Randomized $(\Delta + 1)$-Coloring in $O(\log^* \Delta)$ Congested Clique Rounds

Merav Parter
Weizmann IS, Rehovot, Israel
merav.parter@weizmann.ac.il

Hsin-Hao Su
UNC-Charlotte, North Carolina, USA
hsinhao@csail.mit.edu

Abstract

$(\Delta + 1)$-vertex coloring is one of the most fundamental symmetry breaking graph problems, receiving tremendous amount of attention over the last decades. We consider the congested clique model where in each round, every pair of vertices can exchange $O(\log n)$ bits of information.

In a recent breakthrough, Yi-Jun Chang, Wenzheng Li, and Seth Pettie [CLP-STOC'18] presented a randomized $(\Delta + 1)$-list coloring algorithm in the LOCAL model that works in $O(\log^* n + \mathsf{Det}_{\deg}(\log \log n))$ rounds, where $\mathsf{Det}_{\deg}(n')$ is the deterministic LOCAL complexity of $(\deg+1)$-list coloring algorithm on n'-vertex graphs. Unfortunately, the CLP algorithm uses large messages and hence cannot be efficiently implemented in the congested clique model when the maximum degree Δ is large (in particular, when $\Delta = \omega(\sqrt{n})$).

Merav Parter [P-ICALP'18] recently provided a randomized $(\Delta + 1)$-coloring algorithm in $O(\log \log \Delta \cdot \log^* \Delta)$ congested clique rounds based on a careful partitioning of the input graph into almost-independent subgraphs with maximum degree $\sqrt{n}$. In this work, we significantly improve upon this result and present a randomized $(\Delta + 1)$-coloring algorithm with $O(\log^* \Delta)$ rounds, with high probability. At the heart of our algorithm is an adaptation of the CLP algorithm for coloring a subgraph with $o(n)$ vertices and maximum degree $\Omega(n^{5/8})$ in $O(\log^* \Delta)$ rounds. The approach is built upon a combination of techniques, this includes: the graph sparsification of [Parter-ICALP'18], and a palette sampling technique adopted to the CLP framework.

2012 ACM Subject Classification Theory of computation → Distributed algorithms

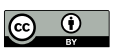
Keywords and phrases Distributed Graph Algorithms, Coloring, congested clique

Digital Object Identifier 10.4230/LIPIcs.DISC.2018.39

1 Introduction & Related Work

Graph coloring is one of the most central symmetry breaking graph problems, and as such it has been receiving much attention. In the distributed setting, vertex coloring as many other symmetry breaking tasks are mostly studied in the LOCAL model where the messages sent in a given round are allowed to be arbitrarily large. Indeed, the recent breakthrough results for vertex coloring [5, 8] and MIS [6] use large messages, potentially of size $\Omega(n)$. This poses a strong motivation for studying these problems in bandwidth restricted models.

The congested clique model of distributed computing was introduced by Lotker, Pavlov, Patt-Shamir, and Peleg [14]. In this model, the communication is all-to-all, and per round, each node can send $O(\log n)$ bits to each other node. One can view the congested clique as being orthogonal to the LOCAL model: the former abstracts away locality (each node is one-hop from each other node), and the latter abstracts away congestion . The fact that the


© Merav Parter and Hsin-Hao Su;
licensed under Creative Commons License CC-BY

congested clique model escapes any locality based lower bounds (e.g., [12]) makes it very attractive for studying the net (or clean) effect of bandwidth limitation on local computation. Ghaffari [6] posed the following question:

> *Can we solve the classic local problems: MIS, maximal matching, $(\Delta+1)$-vertex-coloring, or $(2\Delta-1)$-edge-coloring – much faster in the congested clique model?*

This question was answered in the affirmative in [6] for the MIS problem by presenting a randomized MIS algorithm that works in $\widetilde{O}(\sqrt{\log \Delta})$ congested clique rounds. Very recently, this bound was further improved to $O(\log\log\Delta)$ rounds by Ghaffari et al. [7]. The latter work also improved the round complexity of other local problems (e.g., maximal matching) in the related model of Massively Parallel Computation (MPC), which is arguably the most popular model framework for large-scale computation (e.g., MapReduce, Hadoop and Spark [7])[1]. The problem of $(\Delta+1)$-vertex coloring in the congested clique model was recently studied by [15], presenting a randomized algorithm with $O(\log\log\Delta\cdot\log^*\Delta)$ rounds. We also note that earlier works in the congested clique model considered weaker versions of MIS and coloring, see [3, 10, 9].

An orthogonal line of research considers the power of all-to-all communication for *deterministic* local algorithms. Censor et al. [4] presented a quite general scheme for derandomization in the congested clique model by combining the methods of bounded independence with efficient computation of the conditional expectation. They provided a deterministic MIS algorithm that works in $O(\log\Delta\cdot\log n)$ congested clique rounds. [15] recently showed a deterministic $(\Delta+1)$-coloring in $O(\log\Delta)$ rounds. Barenboim and Khazanov [1] presented improved deterministic local algorithms as a function of the graph's *arboricity*.

1.1 Our Result and Technical Overview

Our main result is an adaptation of the Chang-Li-Pettie (CLP) algorithm to the congested clique model:

▶ **Theorem 1.** *There is a randomized algorithm that computes a $(\Delta+1)$-coloring in $O(\log^*\Delta)$ rounds of the congested clique model, with high probability*[2]*.*

Our algorithm is based on the recent graph sparsfication technique of [15] combined with modified versions of the key coloring algorithms of [5]. The starting observation made in [15] is that the CLP algorithm can be simulated in $O(\log^*\Delta)$ congested clique rounds when $\Delta=O(\sqrt{n})$. To handle a graph with arbitrarily large degrees, [15] applies $O(\log\log\Delta)$ phases of a graph specification procedure, until all unsolved pieces to be colored are subgraphs with maximum degree $O(\sqrt{n})$ and hence can be colored in $O(\log^*\Delta)$ rounds by employing the CLP algorithm. In this work, we significantly improve upon this approach, and break this $\sqrt{n}$-barrier by modifying the key coloring procedures of the CLP algorithm. This modified CLP allows us to color $o(n)$-vertex subgraphs with maximum degree $\Omega(n^{5/8})$ in $O(\log^*\Delta)$ rounds. The high-level description of our algorithm is as follows: Given a graph G with maximum degree Δ, G is carefully partitioned into: (i) a collection of $\Delta^{1/4}$ *independent* subgraphs G_i with maximum degree $\Delta(G_i)=O(\Delta^{3/4})$ and (ii) a left-over subgraph G^* with $N=\widetilde{O}(n/\Delta^{3/8})$ vertices and maximum degree $\Delta^*=\widetilde{O}(\Delta^{5/8})$. The improvement over [15] comes from the fact that our algorithm applies only a constant number of phases of the graph sparsification procedure (for coloring the subgraphs of (i)), rather than $O(\log\log\Delta)$

[1] A general simulation result between these models has been recently provided by [2].
[2] As usual, by high probability we mean $1-1/n^c$ for some constant $c\geq 1$.

as in [15]. The first collection of G_i subgraphs are treated as independent in the sense that each subgraph G_i is given a distinct set of $\Delta(G_i)+1$ colors in $[1, \Delta+1]$ and thus these subgraphs can be colored simultaneously within $O(\log^* \Delta)$ rounds using the [15] algorithm. The partitioning into these graphs is done in a careful manner so that allocating $\Delta(G_i)+1$ colors to each of them still respects the total number $\Delta+1$ of allowed colors. The main challenge is in coloring the left-over subgraph G^* overcoming the fact that its degree is $\Omega(n^{1/2+\epsilon})$. Unlike the previous $\Delta^{1/4}$ subgraphs, here we ran out of budget of free colors and hence this subgraph should be colored using a list-coloring algorithm only after all other subgraphs G_i are *colored*. As will be described later on, the CLP algorithm is based on the knowledge of the second neighborhood of the vertices (which can be obtained in 2 rounds in the LOCAL model). In our setting, the degree of G^* is too large for allowing the vertices collecting their entire second neighborhoods. The key challenges is in bypassing all critical points of the CLP that are based on this kind of knowledge. To do that we employ several congested clique routing techniques combined with a palette sampling technique adopted to the CLP framework.

Technical History of Coloring and a Short Exposition of the CLP Algorithm. The first step for breaking the KMW lower bound [12] was made by Schneider and Wattenhofer [17] who showed that when vertices have sufficiently many excess colors in their palette[3] the graph can be colored considerably faster. Elkin, Pettie and Su made the first connection between the above observation to a concrete structural graph property. Specifically, they showed that an $(1-\epsilon)$-sparse graph[4] can be transformed within a single round into a graph in which each vertex has $\Omega(\epsilon\Delta)$ many access colors in its palette. This graph characterization was the basis of the decomposition technique by Harris, Schneider and Su [8] which we describe next.

For a given parameter $\epsilon \in (0,1)$, [8] decomposed the input graph G into an ϵ-sparse and an ϵ-dense subgraphs. To color the sparse subgraph, [8] employed the approach of [17], and their key contribution is a novel dense coloring procedure. [8] showed that the dense subgraph consists of a collection of almost-clique components with weak diameter 2. Informally, the dense coloring procedure was based on having a leader in each such almost-clique; the leader collected the palettes and neighbor-lists of all the vertices in its clique, and colored them locally such that most of these colors are legal.

The approach of [5] is based on a hierarchical version of [8] with $O(\log\log\Delta)$ sparsity levels. This partitions the vertices in the graph into $O(\log\log\Delta)$ layers. The algorithm further groups these layers into $O(\log^*\Delta)$ *Starta* where all layers in a given strata are colored simultaneously. When coloring vertices in each stratum, the algorithm applies modified versions of the dense coloring procedure in [8], which are based upon collecting the information of each almost-clique to a leader. Since the diameter of the almost-clique is shown to be 2, this information an be easily collected in the LOCAL model, but might take many rounds, when the message size is restricted to $O(\log n)$ bits.

How to Break the $\Delta = O(\sqrt{n})$ Barrier? The main obstacle for simulating the CLP algorithm when $\Delta = \omega(\sqrt{n})$ in the congested clique model concerns two critical places in the CLP algorithm, where vertices collect $O(\Delta^2)$ messages (e.g., their second neighborhood). The first place is for (1) defining the ϵ-dense subgraph and the second place is for (2) coloring the

[3] The excess of colors of vertex v is the number of colors in v's palette minus the number of uncolored neighbors of v.

[4] I.e., a graph in which in the 2-hop neighborhood of each vertex, there are only $(1-\epsilon^2)\Delta^2$ triangles.

dense regions by collecting palettes to the leader of each almost-clique. These steps could be implemented in $O(1)$ rounds only when $\Delta = O(\sqrt{n})$, but require polynomially many rounds for $\Delta = \Omega(n^{1/2+\epsilon})$. Thus breaking this $\sqrt{n}$ barrier calls for alternative procedures that avoid learning the 2-neighborhoods of the vertices. We now elaborate about these technicalities and our approach to handle them.

To compute the dense subgraph, in the CLP algorithm every vertex v computes the number of mutual neighbors with each of its neighbors, i.e., it computes $|N(v) \cap N(u)|$ for every $u \in N(v)$. Indeed this can be easily done if a vertex knows the neighbors of its neighbors. In our setting, we use the fact that the left-over subgraph G^* has $N = \widetilde{O}(n/\Delta^{3/8})$ vertices and allocate each vertex v in G^* a subset of $r = n/N$ relay vertices that share the computational load of vertex v. Specifically, in our scheme, each relay vertex of v is responsible for computing the intersection size $|N(v) \cap N(u)|$ for a subset of Δ/r neighbors $u \in N(v)$, it would then communicate the outcome of this computation to v.

The second spot in which the CLP algorithm collects the second neighborhood of (some of the vertices) is in the dense coloring procedures. Our key technical contribution is in showing that it is sufficient for each almost-clique member to send to its leader a random sample of $O(\sqrt{\Delta})$ colors in its palette rather than its *entire* $\Theta(\Delta)$-size palette. Since each almost-clique contains $O(\Delta)$ vertices and since the maximum degree of the subgraph G^* is $O(n^{2/3})$, each leader is a target of $O(n)$ message. Such a routing pattern can then be implemented $O(1)$ rounds using the routing algorithm of Lenzen. The technical challenge is in showing the even-though the leader of an almost-clique C knows neither internal edges in C nor the complete individual palettes of the vertices in C, it can still mimic the CLP procedures, and color its clique vertices with almost the same success rate.

Lenzen's Routing Algorithm. Almost all congested clique algorithms are based on the Lenzen's routing algorithm [13]. This routing algorithm schedules in $O(1)$ rounds the common communication setting where each vertex needs to send and receive $O(n)$ messages.

2 Coloring Most Vertices Through Graph Sparsfication

We make use of the following version of Chernoff bound:

▶ **Theorem 2** (Simple Corollary of Chernoff Bound). *Suppose $X_1, X_2, \ldots, X_\ell \in [0,1]$ are independent random variables, and let $X = \sum_{i=1}^{\ell} X_i$ and $\mu = \mathbb{E}[X]$. If $\mu \geq 5\log n$, then with probability at least $1 - 1/n^2$, $X \in \mu \pm \sqrt{5\mu \log n}$, and if $\mu < 5\log n$, then $X \leq \mu + 5\log n$.*

The Algorithm. The graph G is partitioned into $\ell = \lceil \Delta^{1/4} \rceil$ subgraphs $G_1, \ldots, G_\ell$, and a left-over subgraph G^*. This is done by dividing the vertices into $\ell + 1$ subsets $V_1, \ldots, V_\ell, V^*$ by letting each vertex join V_i with probability

$$p_i = 1/\ell - 2\sqrt{5\log n}/\sqrt{\Delta \cdot \ell},$$

for every $i \in \{1, \ldots, \ell\}$, and joining V^* with the remaining probability of

$$p^* = 2\sqrt{5\log n} \cdot \sqrt{\ell/\Delta} = \Theta(\log n/\Delta^{3/8}).$$

For every $i \in \{1, \ldots, \ell, *\}$, let $G_i = G[V_i]$ be the induced subgraph and let Δ_i be the maximum degree of G_i. Using Chernoff bound of Theorem 2, for every $i \in \{1, \ldots, \ell\}$, w.h.p., it holds: $\Delta_i \leq \Delta/\ell - 2\sqrt{5\log n} \cdot \sqrt{\Delta/\ell} + \sqrt{5\log n} \cdot \sqrt{\Delta/\ell} \leq \Delta/\ell - 1$.

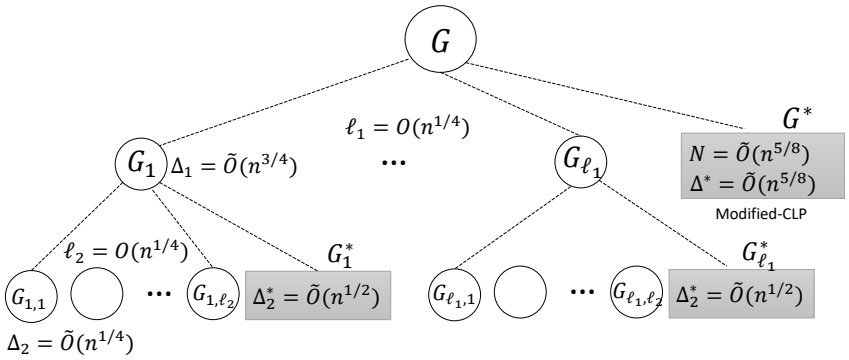


Figure 1 For ease of presentation, we omit log factors from considerations. The graph G is partitioned into $\widetilde{O}(n^{1/4})$ subgrpahs G_i and a left-over subgraph G^*. Each subgraph G_i has maximum degree $\widetilde{O}(n^{1/4})$ and it is further divided into $\widetilde{O}(n^{1/4})$ subgraphs and a left-over subgraph in [15]. The non left-over subgraphs are given independent set of colors and are colored simultaneously by applying CLP. The left-over subgrpahs are colored once all other subgraphs are colored. After all G_i graphs are colored, we apply our modified CLP algorithm to complete the coloring of G^*.

In the first phase of the coloring algorithm, all subgraphs $G_1, \ldots, G_\ell$ are colored independently and simultaneously. This is done by allocating a distinct set of $\Delta_i + 1$ colors for each of the G_i subgraphs. Overall, we allocate $\ell \cdot (\Delta/\ell) \leq \Delta$ colors. Since each $\Delta_i = O(\Delta^{3/4}) = O(n^{3/4})$, this can be done in $O(\log^* \Delta)$ rounds for all $G_1, \ldots, G_\ell$ simultaneously using the following:

▶ **Lemma 3.** *[15] There is a randomized $(\Delta+1)$-coloring algorithm in the congested clique model that works in $O(\log(1/\epsilon)\log^* \Delta)$ rounds when $\Delta = O((n/\log n)^{1-\epsilon})$ for any $\epsilon \in (0,1)$.*

The algorithm of [15] also implies that the same round complexity is obtained where one is given k vertex-disjoint subgraphs each with maximum degree $O((n/\log n)^{1-\epsilon})$.

Coloring the remaining left-over subgraph G^*. The second phase of the algorithm completes the coloring of G^*. This coloring should agree with the colors computed for $G \setminus G^*$. Hence, G^* is colored by employing a *list* coloring algorithm that we describe in the next section. We next bound the number of vertices and the maximum degree $\Delta(G^*)$ of G^* which provides the basis for our ability to list-color it efficiently in the congested clique model using our modified CLP algorithm. By Chernoff bound the following holds:

▶ **Observation 4.** *$|V(G^*)| = O(n \log n/\Delta^{3/8})$ and $\Delta(G^*) = O(\Delta^{5/8} \cdot \log n)$.*

For an illustration of our algorithm, see Figure 1.

3 List-Coloring of the Remaining Subgraph

Recall that the input graph G has n vertices and maximum degree Δ. At the heart of our coloring algorithm is a list-coloring procedure that colors a subgraph $G^* \subseteq G$ with bounded number of vertices $N \leq n$ vertices and bounded maximum degree Δ^*. For ease of presentation, we first assume that each vertex $v \in G^*$ is given a palette of size $\Delta^* + 1$. At the end of the section, we explain the needed adaptation for the case where every vertex $v \in G^*$ has a palette with at least $\max\{\deg(v, G^*) + 1, \Delta^* - (\Delta^*)^{3/5}\}$ colors.

▶ **Theorem 5.** *Given a subgraph $G^* \subseteq G$ with N vertices and maximum degree Δ^*, such that each vertex $v \in G^*$ has a palette of size Δ^*+1. If Δ^* satisfies*

(I) $\Delta^* \in [\sqrt{n}, O(n^{2/3})]$, **(II)** $\Delta^* = O(n/\sqrt{N})$, **(III)** $\Delta^* = O(n^2/N^2)$.

then G^* can be coloring in $O(\log^* \Delta)$ rounds, with high probability.

At the of the section, Lemma 17, we show that the remaining subgraph from the previous section indeed satisfies properties (I-III).

Key Definitions from the CLP Algorithm. For an $\epsilon \in (0,1)$, an edge $e=(u,v)$ is an ϵ-friend if $|N(u) \cap N(v)| \geq (1-\epsilon)\cdot\Delta^*$. The endpoints of an ϵ-friend edge are ϵ-friends. A vertex v is ϵ-dense if v has at least $(1-\epsilon)\cdot\Delta^*$ ϵ-friends; otherwise it is ϵ-sparse.

Given a subset of vertices $\widetilde{V} \subseteq V(G^*)$ (which will be the set of uncolored vertices after the preliminary OneShotColoring algorithm), we define a partition of $\widetilde{V}$ into layers $(V_1, \ldots, V_\ell, V_{sp})$ for $\ell = O(\log\log\Delta)$ based on the local sparsity. Let $(\epsilon_1, \ldots, \epsilon_\ell)$ be the sequence of sparsity parameters where $\epsilon_1 = (\Delta^*)^{-1/10}$, $\epsilon_i = \sqrt{\epsilon_{i-1}}$ for $i \in [2, \ell-1]$, and $\epsilon_\ell = 1/K$ for a large enough constant K. For a sparsity parameter ϵ_i, let $V^d_{\epsilon_i}, V^s_{\epsilon_i}$ be the set of vertices which are ϵ_i-dense (resp., ϵ_i-sparse). This defines a hierarchy of ℓ *layers*: $V_1, \ldots, V_\ell$ where $V_1 = \widetilde{V} \cap V^d_{\epsilon_1}$, $V_i = \widetilde{V} \cap (V^d_{\epsilon_i} \setminus V^d_{\epsilon_{i-1}})$ and $V_{sp} = \widetilde{V} \cap V^s_{\epsilon_\ell}$.

The ϵ_i-dense vertices $V^d_{\epsilon_i}$ are then partitioned into ϵ_i-*almost* cliques for every ϵ_i. The ϵ_i-*almost* cliques are the connected components of the graph induced on $V^d_{\epsilon_i}$ and the ϵ_i-friend edges incident to these vertices. The following lemma developed in [8] contains some important properties of ϵ-almost cliques.

▶ **Lemma 6.** *Fix any $\epsilon < 1/5$. The following conditions are met for each ϵ-almost clique C, and each vertex $v \in C$. (i) The external degree $|N(v) \cup (V^d_\epsilon \setminus C)| \leq \epsilon\Delta^*$ (ii) The anti-degree $|C \setminus (N(v) \cup \{v\})| \leq 3\epsilon\Delta^*$, (iii) $|C| \leq (1+3\epsilon\Delta^*)$, and (iv) $\mathrm{dist}_G(u,v) \leq 2$ for each $u, v \in C$, i.e., C has weak diameter at most 2.*

For $i \in [1, \ell]$, each layer V_i is further partitioned into *blocks* as follows. Let $\{C_1, C_2, \ldots,\}$ be ϵ_i-almost cliques, then each clique C_j defines a block $B_j = C_j \cap V_i$, that is the block B_j contains the subset of vertices in C_j that are ϵ_i-dense but are *not* ϵ_{i-1}-dense. Note that all the blocks form a partition of $\widetilde{V}$.

An Outline of CLP Algorithm. The sketch of a slightly *modified* version of CLP algorithm is outlined in the following. The pseudocodes for the subroutines of [5] are provided in Appendix A.

1. Execute OneShotColoring which takes $O(1)$ rounds. Let $\widetilde{V}$ be the set of uncolored vertices. If $v \in \widetilde{V}$ is in layer i for $i \in [2, \ell]$, v has $\Omega(\epsilon^2_{i-1}\Delta^*)$ excess colors. If $v \in \widetilde{V}$ is in V_{sp}, then it has $\Omega(\epsilon^2_\ell\Delta^*) = \Omega(\Delta^*)$ excess colors.
2. Execute the Dense Coloring Procedure, which takes $O(\log^* \Delta)$ rounds. For every $v \in \widetilde{V}$, the number of uncolored its uncolored neighbors in layer i will be bounded by $O(\epsilon^5_i\Delta^*)$ for $i \in [2, \ell]$. All vertices in layer 1 become colored.
3. After Step 1 and Step 2, for every vertex $v \in \widetilde{V}$, if v is in layer 1, then it will be colored. If v is in layer i for $i \in [2, \ell]$, then it has $\Omega(\epsilon^2_{i-1}\Delta^*)$ excess colors in the palette. Moreover, the number of neighbors of $v \in V_i$ with lower or equal layer is at most $O(\sum_{j=1}^{i} \epsilon^5_j\Delta^*) = O(\epsilon^5_i\Delta^*) = O(\epsilon^{2.5}_{i-1}\Delta^*)$. Since the number of competing neighbors is significantly less than the number of excess colors (i.e. $\epsilon^{2.5}_{i-1}\Delta^* \ll \epsilon^2_{i-1}\Delta^*$), we can color the remaining vertices very efficiently by using the ColorBidding algorithm (c.f. Appendix A) in $O(\log^* \Delta)$ rounds.

Adaptation of CLP to the Congested Clique. Our key contribution is in adopting the above CLP algorithm for coloring G^*, which might have maximum degree $\Omega(n^{5/8})$. For **Step 1**, Alg. OneShotColoring can be trivially implemented in $O(1)$ rounds in the congested clique model, since each vertex is only sending one selected color to its neighbors.

The main challenge lies in **Step 2**. The basic idea of the dense coloring procedure is the following. Since the weak diameter of each block is 2, in the LOCAL model, it is possible for a leader in the block to collect the edges within the block and the palette of each vertex in the block. Then the leader assigns a random proper coloring to each vertex in the block. Since there are no internal conflicts in the block and the external number of neighbors is small for each vertex (i.e., $O(\epsilon_i\Delta^*)$ for a layer-i block), the probability that a vertex is assigned the same color as any of its external neighbors is small ($\mathsf{poly}(\epsilon_i)$ if the vertex in layer i). Intuitively, after $O(1)$ iterations, the probability that the vertex remains uncolored is $O(\epsilon_i^5)$. Therefore, it is plausible that for a given set of layer-i vertices, the number of uncolored vertices is bounded $O(\epsilon_i^5\Delta^*)$. However, this could have problems if a block is too small. In this case, the palette size of each vertex may also be small. The probability a layer-i vertex remains uncolored may no longer be $\mathsf{poly}(\epsilon_i)$. To deal with this issue, [5] groups the blocks into $O(\log^*\Delta)$ strata. They showed that by coloring the strata in the right order, the palette of the vertices will be large enough at the time the procedure is executed. We describe it later in detail in the section **Coloring Vertices by Stratum**.

When it comes to the congested clique model, there are two obstacles. First, in the dense coloring procedure, each vertex has to know which layer and block it is in. For example, we need to compute the ϵ-friends of each vertex *without* collecting its 2-hop neighborhoods. We show this can be achieved by using the idle vertices in $G \setminus G^*$ as relaying vertices. The second obstacle is that in the congested clique, given a block B, we do not have the capacity to let each vertex in B send its incident edges and palette to the leader of B. Instead, we show that it will be sufficient if every vertex in B sends $\sqrt{\Delta^*}$ independently sampled colors from its palette to the leader rather than the whole palette. Moreover, each vertex does not have to send the incident edges to the leader. This will be within our budget since $|B| = O(\Delta^*)$ each leader receives $O(\sqrt{\Delta^*} \cdot \Delta^*) = O(n)$ messages. The colors can be routed by using Lenzen's algorithm. We show such a modification have negligible effects when we color vertices by stratum.

For **Step 3**, in each iteration of ColorBidding algorithm, each vertex v sends a set of colors S_v to all its neighbors. The size of S_v is $\tilde{O}((\Delta^*)^{1/5})$. Since every vertex sends and receives $\tilde{O}((\Delta^*)^{5/4}) = O(n)$ messages (by Property (I)), this can be implemented in $O(1)$ rounds using Lenzen's routing algorithm.

Computation of ϵ-Friends. We describe an $O(1)$-round procedure that allows each vertex v to compute its ϵ_i-friends for each of the $\ell = O(\log\log\Delta)$ sparsity values $\epsilon_1, \ldots \epsilon_\ell$. Using this information, v would be able to compute its minimum sparsity parameter ϵ_i such that v is ϵ_i-dense (but ϵ_{i-1}-sparse). A trivial way to compute the ϵ-friends of each vertex v is by collecting the neighbor-list of the v's neighbors. Since this information contains $\Omega((\Delta^*)^2)$ messages, it cannot be done in $O(1)$ rounds when $\Delta^* = \omega(\sqrt{n})$. Instead, we use the fact that G^* has only N vertices and allocate to each vertex $v \in G^*$, a collection of n/N relay vertices $R(v)$. These relay vertices would help v in the computation of its ϵ-friends. Towards that end, each vertex v first sends the IDs of its Δ^* neighbors to each of its relay vertices $R(v)$. Hence, overall v sends $\Delta^* \cdot |R(v)| = O(n)$ messages. At this point, the relay vertices $R(v)$ of each vertex v know the neighbor list of their designated vertex v. Next, v partitions the "responsibilities" for its Δ^* neighbors among its $R(v)$ vertices. Formally, for $r \in R(v)$, let

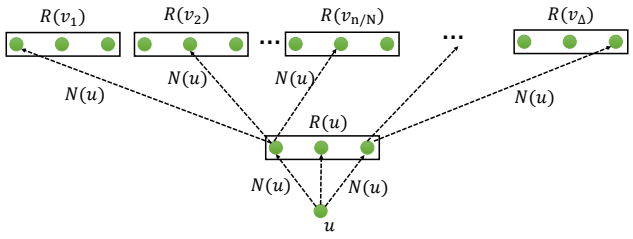


Figure 2 Illustration of ϵ-Friends Computation via Relay Vertices. Verex u sends its neighbor-list $N(u)$ to each $r \in R(u)$. Each $r \in R(u)$ sends $N(u)$ to the corresponding relay vertices of $\Delta^*/|R(u)|$ neighbors in $N(u)$.

$N(v,r) \subseteq N(v)$ be the set of $\Delta^*/|R(v)|$ v's neighbors assigned to r, where $\cup_r N(v,r) = N(v)$. Each $r \in R(v)$ will receive the neighbor list of each vertex $u \in N(v,r)$. In total, each relay vertex $r \in R(v)$ would be a target of $\Delta^* \cdot |N(v,r)| = O((\Delta^*)^2 \cdot N/n) = O(n)$ messages, where the last bound holds due to property (II). To send the neighbor-list to the corresponding relay vertex, each vertex u uses its relay vertices $R(u)$ again. Specifically, for each of its neighbors $v_i \in N(u)$, u knows the ID of the relay vertex to which its neighbor list $N(u)$ should be sent. It then partitions the responsibilities among its relay vertices by assigning $\Delta^*/|R(u)|$ neighbors to each $r' \in N(u)$. The relay $r' \in N(u)$ sends $N(u)$ to the corresponding $\Delta^*/|R(u)|$ relay vertices. Overall, each relay vertex sends $O((\Delta^*)^2/|R(u)|) = O(n)$ messages (by Property (II)). For an illustration see Figure 2. Since each vertex is a source and a target of $O(n)$ messages, this computation can be done in $O(1)$ rounds by using Lenzen's routing algorithm. Each relay node $r \in R(v)$ now holds the neighbor-list of $\Delta^*/|R(v)|$ neighbors of v as well as the neighbor-list of v. This allows r to compute the intersection size between $N(v)$ and $N(u)$ for every $u \in N(v,r)$. More specifically, for each neighbor u of v, the relay node in $R(v)$ responsible for u will send to v the minimum ϵ' value such that u and v are ϵ'-friends. Since v should receive $O(\Delta^*)$ messages and each relay vertex $r \in R(v)$ sends $O(\Delta^*/|R(v)|)$ messages, this can be done in $O(1)$ rounds using Lenzen's algorithm.

Computation of Almost-Cliques and Block Partition. Now each node v knows its ϵ_i-friends for every sparsity level ϵ_i for $i \in [1,\ell]$. It also knows which layer it is in. The next step is for v to know which block it is in.

To achieve this, we first compute ϵ_i-*almost* cliques for each $i \in [1,\ell]$, where $\ell = O(\log\log\Delta)$. Recall that an ϵ_i-almost clique is a component in the subgraph induced by ϵ_i-friend edges and vertices in $V^d_{\epsilon_i}$. For $i \in [1,\ell]$, let C_{v,ϵ_i} be the ϵ_i-almost clique that contains v. Note that $C_{v,\epsilon_i} = \emptyset$ if i is lower than the layer of v. For a specific level i, by using the $O(1)$-round connectivity-identification algorithm of [11], each vertex v is able to learn the ID of every vertex in C_{v,ϵ_i}

However, we cannot afford to apply the connectivity-identification algorithm in a serial manner for each sparsity class. Instead, we will use again n/N relay vertices assigned to each vertex in G^*. This is done as follows. For each $v \in G^*$, we allocate a relay vertex u_i that "plays" the role of v in the i^{th} application of the connectivity-identification algorithm, i.e., for computing ϵ_i-*almost* cliques for every $i \in \{1,\ldots,O(\log\log\Delta)\}$. Since $n/N = \Omega(\log\log\Delta)$, this is within our budget. By letting v sending its Δ^* vertices to each relay vertex u_i, the latter have all the information needed to run the connectivity algorithm on behalf of v. This allows us to apply the $O(\log\log\Delta)$ connectivity-identification algorithms *simultaneously*. At the end of this computation, each relay node of v sends to v the ID of every vertex in its connected component. This is possible since the size of each almost-clique is $O(\Delta^*)$ (by

Lemma 6) and $O(\Delta^* \cdot \log\log\Delta) = O(n)$ (by Property (I)). This allows each vertex $v \in G^*$ to learn all the vertices in each almost clique $C_{v,\epsilon_1}, \ldots, C_{v,\epsilon_\ell}$. Note that we are computing more information than we need here, but it will be used later.

We are now ready to compute the partitioning of the cliques into blocks. Suppose that a vertex v is in layer $i(v)$. The block containing v, B_v, was defined as $(C_{v,\epsilon_{i(v)}} \cap V_{i(v)})$. Vertex v can identify all the members in B_v if it knows the layer of every vertex in G^*. This information can be obtained by every vertex in one round in the congested clique, since every vertex already knows its layer.

Coloring Vertices by Stratum. To ensure that the palettes of the vertices in each block contain enough color in the execution of the dense coloring procedure, the CLP algorithm groups the ℓ layers into $s = O(\log^* \Delta)$ *Strata* $W_1, \ldots, W_s$ where $W_1 = V_1$ and

$$W_k = \bigcup_{i:\epsilon_i \in (\xi_{k-1}, \xi_k]} V_i \quad \text{where } \xi_1 = \epsilon_1 \quad \text{and} \quad \xi_k = 1/\log(1/\xi_{k-1}) \text{ for } k \in [2, s].$$

Each vertex can easily determine the stratum of a vertex by its layer. The blocks are also divided into two categories, *large* blocks and *small* blocks. We say a block B is *good* if $|B| \geq \Delta^*/\log^2(1/\xi_k)$, where k is the stratum that B lies at. A vertex v determines whether its block B_v is large or small by the following criteria. Let $i(v)$ denote the layer of v. If B_v is good and none of the following blocks $(C_{v,\epsilon_{i(v)+1}} \cap V_{i(v)+1}), (C_{v,\epsilon_{i(v)+2}} \cap V_{i(v)+2}), \ldots,$ or $(C_{v,\epsilon_\ell} \cap V_\ell)$ are good, then B_v is a large block. Otherwise, B_v is a small block. Since each vertex v knows the vertices in each almost clique $C_{v,\epsilon_1}, \ldots, C_{v,\epsilon_\ell}$ and v knows the layers and the stratum of all other vertices, whether B_v is small or large can be determined locally.

Define W_k^S and W_k^L be the set of all vertices in stratum-k small blocks and stratum-k large blocks. We have that $\widetilde{V} = (W_1^S, \ldots, W_s^S, W_1^L, \ldots, W_s^L, V_{sp})$. The vertices are colored in $s+2$ stages. First, all the small blocks are colored in s phases: stratum by stratum. In other words, all the vertices in the small block are colored according to the order: $W_s^S, \ldots, W_1^S$. Next, the algorithm colors the vertices of $W' = \bigcup_{j=2}^{s} W_j^L$, i.e., all the vertices in large blocks, except for those belonging to blocks of the first layer W_1^L. Lastly, the vertices of the large blocks in W_1^L are colored. At the end, the the (small) subset of vertices that failed to be colored and the sparse vertices V_{sp} are colored in Step 3.

Suppose that we process the blocks according to the stratum in the order described above. In [5], they showed a crucial property that when we are coloring a small block in W_k^S, each vertex has $\Delta^*/2\log^2(1/\xi_k)$ *excess* colors (i.e. each vertex v has at least $|N(v) \cap W_k^S| + \Delta^*/2\log^2(1/\xi_k)$ colors) in its palette. When we process a large block, since the block is large, each vertex has at least $\Delta^*/2\log^2(1/\xi_k)$ colors in the palette.

The CLP algorithm consists of two different versions of dense coloring procedures, according to whether the blocks being processed are small or large. In a high level way, the differences are the following: In large blocks we do not have the excess colors. However, all blocks belongs to different almost-cliques by the definition of large. In small blocks, when we are processing blocks in W_k^S, it is possible that some blocks belong to the same almost-clique. However, since there are abundant number of excess colors, this allow us to process all these blocks together using a single leader. The *superblocks* are defined for this purpose. Consider W_k^S and suppose that stratum-k spans layer $i_0, \ldots i_1$. Let $\{C_1, C_2 \ldots\}$ be ϵ_{i_1} almost cliques. Each C_j defines a superblock $R_j = C_j \cap W_k^S$. Therefore, $(R_1, R_2, \ldots)$ is a partition of W_k^S. Each vertex v can easily identify the members in its superblock by using the same approach we described for computing the blocks.

Version 1 – Dense Coloring Procedure for Strata of Small Blocks. Consider the set of stratum-k *small blocks* W_k^{S}. Recall that if stratum-k spans layers $i_0, i_0+1, \ldots, i_1$, then a super-block is a maximal almost-ϵ_{i_1} clique induced in W_k^{S}. Let $S = W_k^{\mathsf{S}}$ and $S_1, \ldots, S_g$ be the super-blocks.

Given vertices $u, v \in S$, we say that u has a *higher priority* than v if (1) the layer of u is lower than that of v, or (2) u and v are in the same layer but $ID(u) < ID(v)$. The CLP algorithm for coloring each super-block S_j works as follows. Let $\pi : \{1, \ldots, |S_j|\} \to S_j$ be a permutation ordered by the priority of the vertices, from the highest priority to the lowest. Now a leader processes each vertex $\pi(1), \pi(2), \ldots \pi(|S_j|)$. For vertex $\pi(q)$, it selects a color randomly from its palette excluding the colors used by its neighbors in $\pi(1), \ldots \pi(q-1)$. Also, CLP showed that in small blocks, each vertex has $Z_{ex} = \Delta^*/2\log^2(1/\xi_k)$ excess colors. Let $N'(v)$ denote the higher priority neighbors of v. Suppose that v is in layer i, we must have $|N'(v) \cap (S \setminus S_j)| \leq \epsilon_i \Delta^*$. Therefore, when the vertex is being processed, the probability that it has an external conflict is at most $(\epsilon_i \Delta^*)/(Z_{ex}) \leq 2\epsilon_i \log^2(1/\xi_k) \leq 2\epsilon_i \log^2(1/\epsilon_i)$.

Since we cannot afford each vertex to send the whole palette and its incident edges to the leader in the congested clique model, we let each vertex randomly sample $\sqrt{\Delta^*}$ colors and send them to the leader. Since $|S_j| = O(\Delta^*)$, each leader is receiving at most $\sqrt{\Delta^*} \cdot O(\Delta^*) = O(n)$ messages (by Property (I)). Thus, the set of colors can be routed to the leader by using Lenzen's routing algorithm. The following is the description of our algorithm.

ModifiedSmallDenseColoring (Modified Alg. of DenseColoringStep, version 1 from [5]).

1. Consider a superblock S_j. Let π be the permutation of S_j ordered by the priority of the vertices. Each vertex $\pi(q)$ sends a set $C(\pi(q))$ of $\sqrt{\Delta^*}$ colors selected u.a.r. from its palette to the leader. For each selected vertex $\pi(q)$, if $C(\pi(q)) \setminus \{c(\pi(1)), \ldots, c(\pi(q-1))\}$ is non-empty, then the leader assigns $\pi(q)$ a color $c(\pi(q))$ randomly selected from the set. Otherwise, we say that $\pi(q)$ is *skipped*.
2. Each $v \in S_j$ that has selected a color $c(v)$ permanently color itself $c(v)$, if $c(v)$ is not selected by any vertices $u \in N'(v)$. Otherwise, we say that v is *decolored*.

▶ **Lemma 7.** *Let D_v denotes an upper bound on the external higher priority neighbors (i.e. $|N'(v) \setminus (S \setminus S_j)|$). Let $\delta_v = 2D_v/Z_{ex}$. The probability that $v = \pi(q)$ becomes decolored is at most δ_v, conditioned on any choices of $\pi(1), \pi(2), \ldots, \pi(q-1)$ and all the other higher priority vertices in $S \setminus S_j$.*

Proof. Consider a vertex $\pi(q)$. Since the anti-degree of $\pi(q)$ is at most $3\epsilon_{i_1}\Delta$, at most $3\epsilon_{i_1}\Delta$ vertices in S_j can be non-neighbors of $\pi(q)$. Moreover, $\pi(q)$ has at least Z_{ex} uncolored neighbors outside of S. Therefore, $|\mathsf{Pal}(\pi(q)) \setminus \{c(\pi(1)), \ldots, c(\pi(q-1))\}| \geq Z_{ex} - 3\epsilon_{i_1}\Delta^* \geq Z_{ex}/2$. Consider any assignment of colors $c(\pi(1)), \ldots, c(\pi(q-1))$. Since v selects the colors randomly, any color that is not $c(\pi(1)), \ldots,$ or $c(\pi(q-1))$ has the same probability to be assigned as $c(\pi(q))$. Therefore, the probability that $c(\pi(q))$ is a specific color is at most $1/|\mathsf{Pal}(\pi(q)) \setminus \{c(\pi(1)), \ldots, c(\pi(q-1))\}| \leq 2/Z_{ex}$. Now consider any choices made by $\pi(1), \pi(2), \ldots, \pi(q-1)$ and any choices made by external neighbors in $N'(\pi(q)) \cap (S \setminus S_j)$. Vertices $N'(\pi(q)) \cap (S \setminus S_j)$ are assigned with at most D_v different colors. Vertex v can only become decolored only if $c(\pi(q))$ is one of the colors selected by the vertices in $N'(\pi(q)) \cap (S \setminus S_j)$. Therefore, the probability that $\pi(q)$ is decolored is at most $2D_v/Z_{ex}$. ◀

▶ **Lemma 8.** *In* ModifiedSmallDenseColoring*, with probability at least* $\exp(-\Omega(\mathsf{poly}(\Delta^*)))$*, no vertices are skipped.*

Proof. Recall that S_j is a superblock in stratum k, which spans layer $i_0, \ldots, i_1$. Consider a vertex $\pi(q)$. Since the anti-degree of $\pi(q)$ is at most $3\epsilon_{i_1}\Delta$, at most $3\epsilon_{i_1}\Delta$ vertices in S_j can

be non-neighbors of $\pi(q)$. Therefore, the palette size of $\pi(q)$ is at least $|S_j| - 3\epsilon_{i_1}\Delta^* + Z_{ex}$. The probability that $\pi(q)$ is skipped is at most $\left(\frac{i}{|S_j|+Z_{ex}-3\epsilon_{i_1}\Delta^*}\right)^{\sqrt{\Delta^*}}$. Note that

$$\begin{aligned} Z_{ex} - 3\epsilon_{i_1}\Delta^* &= \left(\frac{1}{(2\log^2(1/\xi_k))} - 3\epsilon_{i_1}\right)\cdot\Delta^* \geq \left(\frac{1}{(2\log^2(1/\epsilon_{i_1}))} - 3\epsilon_{i_1}\right)\cdot\Delta^* \\ &\geq \frac{\Delta^*}{4\log^2(1/\epsilon_{i_1})} && \tfrac{1}{\epsilon_i} \geq K \text{ for large enough constant } K \\ &\geq C\cdot\frac{\Delta^*}{\log^2(\Delta^*)} && \text{for some constant } C>0 \end{aligned}$$

Let X be the random variable denoting the total number of vertices skipped. We have

$$\begin{aligned} \mathbf{E}[X] &= \sum_{i=1}^{|S_j|}\left(\frac{i}{|S_j|+Z_{ex}-3\epsilon\Delta^*}\right)^{\sqrt{\Delta^*}} \leq |S_j|\cdot\left(\frac{|S_j|}{|S_j|+C\cdot\frac{\Delta^*}{\log^2(\Delta^*)}}\right)^{\sqrt{\Delta^*}} \\ &\leq 2\Delta^*\cdot\left(\frac{2\Delta^*}{2\Delta^*+C\cdot\frac{\Delta^*}{\log^2(\Delta^*)}}\right)^{\sqrt{\Delta^*}} \leq 2\Delta^*\cdot\left(\frac{1}{1+\frac{C}{2}\cdot\frac{1}{\log^2(\Delta^*)}}\right)^{\sqrt{\Delta^*}} && |S_j|\leq 2\Delta^* \\ &\leq 2\Delta^*\cdot\left(\frac{1}{\exp(\frac{C}{4\log^2(\Delta^*)})}\right)^{\sqrt{\Delta^*}} && 1+x\geq\exp(x/2) \text{ for } 0<x\leq 2 \\ &\leq 2\Delta^*\cdot\exp\left(-\frac{C}{4}\cdot\frac{\sqrt{\Delta^*}}{\log^2(\Delta^*)}\right) = \exp(-\Omega(\mathsf{poly}(\Delta^*))) \end{aligned}$$

◀

Therefore, by Lemma 7 and Lemma 8, a similar version of Lemma 17 in [5] holds.

▶ **Lemma 9.** *Consider an execution of* ModifiedSmallDenseColoring*. Let T be any subset of S and let $\delta = \max_{v\in T}\delta_v$. For any t, the number of uncolored vertices in T is at least t with probability at most* $\Pr(\text{Binomial}(|T|,\delta)\geq t) + \exp(-\Omega(\mathsf{poly}(\Delta^*)))$.

Proof. The proof is essentially the same with that of Lemma 17 in [5]. Let $T = \{v_1,\ldots v_{|T|}\}$ be the vertices listed according to their priorities. Conditioned on any choices of $v_1\ldots v_{q-1}$, the probability that v_q is decolored is at most $\delta_v \leq \delta$ by Lemma 7. Therefore, the probability that at least t vertices are decolored is at most $\Pr(\text{Binomial}(|T|,\delta)\geq t)$. The probability that there is any skipped vertex is at most $\exp(-\Omega(\mathsf{poly}(\Delta^*)))$ by Lemma 8. If there are at least t uncolored vertices, then either there are at least t vertices that are decolored or some vertices are skipped. By taking an union over the two events, we conclude the probability there are at least t uncolored vertices is at most $\Pr(\text{Binomial}(|T|,\delta)\geq t) + \exp(-\Omega(\mathsf{poly}(\Delta^*)))$. ◀

Completing the proof for small blocks, other than stratum 1. We will show that Lemma 6 in [5] holds by using our simulation. That is, we show that after $O(1)$ iterations of ModifiedSmallDenseColoring, w.h.p. for every $v\in\widetilde{V}$ and for each layer $i\in[2,l]$, the number of uncolored layer-i neighbors of v that are in W_k^{S} is at most $\epsilon_i^5\Delta^*$.

Consider a vertex v in $\widetilde{V}$. Let T be the set of layer-i neighbors of v in S. Let $\delta = \max_{u\in T}\delta_u \leq 2\epsilon_i\Delta^*/Z_{ex} \leq 4\epsilon_i\log^*(1/\epsilon_i)$. We execute 6 iterations of procedure ModifiedSmallDenseColoring. Let $t_0 = |T|$ and $t_l = \max(2\delta t_{l-1}, \epsilon^5\Delta^*)$. Note that $t_6\leq\epsilon^5\Delta^*$. Suppose that the number of uncolored vertices in T is at most t_{l-1} at the beginning of iteration l. By Lemma 9, with probability at least $1-\Pr(\text{Binomial}(t_l,\delta)\geq t_{l-1}) - \exp(-\Omega(\mathsf{poly}(\Delta^*))) = 1-\exp(-\Omega(\mathsf{poly}(\Delta^*))) = 1-1/\mathsf{poly}(n)$, at the end of iteration l, the number of uncolored vertices in T is at most t_l. By an union bound over such events over the 6 iterations, with probability at least $1-1/\mathsf{poly}(n)$, the number of uncolored vertices in T is at most $\epsilon_i^5\Delta^*$.

Completing the proof small blocks of stratum 1. Note that stratum 1 only consists of vertices that are in layer 1. Instead of proving an analogous lemma to Lemma 7 in [5], we prove the following lemma that bounds the maximum degree on each layer-1 vertex.

▶ **Lemma 10.** *Suppose that each vertex in W_1^{S} has at least $\Delta^*/2\log^2(1/\mathcal{E}_1)$ excess colors w.r.t. W_1^{S}. Let $v \in \widetilde{V}$. By executing* ModifiedSmallDenseColoring *for $O(1)$ rounds, w.h.p. the number of uncolored vertices in $|N(v) \cap W_1^{\mathsf{S}}|$ is at most $(\Delta^*)^{1/20}$ for all $v \in W_1^{\mathsf{S}}$.*

Proof. Let $T = N(v) \cap W_1^{\mathsf{S}}$ be the neighbors of v in W_1^{S}. Let $\delta = \max_{u \in T} \delta_u \leq 2\epsilon_1 \Delta^*/Z_{ex} = 4\epsilon_1 \log^*(1/\epsilon_1) < (\Delta^*)^{-1/20}/2$, where the last inequality holds since $\epsilon_1 = (\Delta^*)^{-1/10}$. We will execute 19 iterations of ModifiedSmallDenseColoring.

Let $t_0 = |T|$ and $t_l = \max((\Delta^*)^{-1/20} t_{l-1}, (\Delta^*)^{1/20})$ for $1 \leq l \leq 19$. Suppose that the number of uncolored vertices in T is at most t_{l-1} at the beginning of iteration l. By Lemma 9, with probability at least $1 - \Pr(\text{Binomial}(t_l, \delta) \geq t_{l-1}) - \exp(-\Omega(\mathsf{poly}(\Delta^*))) = 1 - \exp(-\Omega(\mathsf{poly}(\Delta^*))) = 1 - 1/\mathsf{poly}(n)$, at the end of iteration l, the number of uncolored vertices in T is at most t_l. Therefore, by an union bound on the events over the 19 iterations, w.h.p. the number of uncolored vertices in T is at most $t_l \leq |T| \cdot (\Delta^*)^{19/20} \leq (\Delta^*)^{1/20}$. ◀

By Lemma 10, since the maximum degree of the induced subgraph of layer 1 vertices is at most $(\Delta^*)^{1/20}$ and $N \cdot (\Delta^*)^{1/20} = O(n)$ (by Property (III)), a leader can collect the entire topology and the palette of each vertex and compute a coloring of the W_1^{S} locally.

Version 2 – Dense Coloring Procedure for Strata of Large Blocks. Let $S = W_2^{\mathsf{L}} \cup W_3^{\mathsf{L}} \cup \ldots \cup W_s^{\mathsf{L}}$ or $S = W_1^{\mathsf{L}}$ be a set vertices in these large blocks. A crucial difference between large blocks and small blocks is that for any block $B \subseteq S$, if $v \in B$, the number of external neighbors in other blocks with lower or equal layers in S is at most $O(\epsilon\Delta^*)$. This property allows us to deal with all the large blocks simultaneous. Suppose that S is partitioned into $S_1, \ldots, S_g$ (vertex-disjoint) blocks, where each block S_j is associated with an ID, $ID(S_j) = \min_{v \in S_j} ID(v)$. We associate each S_j with parameters D_j and δ_j. Roughly speaking, D_j represents an upper bound on both the external degree and the anti-degree of each vertex in S_j and δ_j is an upper bound on the probability that a vertex in S_j fails to be colored in a single iteration of ModifiedLargeDenseColoring. We say a S_j has a higher priority than $S_{j'}$ if either (1) $\delta_j < \delta_{j'}$ or (ii) $\delta_j = \delta_{j'}$ and $ID(S_j) < ID(S_{j'})$. For a vertex $v \in S_j$, let $N''(v)$ be the neighbors of v higher priority blocks in S. We simplify the analysis (compared to that of [5]) by choosing a slightly larger δ_j. We let $\delta_j = 2 \cdot \sqrt{D_j/Z_j}$. We modify the algorithm as follows:

ModifiedLargeDenseColoring (Modified Alg. of DenseColoringStep, version 2 from [5]).

1. Each cluster S_j selects $(1-\delta)|S_j|$ vertices S_j' u.a.r. The vertices in S_j' are the *selected* vertices. Let π be a random permutation of selected vertices, chosen u.a.r. Each selected vertex $\pi(q)$ send a set $C(\pi(q))$ of $\sqrt{\Delta^*}$ colors selected u.a.r. from its palette to the leader. For each selected vertex $\pi(q)$, if $C(\pi(q)) \setminus \{c(\pi(1)), \ldots, c(\pi(q-1))\}$ is non-empty, then the leader assigns $\pi(q)$ a color $c(\pi(q))$ randomly selected from the set. Otherwise, we say $\pi(q)$ is *skipped*.
2. Each $v \in S_j$ that has selected a color $c(v)$ permanently color itself $c(v)$, if $c(v)$ is not selected by any vertices $u \in N''(v)$. Otherwise, we say that v is *decolored*.

The algorithm above will be executed for $O(1)$ iterations.

▶ **Lemma 11.** *[Analogue of Lemma 19 in [5]] Let $T = \{v_1, \ldots, v_k\}$ be any subset of uncolored vertices of S_j. The probability that v is decolored for all $v \in T$ is $O(\delta_j)^{|T|}$, conditioned on*

any choices made by vertices in higher priority blocks than S_j and on whether vertices are selected in $S_j \setminus T$.

Proof. We assume that $v_1, \ldots v_k$ are among the $(1-\delta_j)|S_j|$ vertices that are selected. Otherwise, the probability that all vertices in T are decolored would be 0. Let $c_1, \ldots, c_k$ be a sequence of colors. Let $\mathcal{E}_q$ denote the event that v_m select c_m for $m \in [1,q]$. We have:

$$\Pr(\mathcal{E}_q \mid \mathcal{E}_{q-1}) \leq \frac{1}{\delta_j|S_j| - D_j} \leq \frac{1}{\delta_j \cdot Z_j - D_j} \leq \frac{1}{2 \cdot \sqrt{D_j Z_j} - D_j} \leq \frac{1}{\sqrt{D_j Z_j}}$$

Therefore, $\Pr(\mathcal{E}_k) \leq \left(\frac{1}{\sqrt{D_j Z_j}}\right)^k$. For every vertex v_q, there are at most D_j different colors chosen by $N''(v_q)$ that can cause v_q to become decolored. By considering the $(D_j)^k$ combination of forbidden colors for $v_1 \ldots, v_k$, we conclude the probability that all vertices in T are decolored is at most $(D_j/\sqrt{D_j Z_j})^k = O(\delta_j)^k$. ◀

▶ **Lemma 12.** *Let $T = \{v_1, \ldots, v_k\}$ be any subset vertices of S_j. The probability that v_q is skipped for all $q \in [1,k]$ is $O(1/\sqrt{\Delta^*})^{|T|}$. The statement is true even if we are conditioning on any choices made by vertices in higher priority blocks than S_j and on whether vertices are selected and decolored in $S_j \setminus T$.*

Proof. First we assume all nodes in T are selected in S'_j. Otherwise, the probability that v_q is skipped for all $q \in [1,k]$ is 0. Let S''_j denote the set of vertices in S'_j that are skipped.

Let r_q be the rank of v_q among $(S'_j \setminus T) \cup \{v_1 \ldots v_q\}$ in the permutation. Let $\mathcal{E}_q$ denote the event that for $m \in [1,q]$, all the colors picked by v_m are assigned to some vertices in $S'_j \setminus T$ with smaller ranks than v_m. Note that $T \subseteq S''_j$ if and only if $\mathcal{E}_k$ holds. If r_q is fixed, we have $\Pr(\mathcal{E}_q \mid \mathcal{E}_{q-1}) \leq \left(\frac{r_q - 1}{|S_j| - D_j}\right)^{\sqrt{\Delta^*}}$. Therefore,

$$\Pr(T \subseteq S''_j) = \mathbf{E}\big[\Pr(T \subseteq S''_j)\big]_{r_1 \ldots r_k} \leq \mathbf{E}\left[\prod_{q=1}^{k} \left(\frac{r_q - 1}{|S_j| - D_j}\right)^{\sqrt{\Delta^*}}\right]_{r_1 \ldots r_k}$$

$$= \prod_{q=1}^{k} \mathbf{E}\left[\left(\frac{r_q - 1}{|S_j| - D_j}\right)^{\sqrt{\Delta^*}} \mid r_1 \ldots r_{q-1}\right]_{r_q}$$

$$\leq \prod_{q=1}^{k} \left(\frac{1}{|S'_j| - k + q} \sum_{x=1}^{|S'_j|-k+q} \left(\frac{x-1}{|S_j| - D_j}\right)^{\sqrt{\Delta^*}}\right) \leq \left(\frac{1}{|S'_j|} \cdot \sum_{x=1}^{|S'_j|} \left(\frac{x-1}{|S_j| - D_j}\right)^{\sqrt{\Delta^*}}\right)^k$$

$$\leq \left(\frac{1}{|S'_j|} \cdot O\left(\frac{|S'_j|^{\sqrt{\Delta^*}+1}}{\sqrt{\Delta^*} \cdot (|S_j| - D_j)^{\sqrt{\Delta^*}}}\right)\right)^k \leq O\left(\frac{1}{\sqrt{\Delta^*}}\right)^k \qquad |S'_j| \leq |S_j| - D_j$$ ◀

▶ **Lemma 13.** *Let T be any subset vertices of S that have not been assigned a color and let $\delta = \max_{j:S_j \cap T \neq} \delta_j$. After an iteration of the algorithm, the probability that the number of uncolored vertices in T is at least t is at most $\binom{|T|}{t} \cdot (O(\delta + (\Delta^*)^{-1/2}))^t$.*

Proof. Suppose that the clusters $S_1, \ldots, S_g$ are ordered by their priorities, from the highest to the lowest. Let U be a size-t subset of T. we consider the 3^t ways of partitioning U into U_1, U_2, and U_3. Note that a vertex remains uncolored only if it is either unselected, decolored, or skipped. We will calculate the probability that the vertices in U_1 are not selected, vertices in U_2 are decolored, and vertices in U_3 are skipped.

Define $U_i^{(j)} = U_i \cap S_j$ for $i = 1, 2, 3$ and $j = 1, 2 \ldots, g$. The probability that every vertex in $U_1^{(j)}$ is unselected is at most $O(\delta_j)^{|U_1^{(j)}|} \leq O(\delta)^{|U_1^{(j)}|}$. By Lemma 11, the probability that every vertex in $U_2^{(j)}$ is decolored is at most $O(\delta_j)^{|U_2^{(j)}|} \leq O(\delta)^{|U_2^{(j)}|}$. By Lemma 12, the probability that every vertex in $U_3^{(j)}$ is skipped is at most $O(\frac{1}{\sqrt{\Delta^*}})^{|U_3^{(j)}|}$. Therefore, the probability that all vertices in U_1 are unselected, all vertices in U_2 are decolored, all vertices in U_3 are skipped are at most $\prod_{j=1}^{g} O(\delta)^{|U_1^{(j)}|} \cdot O(\delta)^{|U_2^{(j)}|} \cdot O((\Delta^*)^{-1/2})^{|U_3^{(j)}|} = O(\delta + (\Delta^*)^{-1/2})^t$. By an union over all possible size-t sets and partitions, the probability there are at least t uncolored vertices is at most $3^t \cdot \binom{|T|}{t} \cdot O(\delta + (\Delta^*)^{-1/2})^t \leq \binom{|T|}{t} \cdot O(\delta + (\Delta^*)^{-1/2})^t$. ◀

Maintenance of Invariants. Suppose that S_j is a layer-i block. We show that w.h.p. the following invariants are maintained after each iteration l.

- Invariant $\mathcal{H}_l(v)$: Both the anti-degree the external degree of v are at most $D_j^{(l+1)}$.
- Invariant $\mathcal{H}_l(S_j)$: the number of uncolored vertices of S_j is at least $Z_j^{(l+1)}$.
- Sequence $\left(D_j^{(l)}\right)$: $D^{(1)} = 3\epsilon_i \Delta^*$, and $D_j^{(l)} = \beta \cdot \delta_j^{(l-1)} \cdot D_j^{(l-1)}$, for $l > 1$, where $\beta > 1$ is an absolute constant.
- Sequence $\left(Z_j^{(l)}\right)$: $Z^{(1)} = \frac{\Delta^*}{\log^2(1/\epsilon_i)}$, and $Z_j^{(l)} = \delta_j^{(l-1)} \cdot Z_j^{(l-1)}$, for $l > 1$.

▶ **Observation 14.** *$\delta_j^{(l)} = \Omega(1/\sqrt{\Delta^*})$ for every $l \geq 1$ and so the probability in Lemma 13 is $\binom{|T|}{t} \cdot (O(\delta))^t$.*

Proof. For a layer-i block, $D_j^{(1)} = 3\epsilon_i \cdot \Delta^*$, $Z_j^{(1)} = \Delta^* / \log^2(1/\epsilon_i)$, thus $\delta_j^{(1)} = 2\sqrt{D_j^{(1)}/Z_j^{(1)}} = \Omega(\sqrt{\epsilon_i \cdot \log^2(1/\epsilon_i)}) = \Omega((\Delta^*)^{-1/20}) = \Omega((\Delta^*)^{-1/2})$, since $\epsilon_i \geq 1/(\Delta^*)^{1/10}$. Next, note that $\delta_j^{(l)}$ is increasing with l, since $\delta_j^{(l)} = 2\sqrt{\beta} \cdot \delta_j^{(l-1)} > \delta_j^{(l-1)}$. ◀

For every block S_j, in the beginning it is clearly true that $\mathcal{H}_0(v)$ holds for $v \in S_j$ and $\mathcal{H}_0(S_j)$ holds. Suppose that for every block S_j, $\mathcal{H}_{l-1}(v)$ holds for $v \in S_j$ and $\mathcal{H}_{l-1}(S_j)$ holds. Consider a block S_j. Since the number of unselected vertices is always at least $\delta_j^{(l)} \cdot |S_j| \geq \delta_j^{(l)} \cdot Z_j^{(l)} = Z_j^{(l+1)}$, $\mathcal{H}_l(S_j)$ hold with probability 1.

Consider a vertex $v \in S_j$. Let T be the set of uncolored external neighbors of v or the set of uncolored non-neighbor of v in S_j. Since $\mathcal{H}_{l-1}(v)$ holds, $|T| \leq D_j^{(l)}$. Let $t = \beta \delta_j^{(l)} \cdot D_j^{(l)}$. By Lemma 13 and Obs. 14, we have $\Pr(\mathcal{H}_l(v)) \geq 1 - \binom{|T|}{t} \cdot (K\delta_j^{(l)})^t \geq 1 - (\frac{Kt}{\beta T} \cdot \frac{e|T|}{t})^t \geq 1 - 1/\mathsf{poly}(n)$. By taking an union bound on the event $\mathcal{H}^l(v)$ holds for all $v \in S$, w.h.p. the invariants hold after iteration l.

Completing the proof for large blocks, other than stratum 1. We show that after $O(1)$ iterations of ModifiedLargeDenseColoring, w.h.p. for every $v \in \widetilde{V}$ and for each layer $i \in [2, l]$, the number of uncolored layer-i neighbors of v that are in $W_2^{\mathsf{L}} \cup W_3^{\mathsf{L}} \cup \ldots \cup W_s^{\mathsf{L}}$ is at most $\epsilon_i^5 \Delta^*$. (This is the analogue of Lemma 8 [5].) We execute Algorithm ModifiedLargeDenseColoring for 12 iterations. W.h.p. for $l \in [0, 12]$, the invariants $\mathcal{H}_l(S_j)$ hold for every S_j and $\mathcal{H}_l(v)$ holds for every $u \in S$. Let T be the set of layer-i neighbors of v in S and

$$\delta^{(l)} = \max_{j: S_j \cap T \neq} \delta_j^{(l)} = 2 \cdot \sqrt{D_j^{(l)}/Z_j^{(l)}} = 2\sqrt{\beta}^{l-1} \cdot \sqrt{3\epsilon_i \cdot \log^2(1/\epsilon_i)}.$$

Let $t_0 = |T|$ and $t_l = \max(\beta \delta^{(l)} t_{l-1}, \epsilon^5 \Delta^*)$. Suppose that at the beginning of iteration l, the number of uncolored vertices in T is at most t_{l-1}. By Lemma 13 and Obs. 14, after iteration l, the probability that the number of uncolored vertices is more than t_l

is at most $\binom{t_{l-1}}{t_l}(K \cdot \delta^{(l)})^{t_l} \leq \exp(-\Omega(t_l)) = 1/\mathsf{poly}(n)$. Note that since $\prod_{l=1}^{12} \delta^{(l)} \leq \prod_{l=1}^{12} \left(2\sqrt{\beta}^{l-1} \cdot \sqrt{3\epsilon_i \cdot \log^2(1/\epsilon_i)}\right) \leq (2\sqrt{3})^{12}\beta^{66} \cdot \epsilon_i^6 \cdot \log^{12}(1/\epsilon) \leq \epsilon_i^5$, $t_l = \epsilon^5\Delta^*$. By taking an union on the events over 12 iterations, we conclude w.h.p. the number of uncolored neighbors in T is at most $t_l = \epsilon^5\Delta^*$.

Completing the proof large blocks of stratum 1. Note that stratum 1 only consists of vertices that are in layer 1. Instead of proving an analogous lemma to Lemma 9 in [5], we prove the following lemma that bounds the maximum degree on each layer-1 vertex.

▶ **Lemma 15.** *Let $v \in \widetilde{V}$. By executing* ModifiedLargeDenseColoring *for $O(1)$ rounds, w.h.p. the number of uncolored vertices in $|N(v) \cap W_1^{\mathsf{L}}|$ is at most $(\Delta^*)^{1/20}$ for all $v \in W_1^{\mathsf{L}}$.*

Proof. We execute 19 iterations of ModifiedLargeDenseColoring. W.h.p. for $l \in [0, 38]$, the invariants $\mathcal{H}_l(S_j)$ hold for every S_j and $\mathcal{H}_l(v)$ holds for every $u \in S$. Let $T = N(v) \cap W_1^{\mathsf{L}}$ be the neighbors of v in W_1^{S}. Let $\delta^{(l)} = \max_{j:S_j \cap T \neq} \delta_j^{(l)} = 2 \cdot \sqrt{D_j^{(l)}/Z_j^{(l)}} = 2\sqrt{\beta}^{l-1} \cdot \sqrt{3\epsilon_i \cdot \log^2(1/\epsilon_i)} \leq (\Delta^*)^{-1/20}/100$.

Let $t_0 = |T|$ and $t_l = \max((\Delta^*)^{-1/20}t_{l-1}, (\Delta^*)^{1/20})$ for $1 \leq l \leq 19$. Suppose that the number of uncolored vertices in T is at most t_{l-1} at the beginning of iteration l. By Lemma 13 and Obs. 14, after iteration l, the probability that the number of uncolored vertices is more than t_l is at most $\binom{t_{l-1}}{t_l}(K \cdot \delta^{(l)})^{t_l} \leq \exp(-\Omega(t_l)) = 1/\mathsf{poly}(n)$. Therefore, by an union bound on the events over the 19 iterations, w.h.p. the number of uncolored vertices in T is at most $t_l \leq |T| \cdot (\Delta^*)^{19/20} \leq (\Delta^*)^{1/20}$. ◀

By Lemma 15, since the maximum degree of the induced subgraph of layer 1 vertices is at most $(\Delta^*)^{1/20}$ and $N \cdot (\Delta^*)^{1/20} = O(n)$ (by Property (III)), a leader can collect the entire topology and the palette of each vertex and compute a coloring of the W_1^{L} locally.

Coloring the Remaining Vertices – Simulation of ColorBidding Algorithm. It is therefore remains to color two subsets of vertices: a subset U of vertices that were not colored by the Dense Coloring Procedures and V_{sp}. Similarly to [5], we will apply the ColorBidding Algorithm to first color all the vertices in U. By Lemma 10 in [5], after $O(\log^* \Delta)$ iterations, the probability that a vertex remains uncolored is $\exp(-\mathsf{poly}(\Delta^*)) = 1/\mathsf{poly}(n)$. Then we repeat the same procedure to color vertices in V_{sp} in $O(\log^* \Delta)$ rounds w.h.p.

The analysis is mostly straightforward from [5] and the main missing argument is in showing that a single iteration of Alg. ColorBidding can be simulated in $O(1)$ rounds in the congested clique model. A simple calculation yields that each vertex selects $\widetilde{O}(\Delta^*)^{1/5}$ colors in this palette. Hence, each vertex is a target and a sender of $\widetilde{O}(\Delta^*)^{4/5} = O(n)$ messages, which fits the scheme of Lenzen's routing. A more detailed description is in the full version. The next observation follows from Lemma 2.3 in [16].

▶ **Observation 16.** *The list-coloring algorithm of Theorem 5 holds up to minor modifications even when every $v \in G^*$ has at least $\Delta^* - (\Delta^*)^{3/5}$ available colors in its palette.*

Putting it all together. It remains to show that the subgraph G^* from Section 2 indeed satisfies the conditions of Theorem 5 and Observation 16.

▶ **Lemma 17.** *(1) The subgraph G^* satisfies all the properties of Sec. 3. (2) Every $v \in G^*$ has at least $\max\{\deg(v, G^*) + 1, \Delta^* - (\Delta^*)^{3/5}\}$ available colors in its palette after coloring all its neighbors in $G \setminus G^*$.*

Proof. Part (1) follows by plugging the bounds of Observation 4. Next consider Claim (2). First, consider the case where $\deg(v,G) \leq \Delta - (\Delta^* - \sqrt{5\Delta^* \cdot \log n})$. In such case, even after coloring all neighbors of v, it still has an access of $\Delta^* - \sqrt{5\Delta^* \cdot \log n} \geq \Delta^* - (\Delta^*)^{3/5}$ colors in its palette after coloring $G \setminus G^*$ in the first phase. Now, consider a vertex v with $\deg(v,G) \geq \Delta - (\Delta^* - \sqrt{5\Delta^* \cdot \log n})$. Using Chernoff bound, w.h.p., $\deg(v,G^*) > (\Delta - (\Delta^* - \sqrt{\Delta^* \cdot 5\log n})) \cdot p^* - \sqrt{5\log n \Delta p^*} \geq \Delta^* - (\Delta^*)^{3/5}$. A vertex $v \in G^*$ has at least $\deg(v,G^*)+1$ available colors, since all its neighbors in G^* are uncolored at the beginning of the second phase and initially it was given $(\Delta+1)$ colors. ◀

References

1 Leonid Barenboim and Victor Khazanov. Distributed symmetry-breaking algorithms for congested cliques. *arXiv preprint arXiv:1802.07209*, 2018.

2 Soheil Behnezhad, Mahsa Derakhshan, and MohammadTaghi Hajiaghayi. Brief announcement: Semi-mapreduce meets congested clique. *arXiv preprint arXiv:1802.10297*, 2018.

3 Andrew Berns, James Hegeman, and Sriram V Pemmaraju. Super-fast distributed algorithms for metric facility location. In *International Colloquium on Automata, Languages, and Programming*, pages 428–439. Springer, 2012.

4 Keren Censor-Hillel, Merav Parter, and Gregory Schwartzman. Derandomizing local distributed algorithms under bandwidth restrictions. In *31st International Symposium on Distributed Computing, DISC 2017, October 16-20, 2017, Vienna, Austria*, pages 11:1–11:16, 2017.

5 Yi-Jun Chang, Wenzheng Li, and Seth Pettie. An optimal distributed $(\Delta+1)$-coloring algorithm? In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA*, pages 445–456. ACM, 2018.

6 Mohsen Ghaffari. Distributed MIS via all-to-all communication. In *Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC 2017, Washington, DC, USA, July 25-27, 2017*, pages 141–149, 2017.

7 Mohsen Ghaffari, Themis Gouleakis, Christian Konrad, Slobodan Mitrovic, and Ronitt Rubinfeld. Improved massively parallel computation algorithms for mis, matching, and vertex cover. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing, PODC 2018, Egham, United Kingdom*, pages 129–138, 2018.

8 David G Harris, Johannes Schneider, and Hsin-Hao Su. Distributed $(\Delta+1)$-coloring in sublogarithmic rounds. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, pages 465–478. ACM, 2016.

9 James W Hegeman and Sriram V Pemmaraju. Lessons from the congested clique applied to mapreduce. *Theoretical Computer Science*, 608:268–281, 2015.

10 James W Hegeman, Sriram V Pemmaraju, and Vivek B Sardeshmukh. Near-constant-time distributed algorithms on a congested clique. In *International Symposium on Distributed Computing*, pages 514–530. Springer, 2014.

11 Tomasz Jurdzinski and Krzysztof Nowicki. MST in $O(1)$ rounds of congested clique. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 2620–2632, 2018.

12 Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. Local computation: Lower and upper bounds. *Journal of the ACM (JACM)*, 63(2):17, 2016.

13 Christoph Lenzen. Optimal deterministic routing and sorting on the congested clique. In *Proceedings of the 2013 ACM symposium on Principles of distributed computing*, pages 42–50. ACM, 2013.

14 Zvi Lotker, Boaz Patt-Shamir, Elan Pavlov, and David Peleg. Minimum-weight spanning tree construction in $O(\log \log n)$ communication rounds. *SIAM Journal on Computing*, 35(1):120–131, 2005.

15 Merav Parter. $(\Delta+1)$ coloring in the congested clique model. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, volume 107 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 160:1–160:14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018.

16 Merav Parter. $(\Delta + 1)$ coloring in the congested clique model. *arXiv preprint*, 2018. `arXiv:1805.02457`.

17 Johannes Schneider and Roger Wattenhofer. A new technique for distributed symmetry breaking. In *Proceedings of the 29th ACM SIGACT-SIGOPS symposium on Principles of distributed computing*, pages 257–266. ACM, 2010.

A Missing Pseudocodes for the Subroutines of [5]

OneShotColoring. Each uncolored vertex v decided to participates independently with probability p. Each participating vertex v selects a color $c(v)$ from its palette $\Psi(v)$ uniformly at random. A participating vertex v successfully colors itself if $c(v)$ is not chosen by any vertex in $N^*(v)$, where $N^*(v) = \{u \in N(v) \mid ID(u) < ID(v)\}$.

In the ColorBidding procedure each vertex v is associated with a parameter $p_v \geq |\Psi(v)| - \text{outdeg}(v)$ and $p^* = \min_v p_v$. Let C be a constant satisfying that $\sum_{u \in N_{out}(v)} 1/p_u \leq 1/C$.

ColorBidding. Each color $c \in \Psi(v)$ is added to S_v with probability $C/2p_v$ independently. If there exists a color $c^* \in S_v$ that is not selected by vertices in $N_{out}(v)$, v colors itself c^*. $N_{out}(v)$ is the set of neighbors of v that have higher priority than v.

The CLP algorithm contains two versions of the dense coloring procedures. Version 1 is used to color the small blocks and version 2 is used to color the large blocks. All vertices in S agree on a parameter Z_{ex} which is a lower bound on the number of excess colors with respect to S. Each vertex $v \in S_j$ is associated with a parameter D_v. Let $N'(v) = \{u \in N(v) \mid D_u \leq D_v \text{ or } D_u = D_v \text{ and } ID(u) < ID(v)\}$ to be the neighbors of v with higher priority. If $v \in S_j$, the parameter D_v must satisfy $|N'(v) \cap (S \setminus S_j)| \leq D_v$. Define $\delta_v = D_v/Z_{ex}$.

Procedure DenseColoringStep (Version 1)

1. Let $\pi = \{1, \ldots, |S_j|\} \to S_j$ be the permutation that lists S_j in increasing order by D-value, breaking ties by ID. For q from 1 to $|S_j|$, the vertex $\pi(q)$ selects a color $c(\pi(q))$ uniformly at random from $\Psi(\pi(q)) \setminus \{c(\pi(q')) \mid q' < q \text{ and } \{\pi(q), \pi(q')\} \in E(G)\}$.
2. Each $v \in S_j$ permanently colors itself $c(v)$ if $c(v)$ is not selected by any vertices in $N'(v)$.

For version 2, for each vertex $v \in S$, define $N''(v)$ to be the set of vertices $u \in N(v) \cap S$ such that (i) $\delta_{j'} > \delta_j$ or (ii) $\delta_{j'} = \delta_j$ and $ID(S_j) < ID(S_{j'})$, where $v \in S_j$ and $u \in S_{j'}$.

Procedure DenseColoringStep (Version 2)

1. Each cluster S_j selects $(1-\delta_j)|S_j|$ vertices u.a.r. and generates a permutation π of those vertices u.a.r. The vertex $\pi(q)$ selects a color $c(\pi(q))$ u.a.r. from

$$\Psi(\pi(q)) \setminus \{c(\pi(q')) \mid q' < q \text{ and } \{\pi(q), \pi(q')\} \in E(G)\}.$$

2. Each $v \in S_j$ that has selected a color $c(v)$ permanently colors itself $c(v)$ if $c(v)$ is not selected by any vertices $u \in N''(v)$.