre}(c) \geq t/2$ for these $\Delta/2$ colors, making them heavy. \square [Claim 14](#)

We now use the above claim to say that whenever we are processing a heavy color, there is a “good” probability that we add a new edge to \bar{M} .

Claim 15. *For any heavy color c , $\Pr(c \text{ is successful}) \geq 10^{-5} \cdot t/\varepsilon\Delta^2$.*

Proof. The color c is successful if at least one non-edge $(u, v) \in \text{Present}(c)$ have $c \in L_i(u) \cap L_i(v)$. By a simple application of the inclusion-exclusion principle,

$$\begin{aligned} \Pr(c \text{ is successful}) &\geq \Pr(\text{at least one } (u, v) \in \text{Present}(c) \text{ have } c \in L_i(u) \cap L_i(v)) \\ &\geq \sum_{\substack{(u,v) \in \\ \text{Present}(c)}} \Pr(c \in L_i(u) \cap L_i(v)) - \sum_{\substack{(u,v), (u',v') \in \\ \text{Present}(c)}} \Pr(c \in L_i(u) \cap L_i(v) \cap L_i(u') \cap L_i(v')). \end{aligned} \quad (3)$$

The first term above is easy to bound as each c belongs to $L_i(u) \cap L_i(v)$ w.p. q^2 , hence,

$$\sum_{\substack{(u,v) \in \\ \text{Present}(c)}} \Pr(c \in L_i(u) \cap L_i(v)) = \text{pre}(c) \cdot q^2.$$

For the second term of Eq (3), there are two types of pairs of (distinct) non-edges $(u, v), (u', v')$ that we need to take into account: the ones that share exactly one endpoint and the ones that do not share any endpoint. There are at most $\text{pre}(c) \cdot 10\varepsilon\Delta$ many edges of the first type as maximum non-degree of any vertex is at most $10\varepsilon\Delta$ by Proposition 8; there are also at most $\text{pre}(c)^2$ many edges of the second type. Hence,

$$\sum_{\substack{(u,v), (u',v') \in \\ \text{Present}(c)}} \Pr(c \in L_i(u) \cap L_i(v) \cap L_i(u') \cap L_i(v')) \leq \text{pre}(c) \cdot 10\varepsilon\Delta \cdot q^3 + \text{pre}(c)^2 \cdot q^4.$$

We can plugin the above two bounds in Eq (3), and have,

$$\begin{aligned} \Pr(c \text{ is successful}) &\geq \text{pre}(c) \cdot q^2 - \text{pre}(c) \cdot 10\varepsilon\Delta \cdot q^3 - \text{pre}(c)^2 \cdot q^4 \\ &\geq (0.9) \cdot \text{pre}(c) \cdot q^2 - \text{pre}(c)^2 \cdot q^4 \\ &\quad (\text{as } q = (1/100\sqrt{\varepsilon}\Delta \text{ and thus } 10\varepsilon\Delta \cdot q < (0.1) \text{ for } \varepsilon < 1)) \\ &= (0.9) \cdot \text{pre}(c) \cdot q^2 \cdot (1 - 20\varepsilon\Delta^2 \cdot q^2) \\ &\quad (\text{as } \text{pre}(c) \leq |\bar{F}| \leq 20\varepsilon\Delta^2 \text{ by the decomposition of Proposition 8}) \\ &\geq (0.8) \cdot \text{pre}(c) \cdot q^2 \quad (\text{as } q = (1/100\sqrt{\varepsilon}\Delta) \text{ and thus } (1 - 20\varepsilon\Delta^2 \cdot q^2 = 0.998)) \\ &\geq \frac{t}{10^5 \cdot \varepsilon\Delta^2}. \end{aligned}$$

as desired. \square Claim 15

We are now ready to conclude the proof of Lemma 13. Let $\theta := 10^{-5} \cdot t/\varepsilon\Delta^2$ (the RHS of Claim 15); note that $\theta < 1$ because $t \leq |\bar{F}| \leq 20\varepsilon\Delta^2$ by the decomposition of Proposition 8. Let Z be a random variable sampled from binomial distribution $\mathcal{B}(\Delta/2, \theta)$. By Claim 15 and the fact that there are $\Delta/2$ heavy colors, plus a straightforward coupling argument, for every $\ell' \leq \ell$:

$$\Pr(|\bar{M}| \geq \ell') \geq \Pr(Z \geq \ell'). \quad (4)$$

On the other hand, $\mathbb{E}[Z] = \Delta/2 \cdot \theta = 10^{-5} \cdot t/2\varepsilon\Delta$. By Chernoff bound (Proposition 2),

$$\begin{aligned} \Pr\left(Z < \frac{1}{2} \cdot \mathbb{E}[Z]\right) &\leq \exp\left(-\frac{\mathbb{E}[Z]}{6}\right) \\ &= \exp\left(-\frac{t}{12 \cdot 10^5 \cdot \varepsilon\Delta}\right) \leq \exp\left(-\frac{10^7 \cdot \varepsilon\Delta}{12 \cdot 10^5 \cdot \varepsilon\Delta}\right) \quad (\text{as } t \geq 10^7 \cdot \varepsilon\Delta) \\ &= \exp(-5) < 1/2. \end{aligned}$$

As such, with probability at least $1/2$,

$$Z \geq 10^{-5} \cdot t/4\varepsilon\Delta > 10^{-6} \cdot t/\varepsilon\Delta.$$

proving the lemma by Eq (4). \square

Now, considering we can run [Algorithm 5](#) for $\Theta(\log n)$ choices of lists $\{L_i(v) \mid v \in V\}$, we obtain that with high probability, there is a colorful non-edge matching \bar{M} of size $\ell = t/\varepsilon\Delta$ in the almost-clique C . We can thus **color** the vertices of \bar{M} accordingly, and by [Eq \(2\)](#), obtain the following lemma as a direct corollary.

Lemma 16. *Suppose the number of non-edges of C is $t \geq 10^7 \cdot \varepsilon\Delta$. Then, with high probability, we can color a subset of vertices in C without using $\{L^*(v) \mid v \in C\}$ such that:*

$$|\text{Remain}(C)| = |C| - 2 \cdot \frac{t}{10^6 \cdot \varepsilon\Delta}, \quad |\text{Colors}(C)| = \Delta + 1 - \frac{t}{10^6 \cdot \varepsilon\Delta},$$

where $\text{Remain}(C)$ is the set of uncolored vertices in C and $\text{Colors}(C)$ is the set of colors not used in C .

This concludes the first part of coloring the almost-clique C . We emphasize that we only run [Algorithm 5](#) when the preconditions of [Lemma 16](#) are satisfied and otherwise directly go to the next step.

Part Two-(B): Final Coloring of the Almost-Clique C

The final part of coloring the almost-clique C is to use a similar random graph theory type argument in the spirit of the case for cliques in [Lemma 5](#). Define the following bipartite graph $\mathcal{G}_{\text{base}} := (\mathcal{L}, \mathcal{R}, \mathcal{E}_{\text{base}})$, called the *base palette graph*, for the almost-clique C :

- $\mathcal{L} := \text{Remain}(C)$, i.e., the set of vertices yet to be colored in C ; we will use vertices of $\text{Remain}(C)$ and vertices of \mathcal{L} interchangeably;
- $\mathcal{R} := \text{Colors}(C)$, i.e., the set of colors that are yet to be used inside C ; we will use colors of $\text{Colors}(C)$ and vertices of \mathcal{R} interchangeably;
- $\mathcal{E}_{\text{base}}$: any vertex $v \in \mathcal{L}$ has an edge to any color $c \in \mathcal{R}$ iff c is not used to color any neighbor of v in G outside C .

Similarly, define the *palette graph* of the almost-clique C as the bipartite graph $\mathcal{G} := (\mathcal{L}, \mathcal{R}, \mathcal{E})$ on the same set of vertices as $\mathcal{G}_{\text{base}}$ with the following edges:

- $\mathcal{E} \subseteq \mathcal{E}_{\text{base}}$: we only include an edge $(v, c) \in \mathcal{E}_{\text{base}}$ inside \mathcal{E} as well if $c \in L^*(v)$.

As a result, \mathcal{G} is a subgraph of $\mathcal{G}_{\text{base}}$ obtained by sampling each edge of $\mathcal{G}_{\text{base}}$ with probability p .

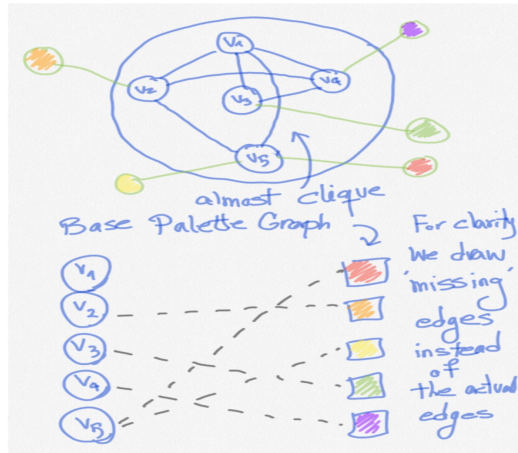


Figure 8: An illustration of the base palette graph—we note that in the figure above, the depicted edges are the ones missing from the base palette graph (i.e., the figure shows the complement of the graph).

In the following, we shall list several key properties of the base palette graph. To start, let us show that \mathcal{L} is always at most as large as \mathcal{R} or alternatively, the number of remaining vertices to be colored is at most the number of available colors.

Claim 17. *In \mathcal{G}_{base} and \mathcal{G} , $|\mathcal{L}| \leq |\mathcal{R}|$.*

Proof. Suppose first that the number of non-edges t in C is $< 10^7 \cdot \varepsilon \Delta$ which is $< \Delta$. Then, $|Remain(C)| = |C| \leq (\Delta + 1)$ (because when $|C| > \Delta + 1$, we right away have $t > \Delta$ by just the bound on maximum degree) and $|Colors(C)| = \Delta + 1$ (because we do not even run [Algorithm 5](#) in this case). Thus, $|\mathcal{L}| \leq |\mathcal{R}|$.

Now suppose $|C| = \Delta + 1 + k$ for some $k > 1$ which necessarily implies $t \geq \Delta > 10^7 \cdot \varepsilon \Delta$ and thus [Algorithm 5](#) was used in this case. We have $t \geq k \cdot \Delta$, which, by [Lemma 16](#), implies that we will find a non-edge matching of size $10^{-6} \cdot k \cdot \Delta / \varepsilon \Delta = 10^{-6} \cdot k / \varepsilon > 2k$. As a result, we have,

$$|\mathcal{L}| = |Remain(C)| = \Delta + 1 + k - 4k = \Delta + 1 - 3k \quad \text{and} \quad |\mathcal{R}| = |Colors(C)| = \Delta + 1 - 2k,$$

which concludes the proof. \square

The fact that $|\mathcal{L}| \leq |\mathcal{R}|$ suggests that that possibility of finding a \mathcal{L} -perfect matching \mathcal{M} in \mathcal{G} that matches all vertices of \mathcal{L} is not entirely out of the question. Assuming such a matching exists in \mathcal{G} , we will be done as in [Lemma 5](#): we can color each vertex v of C with the color of $\mathcal{M}(v)$ and obtain a coloring of C from lists $\{L^*(v) \mid v \in C\}$ (by definition of \mathcal{G}) which is consistent with outside coloring as well (by definition of \mathcal{G}_{base}).

Of course, the fact that $|\mathcal{L}| \leq |\mathcal{R}|$ by no means implies that \mathcal{G} (or even \mathcal{G}_{base}) has a \mathcal{L} -perfect matching. In the following, we list further properties of \mathcal{G}_{base} , which are consequences of the decomposition in [Proposition 8](#) and the preprocessing step in [Lemma 16](#). We then use these to prove the existence of the required matching.

Properties of \mathcal{G}_{base} . Let m denote the number of vertices in \mathcal{L} , i.e. $m := |\mathcal{L}|$ in \mathcal{G}_{base} . Conditioned on the high probability event of [Lemma 16](#), \mathcal{G}_{base} satisfies the following properties:

(i) $m \geq (2/3) \cdot \Delta$;

Proof: the maximum value of t , the number of non-edges in C is at most $20\varepsilon\Delta^2$ by [Proposition 8](#). As such, the minimum possible size of C is at least $(1 - 5\varepsilon)\Delta - \frac{40\varepsilon\Delta^2}{10^6 \cdot \varepsilon \Delta}$ which is at least $(2/3) \cdot \Delta$.

(ii) $m \leq |\mathcal{R}| \leq 2m$;

Proof: by part (i) and the fact that $|\mathcal{R}| \leq \Delta + 1$.

(iii) minimum degree of vertices in \mathcal{L} is at least $2m/3$, i.e., $\min_{v \in \mathcal{L}} \deg_{\mathcal{G}_{base}}(v) \geq 2m/3$.

Proof: any vertex $v \in Remain(C)$ has up to $10\varepsilon\Delta$ edges out of C by [Proposition 8](#) which implies that up to $10\varepsilon\Delta$ colors in \mathcal{R} may be “blocked” for v meaning that v has no edges to them in \mathcal{R} . All the other colors are available to v making the degree of v at least $|\mathcal{R}| - 10\varepsilon\Delta > 2m/3$ by parts (i), (ii).

(iv) for any vertex $v \in \mathcal{L}$,

$$\deg_{\mathcal{G}_{base}}(v) \geq |\mathcal{R}| - (\Delta + 1) + |C| - \overline{\deg}_C(v),$$

where $\overline{\deg}_C(v)$ is the number of non-neighbors of v in C ;

Proof: $\deg_{\mathcal{G}_{base}}(v)$ is equal to the number of colors in $Colors(c)$ that are not blocked for v , i.e., do not appear in the neighborhood of v outside C . Thus, we only need to count the number of neighbors of v outside C which are most $\Delta - (|C| - 1 - \overline{\deg}_C(v))$ as degree of v is at most Δ and it is neighbor to all but $\overline{\deg}_C(v) + 1$ vertices in C (including itself). Thus, $|Colors(C)|$ minus this many colors are not blocked for v , implying the result.

(v) for any set $S \subseteq \mathcal{L}$ of size $|S| \geq m/2$,

$$\sum_{v \in S} \deg_{\mathcal{G}_{base}}(v) \geq (|S| \cdot m) - m/4.^{11}$$

Proof: Suppose first that $t < 10^7 \cdot \varepsilon \Delta$ and thus $|\mathcal{L}| = |C|$ and $|\mathcal{R}| = \Delta + 1$ (as also argued in [Claim 17](#)). By part (iv), and since $\sum_{v \in S} \overline{\deg}_C(v) \leq 2t$, we can write:

$$\sum_{v \in S} \deg_{\mathcal{G}_{base}}(v) \geq \sum_{v \in S} (|\mathcal{L}| - \overline{\deg}_C(v)) \geq |S| \cdot m - 2t \geq |S| \cdot m - 10^7 \cdot 2\varepsilon \Delta \geq |S| \cdot m - m/4,$$

as $\Delta \leq (3/2)m$ by part (i).

Now, consider the other case when $t \geq 10^7 \cdot \varepsilon \Delta$ and let $T := 10^{-6} \cdot t/\varepsilon \Delta$. Thus $|\mathcal{L}| = |C| - 2 \cdot T$ and $|\mathcal{R}| = \Delta + 1 - T$ by [Lemma 16](#). By part (iv),

$$\begin{aligned} \sum_{v \in S} \deg_{\mathcal{G}_{base}}(v) &\geq \sum_{v \in S} (\underbrace{\Delta + 1 - T}_{|\mathcal{R}|} - (\Delta + 1) + \underbrace{|\mathcal{L}| + 2T}_{|C|} - \overline{\deg}_C(v)) \\ &\geq |S| \cdot |\mathcal{L}| + |S| \cdot T - 2t \geq |S| \cdot m + (2/3) \cdot \Delta \cdot 10^{-6} \cdot t/\varepsilon \Delta - 2t > |S| \cdot m, \end{aligned}$$

by the choice of T and part (i) for the second to last inequality.

The above are all the properties of \mathcal{G}_{base} we need for the rest of the proof, i.e., to show that a random subgraph \mathcal{G} of \mathcal{G}_{base} wherein each edge is sampled w.p. q has a \mathcal{L} -perfect matching. As a warm-up, let us first quickly argue that \mathcal{G}_{base} itself has a \mathcal{L} -perfect matching (we actually do not need this claim for the proof but it will provide a basic intuition).

Claim 18. *Any graph \mathcal{G}_{base} satisfying the properties (i) to (v) above has a \mathcal{L} -perfect matching.*

Proof. By Hall's theorem, we only need to show that any set $S \subseteq \mathcal{L}$ has $|N_{\mathcal{G}_{base}}(S)| \geq |S|$. This is true for any set S of $\leq m/2$ as by property (iii), the minimum degree of any vertex in S is already $2m/3 > |S|$. For any set S of size $\geq m/2$, by property (v), there are at least $|S|m - m/4$ edges going out of S . As each vertex in \mathcal{R} can have at most $|S|$ edges to S , there should be at least m neighbors for S in \mathcal{R} as otherwise the total number of edges between the two sets would be $(m-1) \cdot |S| < |S|m - m/4$. As such, by Hall's theorem, \mathcal{G}_{base} has a \mathcal{L} -perfect matching. \square

It is worth pointing out that for [Claim 18](#) to hold, we need the RHS of property (v) to be strictly larger than $|S|m - m$ (which is indeed true as it is $|S|m - m/4$). This is a necessary condition as otherwise \mathcal{G}_{base} maybe a bipartite clique between \mathcal{L} and $m-1$ vertices in \mathcal{R} hence not having any \mathcal{L} -perfect matching.

The very last step of the proof is now to show that random subgraphs of \mathcal{G}_{base} (or rather any graph satisfying properties (i) to (v)) also have a perfect matching with high probability. This can be seen as some form of generalization of typical random graph theory arguments. In (truly) random graphs, we work with random subgraphs of a clique (or a bipartite clique)—here instead, our “base” graph \mathcal{G}_{base} from which the sampling is done is not a clique any more (and in fact can have up to $\Omega(m^2)$ edges missing from a (lopsided) bipartite clique). As such, the argument needs to take care of these differences explicitly.

Lemma 19. *The subgraph \mathcal{G} of \mathcal{G}_{base} obtained by sampling each edge of \mathcal{G}_{base} independently and with probability $p := \frac{100 \ln n}{m}$ (corresponding to (v, c) pairs where $c \in L(c)$ and since $m = \Theta(\Delta)$), with high probability has a \mathcal{L} -perfect matching.*

Proof. Similar to the proof of [Lemma 5](#) (and [Claim 18](#)), we only need to show that for every set $S \subseteq \mathcal{L}$, $|N_{\mathcal{G}}(S)| \geq |S|$. We prove this in the following by considering two different cases based on the size of S .

¹¹Notice that the guarantee provided by this property is much stronger than applying property (ii) to each vertex of S and this is crucial for the proof; see [Claim 18](#) for the importance of this property.

Case 1: When S is “small”, i.e., $|S| \leq m/2$. In this case, we can simply use the fact that minimum degree of \mathcal{L} -vertices in \mathcal{G}_{base} is quite large to argue that $|N_{\mathcal{G}}(S)|$ should be large also. The proof of this part is almost identical to that of [Lemma 5](#). Let us formalize this as follows.

Fix a set $S \subseteq \mathcal{L}$ of size at most $m/2$ and a set $T \subseteq \mathcal{R}$ of size $|S| - 1$. By property (iii) on the min-degree of vertices in $S \subseteq \mathcal{L}$, there are at least $|S| \cdot (2m/3 - |T|) \geq |S| \cdot m/6$ edges that are going from S to outside of T in \mathcal{G}_{base} . As each of these edges appears in \mathcal{G} with probability $p = 100 \cdot \ln n/m$ independently, we have,

$$\Pr(N_{\mathcal{G}}(S) \subseteq T) \leq (1-p)^{|S| \cdot (m/6)} \leq \exp(-|S| \cdot (m/6) \cdot 100 \cdot \ln n/m) \leq n^{-10|S|}.$$

We can now do a union bound over all choices of S, T and have that,

$$\Pr(\text{there is a set } S \subseteq \mathcal{L} \text{ of size } \leq m/2 \text{ with } |N_{\mathcal{G}}(S)| < |S|) \leq \sum_{k=1}^{m/2} \binom{m}{k}^2 \cdot n^{-10k} \leq n^{-9},$$

as $m \leq n$. As such, we proved that there is no Hall’s theorem witness set S with $|S| \leq m/2$.

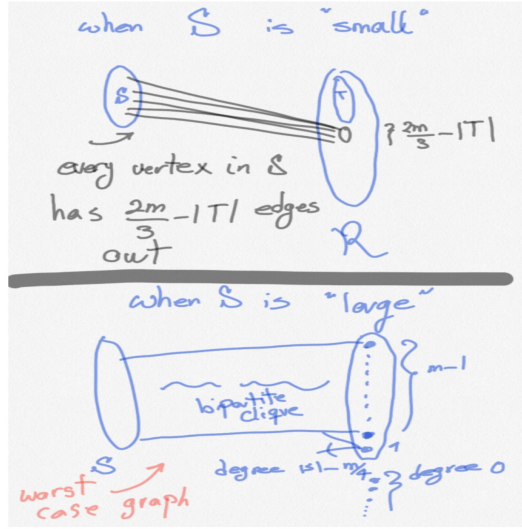


Figure 9: An illustration of the two cases in the proof of [Lemma 19](#): the first case is handled by the simple bound on the minimum degree in property (iii) and the second case is handled by considering the “worst case” graph based on the property (v).

Case 2: When S is “large”, i.e., $|S| > m/2$. This is the main part of the proof. For this case, we can no longer only rely on the property (ii) of \mathcal{G}_{base} and instead have to use property (v) crucially.

For the rest of the proof, let us set up the following notation. Fix a set S of size $\ell \geq m/2$. By property (v) of \mathcal{G}_{base} , there are at least $\ell \cdot m - m/4$ edges incident on S . Let us define the following:

- We pick a exactly $\ell \cdot m - m/4$ edges incident on S and denote them by E_S ;
- For any vertex $c_i \in \mathcal{R}$, we define d_i as the degree of c_i in E_S ;
- For any vertex $c_i \in \mathcal{R}$, we define an indicator random variable X_i where $X_i = 1$ iff c_i does not belong to $N_{\mathcal{G}}(S)$ and otherwise $X_i = 0$. Finally, define $X := \sum_i X_i$.

Note that by this definition X denotes the number of vertices in \mathcal{R} not in $N_{\mathcal{G}}(S)$ and so $|N_{\mathcal{G}}(S)| = r - X$. As such, we are interested in bounding the probability:

$$\Pr(|N_{\mathcal{G}}(S)| < |S|) = \Pr(X > r - \ell). \quad (5)$$

Moreover, by their definition and the definition of \mathcal{G} as a random subgraph of \mathcal{G}_{base} , the random variables X_1, \dots, X_r and corresponding d_1, \dots, d_r satisfy the following properties:

$$\begin{aligned} & X_1, \dots, X_r \text{ are independent;} \\ & \text{for } i \in [r] \quad \Pr(X_i = 1) = (1 - p)^{d_i}; \\ & \text{for } i \in [r] \quad d_i \leq \ell; \\ & \text{and} \quad \sum_{i=1}^r d_i = (\ell \cdot m) - m/4. \end{aligned} \tag{6}$$

It turns out bounding these probabilities directly is rather hard. Instead, in the following, we first show the “worst case” possible values for these variables (corresponding to a natural worst case graph) and then bound the probability of Eq (5) under these variables. Formally,

Claim 20. *Consider the following program on independent random variables Y_1, \dots, Y_r :*

$$\begin{aligned} & \text{maximize}_{\{z_i\}_{i=1}^r} \Pr \left(Y := \sum_i Y_i > r - \ell \right) \quad \text{subject to} \\ & \text{for any } i \in [r] \quad \Pr(Y_i = 1) = (1 - p)^{z_i} \quad \text{and} \quad 0 \leq z_i \leq \ell; \quad \sum_{i=1}^r z_i = (\ell \cdot m) - m/4; \\ & \text{(We emphasize that the only variables of this program are the integers } (z_1, \dots, z_r)). \end{aligned}$$

Then, an optimal solution is $(z_1^*, \dots, z_r^*) = (\underbrace{\ell, \ell, \dots, \ell}_{m-1}, \ell - m/4, 0, \dots, 0)$.

Before getting to the proof of this claim, let us provide some further intuition. Consider the case when the base graph \mathcal{G}_{base} is such that there are $m - 1$ vertices with degree ℓ to S , one vertex with degree $\ell - m/4$, and all other vertices in \mathcal{R} have degree zero to \mathcal{R} (such a choice satisfies Eq (6)). Then, the random variables Y_1, \dots, Y_r and variables z_1^*, \dots, z_r^* in Claim 20 would correspond to random variables X_1, \dots, X_r and d_1, \dots, d_r in Eq (6). By Eq (5) and the optimality of z_1^*, \dots, z_r^* , we obtain that $\Pr(|N_{\mathcal{G}}(S)| < |S|)$ is maximized under such a graph, turning it a worst-case graph for proving the lemma in Case 2.

Another good intuition here is that in defining this worst-case graph, we simply moved degrees of all vertices in \mathcal{R} to a *minimal* set of m vertices while satisfying the degree requirements of Eq (6); one may expect this to be the worst-case example as any deviation from this can only increase the chance of another vertex also joining $N_{\mathcal{G}}(S)$ while decreasing the chance of an original vertices to be in $N_{\mathcal{G}}(S)$ by a *lower* amount¹².

Proof of Claim 20. Consider any solution (z_1, \dots, z_r) to the program. We show that we can transform the solution one variable at a time toward (z_1^*, \dots, z_r^*) , without decreasing the value of the objective function. This will imply the optimality of (z_1^*, \dots, z_r^*) .

Without loss of generality assume $z_1 \geq z_2 \geq \dots \geq z_r$ (as the objective function is symmetric in the variables). If $(z_1, \dots, z_r) \neq (z_1^*, \dots, z_r^*)$, then there is an index i such that $z_i < z_i^*$. Consider the intermediate solution¹³:

$$(z'_1, \dots, z'_r) = (z_1, \dots, z_{i-1}, z_i + 1, z_{i+1} - 1, z_{i+2}, \dots, z_r),$$

which is a feasible solution because $z_i < z_i^* \leq \ell$ and $z_{i+1} > 0$ as $\sum_i z_i = \sum_i z_i^*$ and both sequences are sorted. We now prove that this change of variables increase the value of the objective function. The change above only affects the random variables Y_i and Y_{i+1} in the objective function. As such, we only need to prove that for any $t \in \{1, 2\}$,

$$\Pr(Y_i + Y_{i+1} = t \text{ with variables } z'_i, z'_{i+1}) \geq \Pr(Y_i + Y_{i+1} = t \text{ with variables } z_i, z_{i+1}).$$

¹²We note that all this is just for the intuition—after proving Claim 20, we can directly use the answer to upper bound Eq (6) as the variables X_1, \dots, X_r and d_1, \dots, d_r form a valid solution to the program of Claim 20.

¹³In the context of the worst-case graph intuition, here, we are moving one edge of the base graph from a lower degree vertex v_{i+1} to a higher degree vertex v_i without violating any constraint in Eq (6).

For $t = 1$, this is true because:

$$\begin{aligned}
\Pr(Y_i + Y_{i+1} = 1 \text{ with variables } z'_i, z'_{i+1}) &= (1-p)^{z'_i} + (1-p)^{z'_{i+1}} - (1-p)^{z'_i + z'_{i+1}} \\
&= (1-p)^{z_i+1} + (1-p)^{z_{i+1}-1} - (1-p)^{z_i+z_{i+1}} \\
&= (1-p) \cdot (1-p)^{z_i} + (1-p)^{-1} \cdot (1-p)^{z_{i+1}} - (1-p)^{z_i+z_{i+1}} \\
&\geq (1-p)^{z_i} + (1-p)^{z_{i+1}} - (1-p)^{z_i+z_{i+1}} \\
&\quad (\text{as } (1-p)^{z_i} \leq (1-p)^{z_{i+1}} \text{ and } \alpha \cdot x + \frac{1}{\alpha} \cdot y \geq x + y \text{ for } \alpha \leq 1 \text{ and } x \leq y) \\
&= \Pr(Y_i + Y_{i+1} = 1 \text{ with variables } z_i, z_{i+1}).
\end{aligned}$$

For $t = 2$, this is true because:

$$\begin{aligned}
\Pr(Y_i + Y_{i+1} = 2 \text{ with variables } z'_i, z'_{i+1}) &= (1-p)^{z'_i + z'_{i+1}} \\
&= (1-p)^{z_i + z_{i+1}} \\
&= \Pr(Y_i + Y_{i+1} = 2 \text{ with variables } z_i, z_{i+1}).
\end{aligned}$$

As such, the value of the objective function does not increase by this change of variables. We can thus repeatedly apply such transformation to turn the original variables into (z_1^*, \dots, z_r^*) without ever decreasing the objective value, implying the optimality of the latter solution. \square **Claim 20**

To upper bound the probability in **Eq (5)**, we can instead upper bound the probability of the objective value of the program in **Claim 20** with the assignment (z_1^*, \dots, z_r^*) . Notice that under this assignment, $Y_{m+1} = Y_{m+2} = \dots = Y_r = 1$ already as $z_{m+1}^* = \dots = z_r^* = 0$, and so there is nothing to do for those variables. As such, our goal is to bound the probability

$$\Pr\left(Y := \sum_{i=1}^r Y_i > r - \ell\right) = \Pr\left(\tilde{Y} := \sum_{i=1}^m Y_i > m - \ell\right). \quad (7)$$

Now, fix a set $T \subseteq [m]$ of size $m - \ell + 1$. For $\tilde{Y} > m - \ell$ to be true, there should exist at least one T such that $Y_i = 1$ for all $i \in T$. In the following, we bound this probability for every single T and then do a union bound on all T . By the independence of variable Y_i 's and the values of their marginals under (z_1^*, \dots, z_r^*) in **Claim 20**, we have,

$$\begin{aligned}
\Pr(\text{for all } i \in T, Y_i = 1) &= \prod_{i \in T} \Pr(Y_i = 1) = (1-p)^{\sum_{i \in T} z_i^*} \leq (1-p)^{|T| \cdot \ell - m/4} \\
&\quad (\text{as at most one } z_i^* = \ell - m/4 \text{ and the rest are } \ell) \\
&\leq \exp\left(-\frac{100 \ln n}{m} \cdot ((m - \ell + 1) \cdot \ell - m/4)\right) \\
&\quad (\text{by the choice of } p = \frac{100 \ln n}{m} \text{ and } |T| = m - \ell + 1) \\
&\leq \exp\left(-\frac{100 \ln n}{m} \cdot ((m - \ell) \cdot \ell + m/4)\right) \quad (\text{as } \ell = |S| > m/2 \text{ in Case 2}) \\
&\leq \exp(-25 \ln n \cdot (m - \ell + 1)) \quad (\text{again as } \ell > m/2) \\
&= n^{-25 \cdot (m - \ell + 1)}.
\end{aligned}$$

Applying a union bound over all choices of $T \subseteq [m]$ and **Eq (7)** implies that:

$$\Pr\left(Y := \sum_{i=1}^r Y_i > r - \ell\right) \leq \binom{m}{m - \ell + 1} \cdot n^{-25 \cdot (m - \ell + 1)} \leq n^{-24 \cdot (m - \ell + 1)},$$

as $m \leq n$. By Plugging in this inside **Eq (5)** (by the worst-case guarantee provided by **Claim 20**), we obtain

$$\Pr(|N_{\mathcal{G}}(S)| < |S|) \leq n^{-24 \cdot (m - |S| + 1)}.$$

Finally, we can do a union bound over all sets $S \subseteq \mathcal{L}$ of size $\geq m/2$ and obtain that

$$\begin{aligned} \Pr(\text{there is a set } S \subseteq \mathcal{L} \text{ of size } \geq m/2 \text{ with } |N_{\mathcal{G}}(S)| < |S|) &\leq \sum_{k=m/2}^m \binom{m}{m-k} \cdot n^{-24 \cdot (m-k+1)} \\ &\leq \sum_{k=m/2}^m n^{(m-k)} \cdot n^{-24 \cdot (m-k+1)} \quad (\text{as } m \leq n) \\ &\leq n^{-20}. \end{aligned}$$

As such in Case 2 also there is no Hall's theorem witness set in \mathcal{G} with high probability. This implies that \mathcal{G} has a \mathcal{L} -perfect matching with high probability, concluding the proof of the lemma. \square

By [Lemma 19](#), there is a \mathcal{L} -perfect matching \mathcal{M} in \mathcal{G} with high probability and by the definition of the graph \mathcal{G} , we can use this matching to color any vertex $v \in \text{Remain}(C)$ with the color $c = \mathcal{M}(v)$ which belongs to $L^*(v)$ (by the definition of \mathcal{G}) and also has not appeared in the neighborhood of v outside C (by the definition of \mathcal{G}_{base}) (and as \mathcal{M} is a matching, c also does not appear elsewhere in C). This concludes Part 2-(B) of coloring almost-cliques and consequently the entire proof of [Theorem 7](#).

Remark. Throughout the proof of [Theorem 7](#), we introduced several algorithms. However, we should emphasize that all these algorithms are for the purpose of proving the existence of the list-coloring of the graph G and not necessarily an algorithm we actually run on the input. In particular, these algorithm all assume the knowledge of the decomposition of G and also (at least naively) the edges of G as well, an information that is not available to the streaming algorithm. Nonetheless, the streaming algorithm simply assumes the existence of the list-coloring and at the end can use any algorithm, say a naive exponential-time algorithm to find the coloring. That being said, the algorithms discussed plus some additional ideas (including how to also find an “approximate” decomposition) are used in [\[2\]](#) to find the list-coloring of G efficiently.

References

- [1] N. Alon and S. Assadi. Palette sparsification beyond $(\Delta+1)$ vertex coloring. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2020, August 17-19, 2020, Virtual Conference*, pages 6:1–6:22, 2020. [6](#)
- [2] S. Assadi, Y. Chen, and S. Khanna. Sublinear algorithms for $(\Delta + 1)$ vertex coloring. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 767–786, 2019. [1](#), [2](#), [6](#), [9](#), [23](#)
- [3] B. Bollobás. *Random graphs*. Number 73. Cambridge university press, 2001. [5](#)
- [4] D. G. Harris, J. Schneider, and H.-H. Su. Distributed $(\Delta + 1)$ -coloring in sublogarithmic rounds. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 465–478. ACM, 2016. [9](#)
- [5] M. Molloy and B. Reed. A bound on the strong chromatic index of a graph. *Journal of Combinatorial Theory, Series B*, 69(2):103–109, 1997. [9](#)
- [6] M. Molloy and B. Reed. **Graph coloring and the probabilistic method**. *New York I Springer*, 23:1329–356, 2002. [7](#), [11](#)
- [7] B. Reed. ω , δ , and χ . *Journal of Graph Theory*, 27(4):177–212, 1998. [7](#)