Efficiently list-edge coloring multigraphs asymptotically optimally

Fotis Iliopoulos *
University of California Berkeley
fotis.iliopoulos@berkeley.edu

Alistair Sinclair †
University of California Berkeley
sinclair@berkeley.edu

Abstract

We give polynomial time algorithms for the seminal results of Kahn [18, 19], who showed that the Goldberg-Seymour and List-Coloring conjectures for (list-)edge coloring multigraphs hold asymptotically. Kahn's arguments are based on the probabilistic method and are non-constructive. Our key insight is to show that the main result of Achlioptas, Iliopoulos and Kolmogorov [2] for analyzing local search algorithms can be used to make constructive applications of a powerful version of the so-called Lopsided Lovász Local Lemma. In particular, we use it to design algorithms that exploit the fact that correlations in the probability spaces on matchings used by Kahn decay with distance.

*Research supported by NSF grant CCF-1514434 and the Onassis Foundation.
†Research supported by NSF grants CCF-1514434 and CCF-1815328.

1 Introduction

In graph edge coloring one is given a (multi)graph $G(V, E)$ and the goal is to find an assignment of one of q colors to each edge $e \in E$ so that no pair of adjacent edges share the same color. The *chromatic index*, $\chi_e(G)$, of G is the smallest integer q for which this is possible. In the more general *list-edge coloring* problem, a list of q allowed colors is specified for each edge. A graph is q-list-edge colorable if it has a list-coloring no matter how the lists are assigned to each edge. The *list chromatic index*, $\chi_e^\ell(G)$, is the smallest q for which G is q-list-edge colorable.

Edge coloring is one of the most fundamental and well-studied coloring problems with various applications in computer science (e.g., [6, 11, 17, 18, 19, 29, 31, 33, 34, 35, 37]). To give just one representative example, if edges represent data packets then an edge coloring with q colors specifies a schedule for exchanging the packets directly and without node contention. In this paper we are interested in designing algorithms for efficiently edge coloring and list-edge coloring multigraphs. To formally describe our results, we need some notation.

For a multigraph G let $\mathcal{M}(G)$ denote the set of matchings of G. A *fractional edge coloring* is a set $\{M_1, \ldots, M_\ell\}$ of matchings and corresponding positive real weights $\{w_1, \ldots, w_\ell\}$, such that the sum of the weights of the matchings containing each edge is one. I.e., $\forall e \in E$, $\sum_{M_i : e \in M_i} w_i = 1$. A fractional edge coloring is a *fractional edge c-coloring* if $\sum_{M \in \mathcal{M}(G)} w_M = c$. The *fractional chromatic index* of G, denoted by $\chi_e^*(G)$, is the minimum c such that G has a fractional edge c-coloring.

Let $\Delta = \Delta(G)$ be the maximum degree of G and define $\Gamma := \max_{H \subseteq V, |H| \geq 2} \frac{|E(H)|}{\lfloor |H|/2 \rfloor}$. Both of these quantities are obvious lower bounds for the chromatic index and it is known [8] that $\chi_e^*(G) = \max(\Delta, \Gamma)$. Furthermore, Padberg and Rao [30] show that the fractional chromatic index of a multigraph, and indeed an optimal fractional edge coloring, can be computed in polynomial time.

A famous and long-standing conjecture by Goldberg and Seymour states that every multigraph G satisfies $\chi_e(G) \leq \max(\Delta + 1, \lceil \chi_e^*(G) \rceil)$. In a seminal paper [18], Kahn showed that the Goldberg-Seymour conjecture holds asymptotically:

Theorem 1.1 ([18])**.** *For multigraphs G, $\chi_e(G) \leq (1 + o(1))\chi_e^*(G)$.*

(Here $o(1)$ denotes a term that tends to zero as $\chi_e(G) \to \infty$.) He later [19] proved the analogous result for list-edge coloring, establishing that the List Coloring Conjecture, which asserts that $\chi_e^\ell(G) = \chi_e(G)$ for any multigraph G, also holds asymptotically:

Theorem 1.2 ([19])**.** *For multigraphs G, $\chi_e^\ell(G) \leq (1 + o(1))\chi_e^*(G)$.*

The proofs of Kahn use the probabilistic method and are not constructive. The main contribution of this paper is to provide polynomial time algorithms for the above results, as follows:

Theorem 1.3. *For every $c > 0$, there exists an algorithm that, given a multigraph G on n vertices, constructs a $(1 + o(1))\chi_e^*(G)$-edge coloring of G with probability at least $1 - \frac{1}{n^c}$ in expected polynomial time.*

Theorem 1.4. *For every $c > 0$, there exists an algorithm that, given a multigraph G on n vertices and an arbitrary list of $q = (1 + o(1))\chi_e^*(G)$ colors for each edge, constructs a q-list-edge coloring of G with probability at least $1 - \frac{1}{n^c}$ in expected polynomial time.*

Clearly, Theorem 1.4 subsumes Theorem 1.3. Furthermore, the results of Sanders and Steurer [33] and Scheide [35] already give polynomial time algorithms for edge coloring multigraphs asymptotically optimally, without exploiting the arguments of Kahn. Nonetheless, we choose to present the proof of Theorem 1.3 for three reasons. First and most importantly, its proof is significantly easier than that of Theorem 1.4, while it contains many of the key ideas required for making Theorem 1.2 constructive. Second, our

algorithms and techniques are very different from those of [33, 35]. Finally, as we will see, we will need to show that the algorithm of Theorem 1.3 is *commutative*, a notion introduced by Kolmogorov [22]. This fact may be of independent interest since, as shown in [22, 14], commutative algorithms have several nice properties: they are typically parallelizable, their output distribution has high entropy, etc.

As a final remark, we note that, to the best of our knowledge, Theorem 1.4 is the first result to give an asymptotically optimal polynomial time algorithm for list-edge coloring multigraphs.

1.1 Technical Overview

The proofs of Theorems 1.1 and 1.2 are based on a very sophisticated variation of what is known as the *semi-random method* (also known as the "naive coloring procedure"), which is the main technical tool behind some of the strongest graph coloring results, e.g., [16, 17, 21, 25]. The idea is to gradually color the graph in iterations, until we reach a point where we can finish the coloring using a greedy algorithm. In its most basic form, each iteration consists of the following simple procedure (using vertex coloring as a canonical example): Assign to each vertex a color chosen uniformly at random; then uncolor any vertex which receives the same color as one of its neighbors. Using the Lovász Local Lemma (LLL) [9] and concentration inequalities, one typically shows that, with positive probability, the resulting partial proper coloring has useful properties that allow for the continuation of the argument in the next iteration. For a nice exposition of both the method and the proofs of Theorems 1.1 and 1.2, the reader is referred to [26].

The key new ingredient in Kahn's arguments is the method of assigning colors. For each color c, we choose a matching M_c from some *hard-core* distribution on $\mathcal{M}(G)$ and assign the color c to the edges in M_c. The idea is that by assigning each color exclusively to the edges of one matching, we avoid conflicting color assignments and the resulting uncolorings.

The existence of such hard-core distributions is guaranteed by the characterization of the matching polytope due to Edmonds [8] and a result by Lee [23] (also shown independently by Rabinovich et al. [32]). The crucial fact about them is that they are endowed with very useful approximate stochastic independence properties, as was shown by Kahn and Kayll in [20]. In particular, for every edge e, conditioning on events that are determined by edges far enough from e in the graph does not effectively alter the probability of e being in the matching.

The reason why this property is important is because it enables the application of a sophisticated version of what is known as the *Lopsided* Lovász Local Lemma (LLL). Recall that the original statement of the LLL asserts, roughly, that, given a family of "bad" events in a probability space, if each bad event individually is not very likely and, in addition, is independent of all but a small number of other bad events, then the probability of avoiding all bad events is strictly positive. The Lopsided LLL used by Kahn generalizes this criterion as follows. For each bad event B, we fix a positive real number μ_B and require that conditioning on all but a small number of other bad events doesn't make the probability of B larger than μ_B. Then, provided the μ_B are small enough, the conclusion of the LLL still holds. In other words, one replaces the "probability of a bad event" in the original LLL statement with the "boosted" probability of the event, and the notion of "independence" by the notion of "sufficiently mild negative correlation".

Notably, the breakthrough result of Moser and Tardos [27, 28] that made the LLL constructive for the vast majority of its applications does not apply in this case, mainly for two reasons. First, the algorithm of Moser and Tardos applies only when the underlying probability measure of the LLL application is a *product* over explicitly presented variables. Second, it relies on a particular type of dependency (defined by shared variables). The lack of an efficient algorithm for Lopsided LLL applications is the primary obstacle to making the arguments of Kahn constructive.

Our main technical contribution is the design and analysis of such algorithms. Towards this goal, we use the flaws-actions framework introduced in [1] and further developed in [2, 4, 14, 3]. In particular, we use the algorithmic LLL criterion for the analysis of stochastic local search algorithms developed by Achlioptas,

Iliopoulos and Kolmogorov in [2]. We start by showing that there is a connection between this criterion and the version of the Lopsided LLL used by Kahn, in the sense that the former can be seen as the constructive counterpart of the latter. However, this observation by itself is not sufficient, since the result of [2] is a tool for analyzing a *given* stochastic local search algorithm. Thus, we are still left with the task of designing the algorithm before using it. Nonetheless, this connection provides valuable intuition on how to realize this task. Moreover, we believe it is of independent interest as it provides an explanation for the success of various algorithms (such as [24]) inspired by the techniques of Moser and Tardos, which were not tied to a known form of the LLL.

To get a feeling for the nature of our algorithms it is helpful to have some intuition for the criterion of [2]. There, the input is the algorithm to be analyzed and a probability measure μ over the state space of the algorithm. The goal of the algorithm is to reach a state that avoids a family of bad subsets of the space which we call *flaws*. It does this by focusing on a flaw that is currently present at each step, and taking a (possibly randomized) action to address it. At a high level, the role of the measure is to gauge how efficiently the algorithm rids the state of flaws, by quantifying the trade-off between the probability that a flaw is present at some inner state of the execution of the algorithm and the number of other flaws each flaw can possibly introduce when the algorithm addresses it. In particular, the quality of the convergence criterion is affected by the *compatibility* between the measure and the algorithm.

Roughly, the states of our algorithm will be matchings in a multigraph (corresponding to color classes) and the goal will be to construct matchings that avoid certain flaws. To that end, our algorithm will locally modify each flawed matching by (re)sampling matchings in subgraphs of G according to distributions induced by the hard-core distributions used in Kahn's proof. The fact that correlations decay with distance in these distributions allows us to prove that, while the changes are local, and hence not many new flaws are introduced at each step, the compatibility of our algorithms with these hard-core distributions is high enough to allow us to successfully apply the criterion of [2].

1.2 Organization of the Paper

In Section 2 we present the necessary background. In Section 3 we show a useful connection between the version of the Lopsided LLL used by Kahn and the algorithmic LLL criterion of [2]. In Section 4 we present the proof of Theorem 1.3. In Section 5, we sketch the proof of Theorem 1.2 and then prove Theorem 1.4.

2 Background and Preliminaries

2.1 The Lopsided Lovász Local Lemma

Erdős and Spencer [10] noted that independence in the LLL can be replaced by positive correlation, yielding the original version of what is known as the *Lopsided* LLL. More sophisticated versions of the Lopsided LLL have also been established in [5, 7]. Below we state the Lopsided LLL in one of its most powerful forms.

Theorem 2.1 (General Lopsided LLL)**.** *Let (Ω, μ) be a probability space and $\mathcal{B} = \{B_1, B_2, \ldots, B_m\}$ be a set of m (bad) events. For each $i \in [m]$, let $L(i) \subseteq [m] \setminus \{i\}$ be such that $\mu(B_i \mid \bigcap_{j \in S} \overline{B_j}) \leq b_i$ for every $S \subseteq [m] \setminus (L(i) \cup \{i\})$. If there exist positive real numbers $\{x_i\}_{i=1}^m$ such that*

$$b_i \leq x_i \prod_{j \in L(i)} (1 - x_j) \quad \text{for all } i \in [m] \ , \tag{1}$$

then the probability that none of the events in $\mathcal{B}$ occurs is at least $\prod_{i=1}^m (1 - x_i) > 0$.

The digraph over $[m]$ induced by the sets $L(i)$, $i \in [m]$, is often called a *lopsidependency* digraph.

2.2 An Algorithmic LLL Criterion

Let Ω be a discrete state space, and let $F = \{f_1, f_2, \ldots, f_m\}$ be a collection of subsets (which we call *flaws*) of Ω such that $\bigcup_{i\in[m]} f_i = \Omega^*$. Our goal is to find a state $\sigma \in \Omega \setminus \Omega^*$; we refer to such states as *flawless*.

For a state σ, we denote by $U(\sigma) = \{j \in [m] \text{ s.t. } f_j \ni \sigma\}$ the set of flaws present in σ. We consider local search algorithms working on Ω which, in each flawed state $\sigma \in \Omega^*$, choose a flaw f_i in $U(\sigma)$ and randomly move to a nearby state in an effort to fix f_i. We will assume that, for every flaw f_i and every state $\sigma \in f_i$, there is a non-empty set of *actions* $a(i,\sigma) \subseteq \Omega$ such that *addressing* flaw f_i in state σ amounts to selecting the next state τ from $a(i,\sigma)$ according to some probability distribution $\rho_i(\sigma,\tau)$. Note that potentially $a(i,\sigma) \cap f_i \neq \emptyset$, i.e., addressing a flaw does not necessarily imply removing it. We sometimes write $\sigma \xrightarrow{i} \tau$ to denote that the algorithm addresses flaw f_i at σ and moves to τ.

Throughout the paper we consider Markovian algorithms that start from a state $\sigma \in \Omega$ picked from an initial distribution θ, and then repeatedly pick a flaw that is present in the current state and address it. The algorithm always terminates when it encounters a flawless state.

Definition 2.2 (Causality)**.** *We say that flaw f_i* causes *f_j if there exists a transition $\sigma \xrightarrow{i} \tau$ such that (i) $f_j \ni \tau$; (ii) either $f_i = f_j$ or $f_j \not\ni \sigma$.*

Definition 2.3 (Causality Digraph)**.** *Any digraph $C = C(\Omega, F)$ on $[m]$ that includes every edge $i \to j$ such that f_i causes f_j is called a* causality digraph*. We write $\Gamma(i)$ for the set of out-neighbors of i in this graph.*

Throughout this paper we consider only algorithms with the property that f_i causes f_j if and only if f_j causes f_i. We will thus view the causality graph as an *undirected* graph. We also write $i \sim j$ to denote that $j \in \Gamma(i)$ (or equivalently, $i \in \Gamma(j)$).

For a given probability measure μ supported on the state space Ω, and for each flaw f_i, we define the *charge*

$$\gamma_i = \max_{\tau\in\Omega} \sum_{\sigma\in f_i} \frac{\mu(\sigma)}{\mu(\tau)} \rho_i(\sigma,\tau) \ . \tag{2}$$

In Section 3 we give the intuition behind the definition of charges and also draw a connection with the parameters μ_i in Theorem 2.1. We are now ready to state the main result of [2].

Theorem 2.4. *Assume that, at each step, the algorithm chooses to address the lowest indexed flaw according to an arbitrary, but fixed, permutation of $[m]$. If there exist positive real numbers $x_i \in (0,1)$ for $1 \le i \le m$ such that*

$$\gamma_i \le (1-\epsilon) x_i \prod_{j\in\Gamma(i)} (1-x_j) \quad \textit{for every } i \in [m] \tag{3}$$

for some $\epsilon \in (0,1)$, then the algorithm reaches a flawless object within $(T_0 + s)/\epsilon$ steps with probability at least $1 - 2^{-s}$, where

$$T_0 = \log_2 \left(\max_{\sigma\in\Omega} \frac{\theta(\sigma)}{\mu(\sigma)} \right) + \sum_{j\in[m]} \log_2 \left(\frac{1}{1-x_j} \right) \ .$$

We also describe another theorem that can be used to show convergence in a polynomial number of steps, even when the number of flaws is super-polynomial, assuming the algorithm has a nice property which we describe below.

Definition 2.5. *For $i \in [m]$, let A_i denote the $|\Omega| \times |\Omega|$ matrix defined by $A_i[\sigma,\sigma'] = \rho_i(\sigma,\sigma')$ if $\sigma \in f_i$, and $A_i[\sigma,\sigma'] = 0$ otherwise. A Markovian algorithm defined by matrices A_i, $i \in [m]$, is* commutative with respect to a causality relation $\sim$ *if for every $i, j \in [m]$ such that $i \not\sim j$ we have $A_i A_j = A_j A_i$.*

We note that Definition 2.5 was introduced in [3], as a generalization of the combinatorial definition of commutativity introduced in [22]. While the latter would suffice for our purposes, we choose to work with Definition 2.5 due to its compactness.

Theorem 2.6. *Let $\mathcal{A}$ be a commutative algorithm with respect to a causality relation $\sim$. Assume there exist positive real numbers $\{x_i\}_{i\in[m]}$ in $(0,1)$ such that condition (3) holds. Assume further that the causality graph induced by $\sim$ can be partitioned into n cliques, with potentially further edges between them. Setting $\delta := \min_{i\in[m]} x_i \prod_{j\in\Gamma(i)}(1-x_j)$, the expected number of steps performed by $\mathcal{A}$ is at most $t = O\left(\max_{\sigma\in\Omega} \frac{\theta(\sigma)}{\mu(\sigma)} \cdot \frac{n}{\epsilon} \log \frac{n\log(1/\delta)}{\epsilon}\right)$, and for any parameter $\lambda \geq 1$, $\mathcal{A}$ terminates within λt resamplings with probability $1 - \mathrm{e}^{-\lambda}$.*

Following Theorem 3.2 in [14], the proof of Theorem 2.6 is identical to the analogous result of Hauepler, Saha and Srinivasan [12] for the Moser-Tardos algorithm and hence we omit it.

2.3 Hard-Core Distributions on Matchings

A probability distribution ν on the matchings of a multigraph G is *hard-core* if it is obtained by associating to each edge e a positive real $\lambda(e)$ (called the *activity* of e) so that the probability of any matching M is proportional to $\prod_{e\in M}\lambda(e)$. Thus, recalling that $\mathcal{M}(G)$ denotes the set of matchings of G, and setting $\lambda(M) = \prod_{e\in M}\lambda(e)$ for each $M \in \mathcal{M}(G)$, we have

$$\nu(M) = \frac{\lambda(M)}{\sum_{M'\in\mathcal{M}(G)}\lambda(M')} \ .$$

The characterization of the matching polytope due to Edmonds [8] and a result of Lee [23] (which was also shown independently by Rabinovich et al. [32]) imply the following connection between fractional edge colorings and hard-core probability distributions on matchings. Before describing it, we need a definition.

For any probability distribution ν on the matchings of a multigraph G, we refer to the probability that a particular edge e is in the random matching as the *marginal* of ν at e. We write $(\nu_{e_1}, \ldots, \nu_{e_{|E(G)|}})$ for the collection of marginals of ν at all the edges $e_i \in E(G)$.

Theorem 2.7. *There is a hard-core probability distribution ν with marginals $(\frac{1}{c}, \ldots, \frac{1}{c})$ if and only if there is a fractional c'-edge coloring of G with $c' < c$, i.e., if and only if $\chi_e^* < c$.*

Kahn and Kayll [20] proved that the probability distribution promised by Theorem 2.7 is endowed with very useful approximate stochastic independence properties.

Definition 2.8. *Suppose we choose a random matching M from some probability distribution. We say that an event Q is t-*distant *from a vertex v if Q is completely determined by the choice of all matching edges at distance at least t from v. We say that Q is t-distant from an edge e if it is t-distant from both endpoints of e.*

Theorem 2.9 ([20])**.** *For any $\delta > 0$, there exists a $K = K(\delta)$ such that for any multigraph G with fractional chromatic number c there is a hard-core distribution ν with marginals $(\frac{1-\delta}{c}, \ldots, \frac{1-\delta}{c})$ such that*

(a) for every $e \in E(G)$, $\lambda(e) \leq \frac{K}{c}$ and hence $\forall v \in V(G)$, $\sum_{e\ni v}\lambda(e) \leq K$.

(b) for every $\epsilon \in (0,1)$, if we choose a matching M according to ν then, for any edge e and event Q which is t-distant from e,

$$(1-\epsilon)\Pr[e\in M] \leq \Pr[e \in M \mid Q] \leq (1+\epsilon)\Pr[e\in M] \ ,$$

where $t = t(\epsilon) = 8(K+1)^2\epsilon^{-1} + 2$.

We conclude this subsection with the result of Jerrum and Sinclair [15] for sampling from hard-core distributions on matchings. The algorithm works by simulating a rapidly mixing Markov chain on matchings, whose stationary distribution is the desired hard-core distribution ν, and outputting the final state.

Theorem 2.10 ([15], Corollary 4.3)**.** *Let G be a multigraph, $\{\lambda(e)\}_{e\in E(G)}$ a vector of activities associated with the edges of G, and ν the corresponding hard-core distribution. Let $n = |V(G)|$ and define $\lambda' = \max\{\max_{u,v\in V(G)} \sum_{e\ni\{u,v\}} \lambda(e), 1\}$. There exists an algorithm that, for any $\epsilon > 0$, runs in time $\mathrm{poly}(n, \lambda', \log \epsilon^{-1})$ and outputs a matching in G drawn from a distribution ν' such that $\|\nu - \nu'\|_{\mathrm{TV}} \leq \epsilon$.*

Remark 2.1. *[15] establishes this result for matchings in (simple) graphs. However, the extension to multigraphs is immediate: make the graph simple by replacing each set of multiple edges $e_1, \ldots, e_\ell$ between a pair of vertices u, v by a single edge e of activity $\lambda(e) = \sum_i \lambda(e_i)$; then use the algorithm to sample a matching from the hard-core distribution in the resulting simple graph; finally, for each edge $e = \{u, v\}$ in this matching, select one of the corresponding multiple edges $e_i \ni \{u, v\}$ with probability $\lambda(e_i)/\sum_i \lambda(e_i)$. Note that the running time will depend polynomially on the maximum activity λ' in the simple graph, as claimed.*

3 Causality, Lopsidependency and Approximate Resampling Oracles

In this section we show a connection between Theorem 2.1 and Theorem 2.4. While this section is not essential to the proof of our main results, it does provide useful intuition since it implies the following natural approach to making applications of the Lopsided LLL algorithmic: We start designing a local search algorithm for addressing the flaws that correspond to bad events by considering the family of probability distributions $\{\rho_i(\sigma, \cdot)\}_{i\in[m],\sigma\in f_i}$ whose supports induce a causality graph that coincides with the lopsidependency graph of the Lopsided LLL application of interest. This is typically an automated task. The key to successful implementation is our ability to make the way in which the algorithm addresses flaws sufficiently *compatible* with the underlying probability measure μ. To make this precise, we first recall an algorithmic interpretation of the notion of *charges* defined in (2).

As shown in [2], the charge γ_i captures the compatibility between the actions of the algorithm for addressing flaw f_i and the measure μ. To see this, consider the probability, $\nu_i(\tau)$, of ending up in state τ after (i) sampling a state $\sigma \in f_i$ according to μ, and then (ii) addressing f_i at σ. Define the *distortion* associated with f_i as

$$d_i := \max_{\tau\in\Omega} \frac{\nu_i(\tau)}{\mu(\tau)} \geq 1 \ , \tag{4}$$

i.e., the maximum possible inflation of a state probability incurred by addressing f_i (relative to its probability under μ, and averaged over the initiating state $\sigma \in f_i$ according to μ). Now observe from (2) that

$$\gamma_i = \max_{\tau\in\Omega} \frac{1}{\mu(\tau)} \sum_{\sigma\in f_i} \mu(\sigma)\rho_i(\sigma, \tau) = d_i \cdot \mu(f_i) \ . \tag{5}$$

An algorithm for which $d_i = 1$ is called a *resampling oracle* [13] for f_i, and notice that it perfectly removes the conditional of the addressed flaw. However, designing resampling oracles for sophisticated measures can be impossible by local search. This is because small, but non-vanishing, correlations can travel arbitrarily far in Ω. Thus, allowing for some distortion can be very helpful, especially in cases where correlations decay with distance.

Remark 3.1. *Recalling the definition of matrix A_i in Definition 2.5 and letting $M = \mathrm{diag}(\mu(\sigma))$, we see that $\gamma_i = \|MA_iM^{-1}\|_1$. As shown in [3], this observation can be used to provide an alternative proof of Theorem 2.4 using the fact that any operator norm (and in particular the $\|\cdot\|_1$-norm) bounds the spectral radius of a matrix. Moreover, this linear algebraic point of view leads to significant generalizations of Theorem 2.4. We refer the reader to [3] for details.*

Theorem 3.1 below shows that Theorem 2.4 is the algorithmic counterpart of Theorem 2.1.

Theorem 3.1. *Given a family of flaws $F = \{f_1, \ldots, f_m\}$ over a state space Ω, an algorithm $\mathcal{A}$ with causality graph C with neighborhoods $\Gamma(\cdot)$, and a measure μ over Ω, then for each $S \subseteq F \setminus \Gamma(i)$ we have*

$$\mu\Big(f_i \mid \bigcap_{j\in S} \overline{f_j}\Big) \leq \gamma_i \ , \tag{6}$$

where the γ_i are the charges of the algorithm as defined in (2)*.*

Proof. Let $F_S := \bigcap_{j\in S} \overline{f_j}$. Observe that

$$\begin{aligned} \mu(f_i \mid F_S) &= \frac{\mu(f_i \cap F_S)}{\mu(F_S)} \\ &= \frac{\sum_{\sigma\in f_i\cap F_S} \mu(\sigma) \sum_{\tau\in a(i,\sigma)} \rho_i(\sigma,\tau)}{\mu(F_S)} \\ &= \frac{\sum_{\sigma\in f_i\cap F_S} \mu(\sigma) \sum_{\tau\in F_S} \rho_i(\sigma,\tau)}{\mu(F_S)} \ , \end{aligned} \tag{7}$$

where the second equality holds because each $\rho_i(\sigma,\cdot)$ is a probability distribution and the third by the definition of causality and the fact that $S \subseteq F \setminus \Gamma(i)$. Now notice that changing the order of summation in (7) gives

$$\begin{aligned} \frac{\sum_{\tau\in F_S} \sum_{\sigma\in f_i\cap F_S} \mu(\sigma)\rho_i(\sigma,\tau)}{\mu(F_S)} &= \frac{\sum_{\tau\in F_S} \mu(\tau) \sum_{\sigma\in f_i\cap F_S} \frac{\mu(\sigma)}{\mu(\tau)}\rho_i(\sigma,\tau)}{\mu(F_S)} \\ &\leq \frac{\sum_{\tau\in F_S} \mu(\tau) \left(\max_{\tau'\in\Omega} \sum_{\sigma\in f_i} \frac{\mu(\sigma)}{\mu(\tau')}\rho_i(\sigma,\tau')\right)}{\mu(F_S)} \\ &= \gamma_i \ . \end{aligned}$$

□

In words, Theorem 3.1 shows that causality graph C is a lopsidependency graph with respect to measure μ with $b_i = \gamma_i$ for all $i \in [m]$. Thus, when designing an algorithm for an application of Theorem 2.1 using Theorem 3.1, we have to make sure that the induced causality graph coincides with the lopsidependency graph, and that the measure distortion induced when addressing flaw f_i is sufficiently small so that the resulting charge γ_i is bounded above by b_i.

4 Edge Coloring Multigraphs: Proof of Theorem 1.3

We follow the exposition of the proof of Kahn in [26]. The key to the proof of Theorem 1.3 is the following lemma.

Lemma 4.1. *For all $\epsilon > 0$, there exists $\chi_0 = \chi_0(\epsilon)$ such that if $\chi_e^*(G) \geq \chi_0$ then we can find $N = \lfloor \chi_e^*(G)^{\frac{3}{4}} \rfloor$ matchings in G whose deletion leaves a multigraph G' with $\chi_e^*(G') \leq \chi_e^*(G) - (1+\epsilon)^{-1}N$ in expected $\mathrm{poly}(n, \ln \frac{1}{\epsilon})$ time with probability at least $1 - \frac{1}{n^c}$, for any constant $c > 0$.*

Using the algorithm of Lemma 4.1 recursively, for every $\epsilon > 0$ we can efficiently find an edge coloring of G using at most $(1+\epsilon)\chi_e^* + \chi_0$ colors as follows. First, we compute $\chi_e^*(G)$ using the algorithm of Padberg and Rao. If $\chi_e^* \geq \chi_0$, then we apply Lemma 4.1 to get a multigraph G' with $\chi_e^*(G') \leq \chi_e^*(G) - (1+\epsilon)^{-1}N$. We can now color G' recursively using at most $(1+\epsilon)\chi_e^*(G') + \chi_0 \leq (1+\epsilon)\chi_e^*(G) - N + \chi_0$ colors. Using one extra color for each one of the N matchings promised by Lemma 4.1, we can then complete the coloring of G, proving the claim. In the base case where $\chi_e^*(G) < \chi_0$, we color G greedily using $2\Delta - 1$ colors. The fact that $2\Delta - 1 \leq 2\chi_e^* - 1 < \chi_e^* + \chi_0$ concludes the proof of Theorem 1.3 since the number of recursive calls is at most n.

4.1 The Algorithm

Observe that we only need to prove Lemma 4.1 for $\epsilon < \frac{1}{10}$ since, clearly, if it holds for ϵ then it holds for all $\epsilon' > \epsilon$. So we fix $\epsilon \in (0, 0.1)$ and let $c^* = \chi_e^* - (1+\epsilon)^{-1}N$. Our goal will be to delete N matchings from G to get a multigraph G' which has fractional chromatic index at most c^*.

The flaws. Let $\Omega = \mathcal{M}(G)^N$ be the set of possible N-tuples of matchings of G. For a state $\sigma = (M_1, \ldots, M_N) \in \Omega$ let G_σ denote the multigraph induced by deleting the N matchings $M_1, \ldots, M_N$ from G. For a vertex $v \in V(G_\sigma)$ we define $d_{G_\sigma}(v)$ to be the degree of v in G_σ. We now define the following flaws. For every vertex $v \in V(G)$ let

$$f_v = \left\{ \sigma \in \Omega : d_{G_\sigma}(v) > c^* - \frac{\epsilon}{4}N \right\} \ .$$

For every connected subgraph H of G with an odd number of vertices, let

$$f_H = \left\{ \sigma \in \Omega : H \subseteq G_\sigma, |V(H)| \leq \frac{\Delta}{(\epsilon/4)N} \text{ and } |E(H)| > \left(\frac{|V(H)|-1}{2} \right) c^* \right\} \ .$$

The following lemma states that it suffices to find a flawless state.

Lemma 4.2 ([18])**.** *Any flawless state σ satisfies $\chi_e^*(G_\sigma) \leq c^*$.*

Proof. Edmonds' characterization [8] of the matching polytope implies that the chromatic index of G_σ is less than c^* if

1. $\forall v : d_{G_\sigma}(v) \leq c^*$; and
2. $\forall H \subseteq G_\sigma$ with an odd number of vertices: $E(H) \leq \frac{|V(H)|-1}{2} c^*$.

Now clearly, addressing every flaw of the form f_v establishes condition 1. By summing degrees this also implies that for every subgraph F with an even number of vertices $|E(F)| \leq \left(\frac{|V(F)|}{2} \right) c^*$.

Moreover, any odd subgraph H can be split into a connected component H' with an odd number of vertices, and a subgraph F with an even number of vertices. Thus, in the absence of f_v flaws, it suffices to prove condition 2 for connected H. Again by summing degrees, we see that if no f_v flaw is present, then condition 2 can fail only for H with fewer than $\frac{\Delta}{(\epsilon/4)N}$ vertices, concluding the proof. □

To describe an efficient algorithm for finding flawless states we need to (i) determine the initial distribution of the algorithm and show that is efficiently samplable; (ii) show how to address each flaw efficiently; (iii) show that the expected number of steps of the algorithm is polynomial; and finally (iv) show that we can search for flaws in polynomial time, so that each step is efficiently implementable.

The initial distribution. Let $\delta = \frac{\epsilon}{4}$ and apply Theorem 2.9. Let ν be the promised hard-core probability distribution, $\lambda = \{\lambda(e)\}$ the vector of activities associated with it, and K the corresponding constant. Note that the activities $\lambda(e)$ defining ν are not readily available. However, the next lemma says that we can efficiently compute a set of activities that gives an arbitrarily good approximation to the desired distribution ν.

Lemma 4.3. *For every $\eta > 0$, there exists a $\mathrm{poly}(n, \ln\frac{1}{\eta}, \ln\frac{1}{\delta})$-time algorithm that computes a set of edge activities $\{\lambda'(e)\}_{e\in E(G)}$ such that the corresponding hard-core distribution ν' satisfies $\|\nu - \nu'\|_{\mathrm{TV}} \leq \eta$.*

Proof. Lemma 4.3 is a straightforward corollary of the main results of Singh and Vishnoi [36] and Jerrum and Sinclair [15]. Briefly, the main result of [36] states that finding a distribution that apprxoximates ν can be seen as the solution of a max-entropy distribution estimation problem which can be efficiently solved given a "generalized counting oracle" for ν. The latter is provided by [15]. □

For a parameter $\eta > 0$ and a distribution p, we say that we *η-approximately sample* from p to express that we sample from a distribution $\tilde{p}$ such that $\|p - \tilde{p}\|_{\mathrm{TV}} \leq \eta$. Set $\eta = \frac{1}{n^C}$, where C is a sufficiently large constant to be specified later, and let ν' be the distribution promised by Lemma 4.3. The initial distribution of our algorithm, θ, is obtained by η-approximately sampling N random matchings (independently) from ν'. Observe that $\|\theta - \mu\|_{\mathrm{TV}} \leq 2\eta N$, where μ denotes the probability distribution over Ω induced by taking N independent samples from ν.

Addressing flaws. For an integer $d > 0$ and a connected subgraph H let $S_{<d}(H)$ be the set of vertices within distance strictly less than d of a vertex $u \in V(H)$.

We consider the procedure RESAMPLE below which takes as input a connected subgraph H, a state σ and a positive integer $d \leq n$, and which will be used to address flaws.

---

procedure RESAMPLE(H, σ, d)
  Let $\sigma = (M_1, M_2, \ldots, M_N)$
  for $i = 1$ to N **do**
    Let $E_{i,\geq d}$ be the set of edges of M_i that do not belong to the multigraph induced by $S_{<d+1}(H)$
    Let $E_{i,=d}$ be the set of edges of M_i whose both endpoints are in distance exactly d from H
    Let $V_{i,d}$ be the set of vertices of $S_{<d+1}(H)$ that belong to edges in $E_{i,\geq d} \cup E_{i,=d}$
    Let $G_{i,<d+1}$ be the multigraph induced by $S_{<d+1}(H) \setminus V_{i,d}$
    Let p be the hard-core distribution induced by $\{\lambda'(e)\}_{e\in E(G_{i,<d+1})}$.
    η-approximately sample a matching M from p
    Let $M'_i = (M_i \cap E_{i,\geq d}) \cup M$ ▷ By definition, M'_i is a matching
  Output $\sigma' = (M'_1, M'_2, \ldots, M'_N)$

Notice that Theorem 2.10 implies that procedure RESAMPLE (H, σ, d) terminates in $\mathrm{poly}(n, \ln\frac{1}{\eta})$ time.

Set $t = 8(K+1)^2\delta^{-1} + 2$. To address f_v, f_H at state σ, we invoke procedures RESAMPLE $(\{v\}, \sigma, t)$ and RESAMPLE (H, σ, t), respectively.

Searching for flaws. Notice that we can compute c^* in polynomial time using the algorithm of Padberg and Rao [30]. Therefore, given a state $\sigma \in \Omega$ and c^*, we can search for flaws of the form f_v in polynomial time. However, the flaws of the form f_H are potentially exponentially many, so a brute-force search does not suffice for our purposes.

Fortunately, the result of Padberg and Rao essentially provides a polynomial time oracle for this problem as well. Recall Edmonds' characterization used in the proof of Lemma 4.2. The constraints over odd subgraphs H are called *matching constraints*. Recall further that in the proof of Lemma 4.2 we showed that,

in the absence of f_v-flaws, the only matching constraints that could possibly be violated correspond to f_H flaws. On the other hand, the oracle of Padberg and Rao, given as input $(\frac{1}{c}, \ldots, \frac{1}{c})$ and a multigraph G, can decide in polynomial time whether G has a fractional c-coloring or return a violated matching constraint. Hence, if our algorithm prioritizes f_v flaws over f_H flaws, this oracle can be used to detect the latter in polynomial time.

4.2 Proof of Lemma 4.1

We are left to show that the expected number of steps of the algorithm is polynomial and that each step can be executed in polynomial time. To that end, we will show that both of these statements are true assuming that the initial distribution θ is μ instead of approximately μ, and that in Lines 8, 9 of the procedure RESAMPLE(H, σ, d) we perfectly sample from the hard-core probability distribution induced by activities $\{\lambda(e)\}_{e \in E(G_{i,<d}(H))}$ instead of η-approximately sampling from p. Observe that, since we will prove that in this case the expected running time of the ideal algorithm is polynomial, we can maximally couple the approximate and ideal distributions, and then take the constant C in the definition of the approximation parameter η to be sufficiently large. The latter implies that the probability that the coupling will fail during the execution of the algorithm is negligible (i.e., at most $\frac{1}{n^c}$), establishing that the algorithm converges even if we use approximate distributions.

For an integer $d > 0$ and a vertex v, let $S^*_d(v)$ be the set of flaws indexed by a vertex of $S_{<d}(v)$ or a set H intersecting $S_{<d}(v)$. For each set H for which we have defined f_H we let $S^*_d(H) = \bigcup_{v \in V(H)} S^*_d(v)$. For each flaw f_v we define the causality neighborhood $\Gamma(f_v) = S^*_{t+2}(v)$ and for each flaw f_H we define $\Gamma(f_H) = S^*_{t+2}(H)$, where t is as defined in the previous subsection. Notice that this is a valid choice because flaw f_v can only cause flaws in $S^*_{t+1}(v)$ and flaw f_H can only cause flaws in $S^*_{t+1}(H)$. The reason why we choose these neighborhoods to be larger than seemingly necessary is because, as we will see, with respect to this causality graph our algorithm is commutative, allowing us to apply Theorem 2.6.

Lemma 4.4. *Let $f \in \{f_v, f_H\}$ for a vertex v and a connected subgraph H of G with an odd number of vertices and let $D = \Delta^{t+\Delta^{\frac{1}{3}}+4}$. For every $\zeta > 0$ there exists Δ_ζ such that if $\Delta \geq \Delta_\zeta$ then*

(a) $\gamma_f \leq \frac{1-\zeta}{\mathrm{e}D}$;

(b) $|\Gamma(f)| \leq D$,

where the charges are computed with respect to the measure μ and the algorithm that samples from the ideal distributions.

The proof of Lemma 4.4 can be found in Section 4.3. Lemma 4.5 establishes that our algorithm is commutative with respect to the causality relation $\sim$ induced by neighborhoods $\Gamma(\cdot)$. Its proof can be found in Section 4.4.

Lemma 4.5. *For each pair of flaws $f \nsim g$, the matrices A_f, A_g commute.*

Setting $x_f = \frac{1}{1+\max_{f' \in F} |\Gamma(f')|}$ for each flaw f, we see that condition (3) with $\epsilon = \zeta/2$ is implied by

$$\gamma_f \cdot \left(1 + \max_{f' \in F}(|\Gamma(f')|\right) \cdot \mathrm{e} \leq 1 - \zeta/2 \text{ for every flaw } f \ , \tag{8}$$

which is true for large enough Δ according to Lemma 4.4. Notice further that the causality graph induced by $\sim$ can be partitioned into n cliques, one for each vertex of G, with potentially further edges between them. Indeed, flaws indexed by subgraphs that contain a certain vertex of G form a clique in the causality graph. Combining Lemma 4.5 with the latter observation, we are able to apply Theorem 2.6 which implies that our

algorithm terminates after an expected number of at most $O\big(\max_{\sigma\in\Omega}\frac{\theta(\sigma)}{\mu(\sigma)}\cdot\frac{n}{\zeta}\log\frac{n\log(1/\delta)}{\zeta}\big)=O(n\log n)$ steps. (This is because we assume that $\theta=\mu$ per our discussion above.)

This completes the proof of Lemma 4.1 and hence, as explained at the beginning of Section 4, Theorem 1.3 follows. It remains, however, to go back and prove Lemmas 4.4 and 4.5, which we do in the next two subsections.

4.3 Proof of Lemma 4.4

In this section we prove Lemma 4.4. Given a state $\sigma=(M_1,\ldots,M_N)$, a subgraph H, and $d>0$ let

$$Q_H(d,\sigma)=(M_1-S_{<d}(H),M_2-S_{<d}(H),\ldots,M_N-S_{<d}(H))\quad,$$

where we define $M-X=M\cap E(G-X)$. Moreover, let $Q^i_H(d,\sigma)=M_i-S_{<d}(H)$ denote the i-th entry of $Q_H(d,\sigma)$. Finally, let $G_{<d+1}(H)$ be the multigraph induced by $S_{<d+1}(H)$ and $\mathcal{M}^i_{d+1}(H,\sigma)$ be the set of matchings of $G_{<d+1}(H)$ that are compatible with $Q^i_H(d,\sigma)$. That is, for any matching M in $\mathcal{M}^i_{d+1}(H,\sigma)$ we have that $M\cup Q^i_H(d,\sigma)$ is also a matching of G.

Remark 4.1. *Recall the definition of the multigraph $G_{i,<d+1}$ in Line 7 of procedure* RESAMPLE *and observe that the set of matchings $\mathcal{M}^i_{d+1}(H,\sigma)$ is exactly the set of matchings of this multigraph. As we saw earlier, this implies that any hard-core distribution over $\mathcal{M}^i_{d+1}(H,\sigma)$ is efficiently samplable via the algorithm of [15]. We introduce this equivalent definition of $\mathcal{M}^i_{d+1}(H,\sigma)$ here because it will be convenient in defining events with respect to the probability space induced by μ.*

Proof of part (a). We will need the following key lemma, which was essentially proved in [18]. Its proof can be found in Appendix A. Recall that μ is the distribution over Ω induced by taking N independent samples from ν.

Lemma 4.6. *For every $\zeta>0$ there exists Δ_ζ such that if $\Delta\geq\Delta_\zeta$ then for any random state σ distributed according to μ,*

(i) for every flaw f_v and state $\tau\in\Omega$: $\mu(\sigma\in f_v\mid Q_v(t,\sigma)=Q_v(t,\tau))\leq\frac{1-\zeta}{eD}$, and

(ii) for every flaw f_H and state $\tau\in\Omega$: $\mu(\sigma\in f_H\mid Q_H(t,\sigma)=Q_H(t,\tau))\leq\frac{1-\zeta}{\mathrm{e}D}$.

We show the proof of part (a) of Lemma 4.4 only for the case of f_v- flaws, as the proof for f_H- flaws is very similar. Specifically, our goal will be to prove that

$$\gamma_{f_v}=\max_{\tau\in\Omega}\mu(\sigma\in f_v\mid Q_v(t,\sigma)=Q_v(t,\tau))\quad. \tag{9}$$

Lemma 4.6 then concludes the proof.

Let $x_v(\sigma)=(x_{v,1}(\sigma),\ldots,x_{v,N}(\sigma))$ denote the vector such that $x_{v,i}(\sigma)=|M_i\cap E_v|$, where E_v is the set of edges adjacent to v. Notice that $x_{v,i}(\sigma)\leq 1$ since M_i is a matching. For a vector $x\in\{0,1\}^N$ define $O(x):=\{i\in[N]:x_i=1\}$ and observe that $\sigma\in f_v$ iff $|O(x_v(\sigma))|<d_G(v)-c^*+\frac{\epsilon}{4}N$. Define the set $X_v=\{x\in\{0,1\}^N:x=x_v(\sigma)\text{ for some }\sigma\in f_v\}$ and notice that the latter observation implies that $\sigma\in f_v$ iff $x_v(\sigma)\in X_v$. (In other words, the elements of X_v induce a partition of f_v.) Hence, for a fixed state $\tau\in\Omega$ and a random sample σ from μ, we have

$$\mu(\sigma\in f_v\mid Q_v(t,\sigma)=Q_v(t,\tau))=\sum_{x\in X_v}\prod_{i=1}^{N}\nu\left(x_{v,i}(\sigma)=x_i\mid Q^i_v(t,\sigma)=Q^i_v(t,\tau)\right)\quad, \tag{10}$$

since μ corresponds to N independent samples from ν. Recall that ν is associated with a set of activities $\{\lambda(e)\}_{e\in E}$. Thus, for any vector $x \in X_v$, we obtain

$$\begin{aligned}
\nu\left(x_{v,i}(\sigma) = x_i \mid Q_v^i(t,\sigma) = Q_v^i(t,\tau)\right) &= \frac{\nu\left((x_{v,i}(\sigma) = x_i) \cap \left(Q_v^i(t,\sigma) = Q_v^i(t,\tau)\right)\right)}{\nu\left(Q_v^i(t,\sigma) = Q_v^i(t,\tau)\right)} \\
&= \frac{\sum_{M:|M\cap E_v|=x_i,(M-S_{<t}(v))=Q_v^i(t,\tau)} \lambda(M)}{\sum_{M:(M-S_{<t}(v))=Q_v^i(t,\tau)} \lambda(M)} \\
&= \frac{\sum_{M\in\mathcal{M}_{t+1}^i(v,\tau),|M\cap E_v|=x_i} \lambda(M)}{\sum_{M\in\mathcal{M}_{t+1}^i(v,\tau)} \lambda(M)} \ , \qquad (11)
\end{aligned}$$

where recall that $\mathcal{M}_{t+1}^i(v,\tau)$ denotes the set of matchings of $G_{<t+1}(v)$ that are compatible with $Q_v^i(t,\tau)$. To get (11) we used the form of $\lambda(M)$ to cancel the contributions of edges in $Q_v^i(t,\tau)$.

We will use (10) and (11) to prove that, for σ distributed according to μ, and any state $\tau \in \Omega$,

$$\sum_{\omega\in f_v} \frac{\mu(\omega)}{\mu(\tau)} \rho_{f_v}(\omega,\tau) = \mu(\sigma \in f_v \mid Q_v(t,\sigma) = Q_v(t,\tau)) \ . \qquad (12)$$

According to the definition of γ_{f_v}, maximizing (12) over $\tau \in \Omega$ yields (9) and completes the proof.

Fix $\tau = (M_1, M_2, \ldots, M_N) \in \Omega$. To compute the sum on the left-hand side of (12) we need to determine the set of states $\mathrm{In}_v(\tau) \subseteq f_v$ for which $\rho_{f_v}(\omega,\tau) > 0$. To do this, recall that given as input a state $\omega = (M_1^\omega, M_2^\omega, \ldots, M_N^\omega) \in f_v$, procedure RESAMPLE$(v,\omega,t)$ modifies one by one each matching M_i, $i \in [N]$, "locally" around v. In particular, observe that the support of the distribution for updating M_i is exactly the set $\mathcal{M}_{t+1}^i(v,\omega)$ and, hence, it has to be that $Q_v^i(t,\omega) = Q_v^i(t,\tau)$ for every $i \in [N]$ and state $\omega \in \mathrm{In}_v(\tau)$. This also implies that, for every such ω,

$$\frac{\mu(\omega)}{\mu(\tau)} = \prod_{i=1}^N \frac{\nu(M_i^\omega)}{\nu(M_i)} = \prod_{i=1}^N \frac{\lambda(M_i^\omega \cap E(G_{<t+1}(v)))}{\lambda(M_i \cap E(G_{<t+1}(v)))} \ . \qquad (13)$$

Recall now that we have assumed that the hard-core distribution in Lines 8, 9 of RESAMPLE (v,ω,t) is induced by the ideal vector of activities λ. In particular, we have

$$\rho_{f_v}(\omega,\tau) = \prod_{i=1}^N \frac{\lambda(M_i \cap E(G_{<t+1}(v)))}{\sum_{M\in\mathcal{M}_{t+1}^i(v,\omega)} \lambda(M)} = \prod_{i=1}^N \frac{\lambda(M_i \cap E(G_{<t+1}(v)))}{\sum_{M\in\mathcal{M}_{t+1}^i(v,\tau)} \lambda(M)} \qquad (14)$$

since $Q_v^i(t,\omega) = Q_v^i(t,\tau)$, which combined with (13) yields

$$\frac{\mu(\omega)}{\mu(\tau)} \rho_{f_v}(\omega,\tau) = \prod_{i=1}^N \frac{\lambda(M_i^\omega \cap E(G_{<t+1}(v)))}{\sum_{M\in\mathcal{M}_{t+1}^i(v,\tau)} \lambda(M)} \ . \qquad (15)$$

Finally, recall that $X_v = \{x \in [0,1]^N : x = x_v(\omega) \text{ for some } \omega \in f_v\}$, and specifically that $\omega \in f_v$ iff

$x_v(\omega) \in X_v$. For $x \in X_v$, let $\Omega_{v,x} = \{\omega : x_v(\omega) = x\}$. We now have

$$\begin{aligned}
\sum_{\omega \in f_v} \frac{\mu(\omega)}{\mu(\tau)} \rho_{f_v}(\omega, \tau) &= \sum_{x \in X_v} \sum_{\omega \in \Omega_{v,x}} \frac{\mu(\omega)}{\mu(\tau)} \rho_{f_v}(\omega, \tau) \\
&= \sum_{x \in X_v} \sum_{\omega \in \Omega_{v,x}} \prod_{i=1}^{N} \frac{\lambda(M_i^\omega \cap E(G_{<t+1}(v)))}{\sum_{M \in \mathcal{M}_{t+1}^i(v,\tau)} \lambda(M)} \qquad (16) \\
&= \sum_{x \in X_v} \prod_{i=1}^{N} \sum_{\substack{\omega \in \Omega_{v,x} \\ x_{v,i}(\omega) = x_i}} \frac{\lambda(M_i^\omega \cap E(G_{<t+1}(v)))}{\sum_{M \in \mathcal{M}_{t+1}^i(v,\tau)} \lambda(M)} \qquad (17) \\
&= \sum_{x \in X_v} \prod_{i=1}^{N} \frac{\sum_{M \in \mathcal{M}_{t+1}^i(v,\tau), |M \cap E_v| = x_i} \lambda(M)}{\sum_{M \in \mathcal{M}_{t+1}^i(v,\tau)} \lambda(M)} \ . \qquad (18)
\end{aligned}$$

To get (16) we used (15). For (17) we used the fact that Ω is the product space $\mathcal{M}(G)^N$, so that the choices per matching are independent, while for (18) we used the definition of $x_{v,i}(\omega)$.

Combining (18) with (10) and (11) establishes (12), concluding the proof. □

Proof of part (b). To see part (b) of Lemma 4.4, first notice that every set $S_{<t+2}(v)$ has at most Δ^{t+2} elements. Moreover, the fact that $N = \lfloor \chi_e^*(G)^{3/4} \rfloor = \Theta(\Delta^{3/4})$ implies that $\frac{\Delta}{(\epsilon/4)N} \leq \Delta^{\frac{1}{3}}$ for sufficiently large Δ. So, every vertex u is in at most $\Delta^{\Delta^{\frac{1}{3}}}$ sets H corresponding to a flaw f_H. Hence, every $S_{t+2}^*(v)$ has at most $\Delta^{t+\Delta^{\frac{1}{3}}+3}$ elements. Thus, since every H for which we define $S_{t+2}^*(H)$ has fewer than Δ vertices, every $S_{t+2}^*(H)$ has less than $D = \Delta^{t+\Delta^{\frac{1}{3}}+4}$ elements. □

4.4 Proof of Lemma 4.5

Fix $\sigma_1 = (M_1, M_2, \ldots, M_N) \in f$ and $\sigma_2 = (M_1', M_2', \ldots, M_N') \in g$ such that $f \not\sim g$. To prove that the matrices A_f, A_g commute, we need to show that for every such pair

$$\sum_{\tau} \rho_f(\sigma_1, \tau) \rho_g(\tau, \sigma_2) = \sum_{\tau} \rho_g(\sigma_1, \tau) \rho_f(\tau, \sigma_2) \ . \qquad (19)$$

To that end, let H_f, H_g be the subgraphs (which may consist only of a single vertex) associated with flaws f and g, respectively. Since $f \nsim g$ we have that $\min_{u \in V(H_f), v \in V(H_g)} \mathrm{dist}(u, v) \geq t + 2$, where $\mathrm{dist}(u, v)$ denotes the length of the shortest path between u and v. Notice that this implies that $S_{<t+2}(H_f) \cap S_{<t+2}(H_g) = \emptyset$.

Consider a pair of transitions $\sigma_1 \xrightarrow{f} \tau$, $\tau \xrightarrow{g} \sigma_2$, where $\tau = (M_1'', \ldots, M_N'')$, and so that $\rho_f(\sigma_1, \tau) > 0$, $\rho_g(\tau, \sigma_2) > 0$. The facts that procedure RESAMPLE (σ, f, t) only modifies the input set of matchings locally within $S_{<t+1}(H_f)$, that $\rho_g(\tau, \sigma_2) > 0$, and that $S_{<t+2}(H_f) \cap S_{<t+2}(H_g) = \emptyset$ imply that (i) $\sigma_1 \in g$; and (ii) for every $i \in [N]$, $M_i \cap (S_{<t+2}(H_g)) = M_i'' \cap (S_{<t+2}(H_g))$. Notice now that the probability distribution $\rho_g(\tau, \cdot)$ depends only on $(M_1'' \cap S_{<t+2}(H_g), \ldots, M_N'' \cap S_{t+2}(H_g))$. Hence, (i) and (ii) imply that the probability distribution $\rho_g(\sigma_1, \cdot)$ is well defined and, in addition, there exists a natural bijection b_g between the action set $a(g, \tau)$ and the action set $a(g, \sigma_1)$ so that $\rho_g(\tau, \tau') = \rho_g(\sigma_1, b_g(\tau'))$ for every $\tau' \in a(g, \tau)$. This is because both distributions are implemented by sampling from the set of matchings of the same multigraph, according to the same probability distribution.

Now let $\tau' = b_g(\sigma_2)$. A symmetric argument implies that $\tau' \in f$ and that there exists a natural bijection b_f between $a(f, \sigma_1)$ and $a(f, \tau')$ so that $\rho_f(\sigma_1, \sigma) = \rho_f(\tau', b_f(\sigma))$ for every $\sigma \in a(f, \sigma_1)$. In particular,

notice that $\sigma_2 = b_f(\tau)$ and that

$$\rho_f(\sigma_1, \tau)\rho_g(\tau, \sigma_2) = \rho_g(\sigma_1, \tau')\rho_f(\tau', b_f(\tau)) = \rho_g(\sigma_1, \tau')\rho_f(\tau', \sigma_2) \ . \tag{20}$$

Overall, what we have shown is a bijective mapping that sends any pair of transitions $\sigma_1 \xrightarrow{f} \tau, \tau \xrightarrow{g} \sigma_2$ to a pair of transitions $\sigma_1 \xrightarrow{g} \tau', \tau' \xrightarrow{f} \sigma_2$ and which satisfies (20). This establishes (19), concluding the proof. □

5 List-Edge Coloring Multigraphs: Proof of Theorem 1.4

In this section we review the proof of Theorem 1.2 and then prove its constructive version, Theorem 1.4.

5.1 A High Level Sketch of the Existential Proof

As we explained in the introduction, the non-constructive proof of Theorem 1.2 is a sophisticated version of the semi-random method and proceeds by partially coloring the edges of the multigraph in iterations, until at some point the coloring can be completed greedily. (More accurately, the method establishes the *existence* of such a sequence of desirable partial colorings.)

We will follow the exposition in [26]. In each iteration, we have a list L_e of acceptable colors for each edge e. We assume that each L_e originally has C colors for some $C \geq (1+\epsilon)\chi_e^*(G)$, where $\epsilon > 0$ is an arbitrarily small constant. For each color i, we let G_i be the subgraph of G formed by the edges for which i is acceptable. Since $G_i \subseteq G, \chi_e^*(G_i) \leq \chi_e^*(G)$. Thus, Theorem 2.9 implies that we can find a hard-core distribution on the matchings of G_i with marginals $(\frac{1}{C}, \ldots, \frac{1}{C})$ whose activity vector λ_i satisfies $\lambda_i(e) \leq \frac{K}{C}$ for all e, where $K = K(\epsilon)$ is a constant.

In each iteration, we will use the *same* activity vector λ_i to generate the random matchings assigned to color i. Of course, in each iteration we restrict our attention to the subgraph of G_i obtained by deleting the set E^* of edges colored (with any color) in previous iterations, and the endpoints of the set of edges E_i^* colored i in previous iterations. (Thus, although we use the same activity vector for each color in each iteration, the induced hard-core distributions may vary significantly.) Further, we will make sure that our distributions have the property that for each edge e, the expected number of matchings containing e is very close to 1.

We apply the Lopsided LLL in the following probability space. For each color i, we choose a matching $M_i \in G_i$ from the corresponding distribution, with these choices made independently. Next, we *activate* each edge in M_i independently with probability $\alpha := \frac{1}{\log \Delta(G)}$; we assign colors only to activated edges in order to ensure that very few edges are assigned more than one color. We then update the multigraph by deleting the colored edges, and update the lists L_e by deleting any color assigned to an edge incident to e. We give a more detailed description below.

Notice that our argument needs to ensure that (i) at the beginning of each iteration the induced hard-core distributions are such that, for each uncolored edge e, the expected number of random matchings containing e is very close to 1; and (ii) after some number of iterations, we can complete the coloring greedily.

As far as the latter condition is concerned, notice that if (i) holds throughout then, in each iteration, the probability that an edge retains a color remains close to the activation probability α. This allows us to prove that the maximum degree in the uncolored multigraph drops by a factor of about $1-\alpha$ in each iteration. Hence, after $\log_{\frac{1}{1-\alpha}} 3K$ iterations, the maximum degree in the uncolored multigraph will be less than $\frac{\Delta}{2K}$. Furthermore, for each e and i, the probability that e is in the random matching of color i is at most $\lambda_i(e) \leq \frac{K}{C}$. Since (i) continues to hold, this implies there are at least $\frac{C}{K} > \frac{\Delta}{K}$ colors available for each edge, and so the coloring can be completed greedily. (Recall that the $C > \chi_e^*(G) \geq \Delta$.)

An Iteration.

1. For each color i, pick a matching M_i according to a hard-core probability distribution μ_i on $\mathcal{M}(G_i)$ with activities λ_i such that for some constant K:

   (a) $\forall e \in E(G), \sum_i \mu_i(e \in M_i) \approx 1$

   (b) $\forall i, e \in E(G), \lambda_i(e) \leq \frac{K}{C}$ and hence $\forall v \in V(G), \sum_{L_e \ni i} \lambda_i(e) \leq K$.

2. For each i, activate each edge of M_i independently with probability $\alpha = \frac{1}{\log \Delta(G)}$, to obtain a matching F_i. We color the edges of F_i with color i and delete $V(F_i)$ from G_i. We also delete from G_i every edge not in M_i which is in F_j for some $j \neq i$. We do not delete edges of $(M_i - F_i) \cap F_j$ from G_i. (Note that this may result in edges receiving more than one color, which is not a problem since we can always pick one of them arbitrarily at the end of the iterative procedure.)

3. Note that the expected number of edges that are both colored and removed from G_i in Step 2 is less than $\alpha|E(G_i)|$ because, although the expected number of colors retained by an edge is very close to α, some edges may be assigned more than one color. As is standard in this kind of proof, we will perform an *equalizing coin flip* for each edge e of G_i so that the probability that e is both colored and removed from G_i in either Step 2 or Step 3 is exactly α.

The *outcome* of an iteration is defined to be the choices of matchings, activations, and equalizing coin flips. Let $Q = Q_\ell$ denote the random variable that equals the outcome of the ℓ-th iteration. (In what follows, we will focus on a specific iteration ℓ and so we will omit the subscript.)

For each edge $e = (u, v)$, we define a bad event A_e as follows. Let G'_i be the multigraph obtained after carrying out the modifications to G_i in Steps 2 and 3 of the above iteration. Let $t' = 8(K+1)^2(\log \Delta)^{20} + 2$ and recall the definition of $S_{<t'}(H)$ for subgraph H. Let Z_i be a random matching in $G'_i \cap S_{<t'}(\{u, v\})$ sampled from the hard-core probability distribution induced by activity vector λ_i. Let A_e be the event that

$$\Big| \sum_{i: G'_i \ni e} \Pr(e \in Z_i \mid Q) - \sum_{i: G_i \ni e} \Pr(e \in M_i) \Big| > \frac{1}{2(\log \Delta)^4} \ . \tag{21}$$

To get some intuition behind the definition of event A_e, let M'_i be a random matching in G'_i chosen according to the hard-core distribution with activities λ_i. Since correlations decay with distance, one can show that $\Pr(e \in M'_i \mid Q)$ is within a factor of $1 + \frac{1}{(\log \Delta)^{20}}$ of $\Pr(e \in Z_i \mid Q)$. Thus, according to (21), avoiding bad event A_e implies that $\sum_i \Pr(e \in M'_i) \approx \sum_i \Pr(e \in M_i) \approx 1$, which is what is required in order to maintain property (i) at the beginning of the next iteration. In particular, it is straightforward to see that avoiding all bad events $\{A_e\}_{e \in E(G)}$ guarantees that

$$\Big| \sum_{i: G'_i \ni e} \Pr(e \in M'_i \mid Q) - \sum_{i: G_i \ni e} \Pr(e \in M_i) \Big| \leq \frac{1}{(\log \Delta)^4} \ , \tag{22}$$

for sufficiently large Δ, which is what we really need. The reason we consider Z_i and not M'_i is that events defined with respect to the former are mildly negatively correlated with most other bad events, making it possible to apply the Lopsided LLL.

Further, for each vertex v we define A_v to be the event that the proportion of edges incident to v which are colored in the iteration is less than $\alpha - \frac{1}{(\log \Delta)^4}$.

It can be formally shown that, if we avoid all bad events, then (i) holds, i.e., at the beginning of the next iteration we can choose new probability distributions so that for each uncolored edge e we maintain the property that the expected number of random matchings containing e is very close to 1, and, moreover, after $\log_{\frac{1}{1-\alpha}} 3K$ iterations we can complete the coloring greedily.

Theorem 5.1 ([19]). *Assume that* (22) *holds for the edge marginals of the matching distributions of iteration ℓ. Then, with positive probability, the same is true for the matching distributions of iteration $\ell + 1$.*

Theorem 5.2 ([19]). *If we can avoid the bad events of the first $\log_{\frac{1}{1-\alpha}} 3K$ iterations, then we can complete the coloring greedily.*

Proving Theorems 5.1, 5.2 is the heart of the proof of Theorem 1.2. The most difficult part is proving that for any $x \in V \cup E$ the probability of event A_x is very close to 0 conditioned on any choice of outcomes for distant events. (This is needed in order to apply the Lopsided LLL.) Below we state the key lemma that is proven in [19], and which we will also use in the analysis of our algorithm.

Recall the definition of t' and let $t = (t')^2$. For a subgraph H, we let R_H be the random outcome of our iteration in $G - S_{<t}(H)$, i.e., R_H consists of $\bigcup_i (M_i - S_{<t}(H))$, together with the choices of the activated edges in $G - S_{<t}(H)$ which determine the $\bigcup_i (F_i - S_{<t}(H))$, and the outcomes of the equalizing coin flips for edges in this subgraph.

Lemma 5.3 ([19]). *For every $x \in E \cup V$ and possible choice R_x^* for R_x, there exists Δ_0 such that if $\Delta \geq \Delta_0$, then $\Pr(A_x \mid R_x = R_x^*) \leq \frac{1}{\Delta^{3(t+t'+2)}}$.*

In the next sections we will focus on providing an efficient algorithm for Theorem 5.1 which, combined with Theorem 5.2, will imply the proof of Theorem 1.4.

As a final remark, we note that detecting whether bad events $\{A_e\}_{e \in E(G)}$ are present in a state is not a tractable task since it entails the exact computation of edge marginals of hardcore distributions over matchings. In order to overcome this obstacle, we will define flaws $\{f_e\}_{e \in E(G)}$ whose absence provides somewhat weaker guarantees than ridding of bad events $\{A_e\}_{e \in E(G)}$, but nonetheless implies (22) for every edge. To decide whether flaw a f_e is present in a state, we will we use the results of [15] to estimate the corresponding edge marginals of random variables M_i and Z_i for every color i. Note that since we will only perform an approximation, there is the possibility to deduce that f_e is not present while in reality it is. However, our approximation will be tight enough so that, even in this case, (22) will still hold for every edge. We give the details forthwith.

5.2 The Algorithm

Let $\mathcal{U}$ denote the set of uncolored edges and $N = |\bigcup_{e \in \mathcal{U}} L_e|$, the cardinality of the set of colors that appear in the list of available colors of some uncolored edge. For a color $i \in [N]$, recall that G_i denotes the subgraph of uncolored edges that contain i in their list of available colors. Finally, let $E_i = |E(G_i)|$.

Define $\Omega = \prod_{i \in [N]} \left(\mathcal{M}(G_i) \times \{0,1\}^{E_i} \times \{0,1\}^{E_i}\right)$. We consider an arbitrary but fixed ordering over $\mathcal{U}$, so that each state $\sigma \in \Omega$ can be represented as $\sigma = ((M_1, a_1, h_1), \ldots, (M_N, a_N, h_N))$, where M_i, a_i, h_i are the matching, activation and equalizing coin flip vectors, respectively, that correspond to color i, so that edge e is activated in G_i if $a_i(e) = 1$ and is marked to be removed if $h_i(e) = 1$.

Recall that for color i we choose a matching according to probability distribution μ_i and we define $\mathrm{Eq}_i(e)$ to be the probability of success of the equalizing coin flip that corresponds to edge e and color i. Note that, given access to the marginals of μ_i, the value of $\mathrm{Eq}_i(e)$ can be computed efficiently. (Of course, we will have only (arbitrarily good) estimates of the marginals of μ_i, but as in the proof of Theorem 1.3, this suffices for our purposes.)

We let μ be the probability distribution over Ω that is induced by the product of the μ_i's, activation flips, and equalizing coin flips for each color i. In other words, μ is the probability distribution over Ω induced by the iteration.

The initial distribution. Recall that each edge e initially has a list L_e of size at least $(1+\epsilon)\chi^*_e(G)$. As we have already seen in Lemma 4.3, the results of [15, 36] imply that for every color i and parameter $\eta > 0$, there exists a $\mathrm{poly}(n, \ln\frac{1}{\eta}, \ln\frac{1}{\epsilon})$-algorithm that computes a vector λ'_i such that the induced hard-core distribution η-approximates in variation distance the hard-core distribution induced by vector λ_i. Setting $\eta = \frac{1}{n^\beta}$ for β sufficiently large, let μ' be the distribution obtained in an identical way to μ but using vectors λ'_i instead of vectors λ_i. The initial distribution θ of our algorithm is obtained by η-approximately sampling from μ'. Theorem 2.10 implies that this can be done in polynomial time.

Finding and addressing flaws. We define a flaw f_v for each bad event A_v. Moreover, for each edge e we define flaw f_e to be the set of states $\sigma \in \Omega$ such that

$$\Big|\sum_{i:G'_i \ni e} \Pr(e \in Z_i \mid \sigma) - \sum_{i:G_i \ni e} \Pr(e \in M_i)\Big| > \frac{2}{3(\log\Delta)^4} \ . \tag{23}$$

We fix an arbitrary ordering π over $V \cup E$. In each step, the algorithm finds the lowest indexed flaw according to π that is present in the current state and addresses it.

Clearly, checking if vertex-flaws A_v are present in the current state can be done efficiently.

For edge indexed flaws, we use the results of [15] to approximate the edge marginals of the corresponding distributions within a factor $(1+\eta)$ with probability at least $1-\eta$, in time $\mathrm{poly}(n, \ln\frac{1}{\eta})$). Recalling that $\eta = \frac{1}{n^\beta}$ and taking β to be a sufficient large constant, we can subsume this error probability into the probability that our algorithm fails.

Moreover, since, as we have already mentioned, $\Pr(e \in M'_i \mid \sigma)$ is within a factor $1 + \frac{1}{(\log\Delta)^{20}}$ of $\Pr(e \in Z_i \mid \sigma)$, for Δ and β sufficiently large, deducing that flaw f_e is not present in a state σ using our estimates for the edge marginals implies that (22) holds for edge e at state σ. In other words, if our algorithm decides that it has fixed every flaw, we are guaranteed that (22) holds for its output, even if some flaws are in fact still present.

In the opposite direction, there is the possibility that our algorithm decides that a flaw f_e is present while in reality it is not. In particular, there is a danger that, due to approximation errors, our algorithm effectively attempts to get rid of supersets $\widetilde{f}_e \supseteq f_e$ of the original flaws we defined and, as a result, fails to converge efficiently. Nonetheless, using Lemma 5.3, together with the facts that our approximations can be made arbitrarily accurate and that $A_e \subseteq f_e$ for all $e \in E$, we can still conclude that $\mu(\widetilde{f}_e \mid R_e = R^*_e) \leq \Delta^{-3(t+t'+2)}$.

Summarizing, we may and will assume without loss of generality that we are able to accurately and efficiently search for edge-flaws f_e, and that their probability with respect to measure μ is bounded above by $\Delta^{-3(t+t'+2)}$ conditional on any instantiation of R_e.

Recall the definition of t and the procedure RESAMPLE described in Section 4.1. Below we describe procedure FIX that takes as input a subgraph H and a state σ. In the description of FIX below we invoke procedure RESAMPLE with an extra parameter, namely an activity vector λ'_i for each color i. By that we mean that in Lines 8, 9 of RESAMPLE we use the vector λ'_i to define p.

1: **procedure** FIX(H, σ)
2: Let $\sigma = ((M_1, b_1, h_1), (M_2, b_2, h_2) \ldots, (M_N, b_N, h_N))$
3: $(M'_1, M'_2, \ldots, M'_N) \leftarrow$ RESAMPLE$(H, (M_1, M_2, \ldots, M_N), t, \lambda'_i)$
4: **for** $i = 1$ to N **do**
5: Update a_i to a'_i by activating independently each edge in $G_i \cap S_{<t+1}(H)$ with probability α
6: Update h_i to h'_i by flipping the corresponding equalizing coin for each edge in $G_i \cap S_{<t+1}(H)$
7: Output $\sigma = ((M'_1, a'_1, h'_1), (M'_2, a'_2, h'_2), \ldots, (M'_N, a'_N, h'_N))$

Theorem 2.10 implies that procedure FIX runs in polynomial time for any input subgraph H and state σ. To address flaws $f_v, f_{\{u_1,u_2\}}$ in a state σ we invoke FIX$(\{v\},\sigma)$ and FIX$(\{u_1,u_2\},\sigma)$, respectively.

5.3 Proof of Theorem 1.4

Similarly to the proof of Theorem 1.3, for our analysis we will assume that our algorithm samples from the "ideal" distributions, i.e., the ones induced by the vectors λ_i, rather than by the approximate ones λ'_i. An identical argument shows that this is sufficient if we take the exponent β in the definition of η to be large enough.

For two flaws f_{x_1}, f_{x_2}, where $x_1, x_2 \in V \cup E$, we consider the causality relation $f_{x_1} \sim f_{x_2}$ iff $\operatorname{dist}(x_1, x_2) \leq t + t' + 2$. By inspecting procedure FIX it is not hard to verify that this is a valid choice for a causality graph in the sense that no flaw f can cause flaws outside $\Gamma(f)$. This is because, in order to determine whether a flaw f_x is present in a state σ, we only need information about σ in $G \cap S_{<t'}(x)$, and procedure FIX locally modifies the state within a radius at most t of the input subgraph H.

The algorithmic proof of Theorem 5.1, which as we explained earlier is the key ingredient in making Kahn's result constructive, follows almost immediately by combining Theorem 2.4 with Lemma 5.4 below, whose proof can be found in Section 5.4.

Lemma 5.4. *Let $f \in \{f_e, f_v\}$ for an edge e and a vertex v. There exists Δ_0 such that if $\Delta \geq \Delta_0$ then*

$$\gamma_f \leq \frac{1}{\Delta^{3(t+t'+2)}} \ ,$$

where the charges are computed with respect to measure μ and the algorithm that samples from the ideal distributions.

Constructive Proof of Theorem 5.1. Recall from (8) that, setting $x_f = \frac{1}{1+\max_{f\in F}|\Gamma(f)|}$ for each flaw f, condition (3) with $\epsilon = \zeta/2$ is implied by

$$\max_{f\in F} \gamma_f \cdot \big(1 + \max_{f\in F} |\Gamma(f)|\big) \cdot \mathrm{e} \leq 1 - \zeta/2 \ . \tag{24}$$

Clearly, for each flaw f, $|\Gamma(f)| = O(\Delta^{2(t+t'+2)})$ so, by Lemma 5.4, condition (24) is satisfied for all sufficiently large Δ. Thus, Theorem 2.4 implies that, for every multigraph with large enough degree Δ_0, the algorithm for each iteration terminates after an expected number

$$O\left((m+n)\log_2\left(\frac{1}{1-1/\Delta^{2(t+t'+2)}}\right)\right) = O(n^2)$$

steps. □

Finally, the proof of Theorem 1.4 is concluded by combining the algorithm for Theorem 5.1 with the greedy algorithm of Theorem 5.2. It remains only for us to prove Lemma 5.4 stated above. This we do in the next subsection.

5.4 Proof of Lemma 5.4

Let $\Omega_1 = \prod_{i=1}^N \mathcal{M}(G_i)$ and $\Omega_2 = \Omega_3 = \prod_{i=1}^N \{0,1\}^{E_i}$ and note that each state in $\sigma \in \Omega$ can be represented as $\sigma = (\sigma_1, \sigma_2, \sigma_3) \in \Omega_1 \times \Omega_2 \times \Omega_3$. For notational convenience, sometimes we write $\Omega_1^i = \mathcal{M}(G_i)$ and $\Omega_2^i = \Omega_3^i = \{0,1\}^{E_i}$, for $i \in [N]$.

Let ν_1 be the distribution over Ω_1 induced by the product of distributions μ_i, $i \in [N]$. Let also ν_2, ν_3 be the distributions over Ω_2 and Ω_3 induced by the product of activation and equalizing coin flips of each color

$i \in [N]$, respectively. Recall that $\mu = \nu_1 \times \nu_2 \times \nu_3$ is a product distribution. Moreover, note that each ν_j, is the product of N distributions ν_j^i, one for each color $i \in [N]$. For example, notice that ν_1^i is another name for μ_i, while ν_2^i is the product measure over the edges of G_i induced by flipping a coin with probability α for each edge.

For $\sigma_1 = (M_1, M_2, \ldots, M_N) \in \Omega_1$, a subgraph H, and an integer $d > 0$, we define $Q_H(d, \sigma_1) = (M_1 - S_{<d}(H), \ldots, M_N - S_{<d}(H))$ and $Q_H^i(d, \sigma_1) = M_i - S_{<d}(H)$, similarly to the proof of Lemma 4.4. Moreover, for $\sigma_2 \in \Omega_2$ that represents the outcome of the activations, we let $A_H(d, \sigma_2)$ denote the restriction of σ_2 in $M_i - S_{<d}(H)$, for each color $i \in [N]$. In the same fashion, for $\sigma_3 \in \Omega_3$ that represents the outcome of the equalizing coin flips, we let $C_H(d, \sigma_3)$ denote the restriction of σ_3 in $M_i - S_{<d}(H)$ for each color $i \in [N]$. For $\sigma_2 \in \Omega_2, \sigma_3 \in \Omega_3$, we also define $A_H^i(d, \sigma_2)$ and $C_H^i(d, \sigma_3)$, $i \in [N]$, similarly to $Q_H^i(d, \sigma_1)$. Finally, for $\sigma = (\sigma_1, \sigma_2, \sigma_3) \in \Omega$, define $R_H(d, \sigma) = (Q_H(d, \sigma_1), A_H(d, \sigma_2), C_H(d, \sigma_3))$.

Our goal will be to show that, for every $x \in V \cup E$,

$$\gamma_{f_x} = \max_{\tau \in \Omega} \mu(\sigma \in f_x \mid R_x(t, \sigma) = R_x(t, \tau)) \ , \tag{25}$$

where σ is a random state distributed according to μ. This is because combining (25) with Lemma 5.3 concludes the proof.

We only prove (25) for f_e-flaws, since the proof for f_v flaws is very similar (and we have actually seen a big part of it in the proof of Lemma 4.4). Observe that whether flaw f_e is present at a state σ is determined by $\bigcup_{i=1}^{N} (G_i \cap S_{<t'}(e))$ and the entries of the activation and equalizing flip vectors of each color $i \in [N]$ that correspond to edges in $G_i \cap S_{<t'}(e)$. With that in mind, for each color i let $M_i(t', e) = M_i \cap E(G_i \cap S_{<t'}(e))$ and $a_i(t', e), h_i(t', e)$ denote the (random) vectors constraining the entries of the activation and equalizing coin flip vectors for color i that correspond to the edges of $G_i \cap S_{<t'}(e)$. Let also $\mathcal{D}_i(t', e)$ denote the domain of possible values of $(M_i(t', e), a_i(t', e), h_i(t', e))$.

The fact that we can determine whether f_e is present in a state by examining local information around e implies that there exists a set $X_e = X_e(t')$ of vectors of size N such that the i-th entry of a vector $x \in X_e$ is an element of $\mathcal{D}_i(t', e)$, and so that

$$f_e = \bigcup_{x \in X_e} \bigcap_{i \in [N]} \left(\left(M_i(t', e), a_i(t', e), h_i(t', e) \right) = x_i \right) \ . \tag{26}$$

For a state $\sigma \in \Omega$, let x_e^σ be the N-dimensional vector whose i-th entry is $(M_i(t', e), a_i(t', e), h_i(t', e))$. According to (26), for $\tau \in \Omega$ we have

$$\mu(\sigma \in f_e \mid R_e(t, \sigma) = R_e(t, \tau)) = \sum_{x \in X_e} \prod_{i=1}^{N} \mu(x_{e,i}^\sigma = x_i \mid R_e(t, \sigma) = R_e(t, \tau)) \ , \tag{27}$$

since the random choices of matching, activation, and equalizing coin flips for each color are independent. For an N-dimensional vector x whose i-th entry is an element of $\mathcal{D}_i(t', e)$, we write $x_i(j)$ to denote the j-th element of triple x_i. Thus, recalling the definition of the distributions ν_j^i, we have

$$\mu(x_{e,i}^\sigma = x_i \mid R_e(t, \sigma) = R_e(t, \tau)) = \prod_{j=1}^{3} \nu_j^i(x_{e,i}^\sigma(j) = x_i(j) \mid R_e(t, \sigma) = R_e(t, \tau)) \ , \tag{28}$$

because, for a fixed color, the random choices of matching, activation and equalizing coin flips are independent.

Recall now that for a subgraph H, multigraph $G_{<d+1}(H)$ is induced by $S_{<d+1}(H)$ and $\mathcal{M}^i_{d+1}(H,\sigma)$ is the set of matchings of $G_{<d+1}(H)$ that are compatible with $Q^i_H(d,\sigma_1)$. Hence,

$$\begin{aligned}\nu^i_1(x^\sigma_{e,i}(1)=x_i(1)\mid R_e(t,\sigma)=R_e(t,\tau)) &= \nu^i_1(x^\sigma_{e,i}(1)=x_i(1)\mid Q^i_e(t,\sigma_1)=Q^i_e(t,\tau_1))\\ &= \frac{\nu^i_1(x^\sigma_{e,i}(1)=x_i(1)\cap Q^i_e(t,\sigma_1)=Q^i_e(t,\tau_1))}{\nu^i_1(Q^i_e(t,\sigma_1)=Q^i_e(t,\tau_1))}\\ &= \frac{\sum_{M\in\mathcal{M}^i_{t+1}(e,\tau_1),M\cap S_{<t'}(e)=x_i(1)}\lambda_i(M)}{\sum_{M\in\mathcal{M}^i_{t+1}(e,\tau_1)}\lambda_i(M)}. \qquad (29)\end{aligned}$$

Moreover, we clearly have

$$\nu^i_2(x^\sigma_{e,i}(2)=x_i(2)\mid R_e(t,\sigma)=R_e(t,\tau)) = \nu^i_2(a_i(t',e)=x_i(2))\ , \qquad (30)$$

$$\nu^i_3(x^\sigma_{e,i}(3)=x_i(3)\mid R_e(t,\sigma)=R_e(t,\tau)) = \nu^i_3(h_i(t',e)=x_i(3)) \qquad (31)$$

We will use (27)-(31) to show that, for σ distributed according to μ and any state $\tau\in\Omega$,

$$\sum_{\omega\in f_e}\frac{\mu(\omega)}{\mu(\tau)}\rho_{f_e}(\omega,\tau)=\mu(\sigma\in f_e\mid R_e(t,\sigma)=R_e(t,\tau))\ . \qquad (32)$$

According to the definition of γ_{f_e}, maximizing (32) over $\tau\in\Omega$ yields (25).

To compute the sum in (32) we need to determine the set of states $\mathrm{In}_e(\tau)=\{\omega:\rho_{f_e}(\omega,\tau)>0\}$. We claim that for each $\omega\in\mathrm{In}_e(\tau)$ we have that $R_e(t,\omega)=R_e(t,\tau)$.

To see this, let

$$\begin{aligned}\omega &= (\omega_1,\omega_2,\omega_3)=\big((\omega^1_1,\ldots,\omega^N_1),(\omega^1_2,\ldots,\omega^N_2),(\omega^1_3,\ldots\omega^N_3)\big)\ ,\\ \tau &= (\tau_1,\tau_2,\tau_3)=\big((\tau^1_1,\ldots,\tau^N_1),(\tau^1_2,\ldots,\tau^N_2),(\tau^1_3,\ldots,\tau^N_3)\big)\ ,\end{aligned}$$

where $\omega_j,\tau_j\in\Omega_j$ and $\omega^i_j,\tau^i_j\in\Omega^i_j$. Notice that the probability distribution $\rho_{f_e}(\omega,\cdot)$ can be seen as the product of $3N$ distributions. Namely, for each $i\in[N]$ we have a probability distribution $\rho^{i,1}_{f_e}(\omega^i_1,\cdot)$ corresponding to Line 3 of FIX and color i, and similarly, for ω^i_2,ω^i_3 we have probability distributions $\rho^{i,2}_{f_e}(\omega^i_2,\cdot),\rho^{i,3}_{f_e}(\omega^i_3,\cdot)$, corresponding to Lines 5, 6 of FIX and color i, respectively. Recalling procedure RESAMPLE, we see that the support of $\rho^{i,1}_{f_e}(\omega^i_1,\cdot)$ is $\mathcal{M}^i_{t+1}(e,\omega_1)$ and, thus, it must be the case that $Q^i_e(t,\omega_1)=Q^i_e(t,\tau_1)$ for every $i\in[N]$ and state $\omega\in\mathrm{In}_e(\tau)$. Similarly, by inspecting procedure FIX one can verify that $A^i_e(t,\omega_2)=A^i_e(t,\tau_2)$ and that $C^i_e(t,\omega_3)=C^i_e(t,\tau_3)$ for each $i\in[N]$. Hence, $R_e(t,\omega)=R_e(t,\tau)$, as claimed.

For each $\omega\in f_e$,

$$\frac{\mu(\omega)}{\mu(\tau)}\rho_{f_e}(\omega,\tau)=\prod_{i=1}^N\prod_{j=1}^3\frac{\nu^i_j(\omega^i_j)}{\nu^i_j(\tau^i_j)}\rho^{i,j}_{f_e}(\omega^i_j,\tau)=:\prod_{i=1}^N\prod_{j=1}^3 r_{i,j}(\omega)\ . \qquad (33)$$

We will now give an alternative expression for each $r_{i,j}(\omega)$ in order to relate (33) to (32). We start with $r_{i,1}(\omega)$. The fact that $Q^i_e(t,\omega_1)=Q^i_e(t,\tau_1)$ for each $\omega\in\mathrm{In}_e(\tau)$ implies that

$$\frac{\nu^i_1(\omega^i_1)}{\nu^i_1(\tau^i_1)}=\frac{\lambda_i(\omega^i_1\cap E(G_{<t+1}(e)))}{\lambda_i(\tau^i_1\cap E(G_{<t+1}(e))}\ . \qquad (34)$$

Furthermore, since we have assumed that the hard-core distribution in Lines 8, 9 of RESAMPLE is induced by the ideal vector of activities λ_i, we have

$$\rho_{f_e}(\omega_1^i, \tau_1^i) = \frac{\lambda_i(\tau_1^i \cap E(G_{<t+1}(e))}{\sum_{M \in \mathcal{M}_{t+1}^i(e,\omega_1)} \lambda_i(M)} \ . \tag{35}$$

Combining (34) with (35) and the fact that $Q_e^i(t, \omega_1) = Q_e^i(t, \tau_1)$ we obtain

$$r_{i,1}(\omega) = \frac{\lambda_i(\omega_1^i \cap E(G_{<t+1}(e))}{\sum_{M \in \mathcal{M}_{t+1}^i(e,\tau_1)} \lambda_i(M)} \ . \tag{36}$$

Recall now the definitions of $a_i(t', e)$ and $h_i(t', e)$. The fact that $A_e^i(t, \omega_2) = A_e^i(t, \tau_2)$ for each $\omega \in \mathrm{In}_e(\tau)$ implies that

$$\frac{\nu_2^i(\omega_2^i)}{\nu_2^i(\tau_2^i)} = \frac{\nu_2^i(a_i(t', e) = x_{e,i}^{\omega}(2))}{\nu_2^i(a_i(t', e) = x_{e,i}^{\tau}(2))} \ . \tag{37}$$

Further, since in Line 5 of FIX we simply flip a coin independently with success probability α for each edge of $G_i \cap S_{<t+1}(e)$, we have

$$\rho_{f_e}(\omega_2^i, \tau_2^i) = \frac{\nu_2^i(a_i(t', e) = x_{e,i}^{\tau}(2))}{\sum_a \nu_2^i(a_i(t', e) = a)} \ , \tag{38}$$

where the sum in the denominator ranges over all the possible values for $a_i(t', e)$. Thus, combining (37) with (38) we get

$$r_{i,2}(\omega) = \frac{\nu_2^i(a_i(t', e) = x_{e,i}^{\omega}(2))}{\sum_a \nu_2^i(a_i(t', e) = a)} \ . \tag{39}$$

Finally, an identical argument shows that

$$r_{i,3}(\omega) = \frac{\nu_3^i(h_i(t', e) = x_{e,i}^{\omega}(2))}{\sum_h \nu_3^i(h_i(t', e) = h)} \ . \tag{40}$$

For $x \in X_e$, let $\Omega_{e,x} = \{\omega : x_e^{\omega} = x\}$. For σ distributed according to μ, the left-hand side of (32) can be written as

$$\begin{aligned}
\sum_{x \in X_e} \sum_{\omega \in \Omega_e} \frac{\mu(\omega)}{\mu(\tau)} \rho_{f_e}(\omega, \tau) &= \sum_{x \in X_e} \sum_{\omega \in \Omega_{e,x}} \prod_{i=1}^{N} \prod_{j=1}^{3} r_{i,j}(\omega) \\
&= \sum_{x \in X_e} \prod_{i=1}^{N} \prod_{j=1}^{3} \sum_{\substack{\omega \in \Omega_{e,x} \\ x_{e,i}^{\omega} = x_i(j)}} r_{i,j}(\omega) \qquad (41) \\
&= \sum_{x \in X_e} \prod_{i=1}^{N} \prod_{j=1}^{3} \nu_j^i(x_{e,i}^{\sigma}(j) = x_i(j) \mid R_e(t, \sigma) = R_e(t, \tau)) \qquad (42) \\
&= \mu(\sigma \in f_e \mid R_e(t, \sigma) = R_e(t, \tau)) \ ,
\end{aligned}$$

concluding the proof of (32). Note that (41) follows from the fact that Ω is a product space, and (42) follows by (29) and (36) for $j = 1$, (30) and (39) for $j = 2$, and (31) and (40) for $j = 3$.

6 Acknowledgements

We thank Dimitris Achlioptas for many helpful discussions.

References

[1] Dimitris Achlioptas and Fotis Iliopoulos. Random walks that find perfect objects and the Lovász local lemma. *J. ACM*, 63(3):22:1–22:29, July 2016.

[2] Dimitris Achlioptas, Fotis Iliopoulos, and Vladimir Kolmogorov. A local lemma for focused stochastic algorithms. *arXiv preprint arXiv:1809.01537*, 2018.

[3] Dimitris Achlioptas, Fotis Iliopoulos, and Alistair Sinclair. A new perspective on stochastic local search and the Lovász local lemma. *CoRR*, abs/1805.02026, 2018.

[4] Dimitris Achlioptas, Fotis Iliopoulos, and Nikos Vlassis. Stochastic control via entropy compression. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 83:1–83:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[5] Michael Albert, Alan Frieze, and Bruce Reed. Multicoloured Hamilton cycles. *the electronic journal of combinatorics*, 2(1):R10, 1995.

[6] Guantao Chen, Xingxing Yu, and Wenan Zang. Approximating the chromatic index of multigraphs. *Journal of Combinatorial Optimization*, 21(2):219–246, 2011.

[7] Andrzej Dudek, Alan Frieze, and Andrzej Ruciński. Rainbow Hamilton cycles in uniform hypergraphs. *the electronic journal of combinatorics*, 19(1):46, 2012.

[8] Jack Edmonds. Maximum matching and a polyhedron with 0, 1-vertices. *Journal of Research of the National Bureau of Standards B*, 69(125-130):55–56, 1965.

[9] Paul Erdős and László Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. In *Infinite and finite sets (Colloq., Keszthely, 1973; dedicated to P. Erdős on his 60th birthday), Vol. II*, pages 609–627. Colloq. Math. Soc. János Bolyai, Vol. 10. North-Holland, Amsterdam, 1975.

[10] Paul Erdös and Joel Spencer. Lopsided Lovász local lemma and latin transversals. *Discrete Applied Mathematics*, 30(2-3):151–154, 1991.

[11] Uriel Feige, Eran Ofek, and Udi Wieder. Approximating maximum edge coloring in multigraphs. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 108–121. Springer, 2002.

[12] Bernhard Haeupler, Barna Saha, and Aravind Srinivasan. New constructive aspects of the Lovász local lemma. *J. ACM*, 58(6):Art. 28, 28, 2011.

[13] Nicholas J. A. Harvey and Jan Vondrák. An algorithmic proof of the Lovász local lemma via resampling oracles. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1327–1346. IEEE Computer Society, 2015.

[14] Fotis Iliopoulos. Commutative algorithms approximate the LLL-distribution. In Eric Blais, Klaus Jansen, José D. P. Rolim, and David Steurer, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2018, August 20-22, 2018 - Princeton, NJ, USA*, volume 116 of *LIPIcs*, pages 44:1–44:20. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.

[15] Mark Jerrum and Alistair Sinclair. Approximating the permanent. *SIAM journal on computing*, 18(6):1149–1178, 1989.

[16] A. Johansson. Asympotic choice number for triangle free graphs., 1996.

[17] Jeff Kahn. Asymptotically good list-colorings. *Journal of Combinatorial Theory, Series A*, 73(1):1–59, 1996.

[18] Jeff Kahn. Asymptotics of the chromatic index for multigraphs. *journal of combinatorial theory, Series B*, 68(2):233–254, 1996.

[19] Jeff Kahn. Asymptotics of the list-chromatic index for multigraphs. *Random Structures & Algorithms*, 17(2):117–156, 2000.

[20] Jeff Kahn and P Mark Kayll. On the stochastic independence properties of hard-core distributions. *Combinatorica*, 17(3):369–391, 1997.

[21] Jeong Han Kim. On Brooks' theorem for sparse graphs. *Combinatorics, Probability and Computing*, 4(2):97–132, 1995.

[22] Vladimir Kolmogorov. Commutativity in the algorithmic Lovász local lemma. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 780–787. IEEE Computer Society, 2016.

[23] Carl W Lee. Some recent results on convex polytopes. *Contemporary Math*, 114:3–19, 1990.

[24] Michael Molloy. The list chromatic number of graphs with small clique number. *arXiv preprint arXiv:1701.09133*, 2017.

[25] Michael Molloy and Bruce Reed. A bound on the total chromatic number. *Combinatorica*, 18(2):241–280, 1998.

[26] Michael Molloy and Bruce Reed. *Graph colouring and the probabilistic method*, volume 23 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 2002.

[27] Robin A. Moser. A constructive proof of the Lovász local lemma. In *STOC'09—Proceedings of the 2009 ACM International Symposium on Theory of Computing*, pages 343–350. ACM, New York, 2009.

[28] Robin A. Moser and Gábor Tardos. A constructive proof of the general Lovász local lemma. *J. ACM*, 57(2):Art. 11, 15, 2010.

[29] Takao Nishizeki and Kenichi Kashiwagi. On the 1.1 edge-coloring of multigraphs. *SIAM Journal on Discrete Mathematics*, 3(3):391–410, 1990.

[30] Manfred W Padberg and M Ram Rao. Odd minimum cut-sets and b-matchings. *Mathematics of Operations Research*, 7(1):67–80, 1982.

[31] Michael Plantholt. A sublinear bound on the chromatic index of multigraphs. *Discrete Mathematics*, 202(1-3):201–213, 1999.

[32] Yuri Rabinovich, Alistair Sinclair, and Avi Wigderson. Quadratic dynamical systems. In *Foundations of Computer Science, 1992. Proceedings., 33rd Annual Symposium on*, pages 304–313. IEEE, 1992.

[33] Peter Sanders and David Steurer. An asymptotic approximation scheme for multigraph edge coloring. *ACM Transactions on Algorithms (TALG)*, 4(2):21, 2008.

[34] Diego Scheide. *On the 15/14 edge-colouring of multigraphs*. Institut for Matematik og Datalogi, Syddansk Universitet, 2007.

[35] Diego Scheide. Graph edge colouring: Tashkinov trees and Goldberg's conjecture. *Journal of Combinatorial Theory, Series B*, 100(1):68–96, 2010.

[36] Mohit Singh and Nisheeth K Vishnoi. Entropy, optimization and counting. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 50–59. ACM, 2014.

[37] Vadim G Vizing. On an estimate of the chromatic class of a p-graph. *Discret Analiz*, 3:25–30, 1964.

A Proof of Lemma 4.6

We will need the following standard concentration bound (see, e.g., Chapter 10, Section 10.1 of [26]).

Lemma A.1. *Let X be a random variable determined by n independent trials $T_1, \ldots, T_n$, and such that changing the outcome of any one trial can affect X by at most c. Then*

$$\Pr[|X - \mathbb{E}[X]| > \lambda] \leq 2\mathrm{e}^{-\frac{\lambda^2}{2c^2 n}} \ .$$

Proof of Part (a) of Lemma 4.6. Recall that $t = 8(K+1)^2\delta^{-1} + 2$ and that $\delta = \frac{\epsilon}{4}$. Consider a random state σ distributed according to μ and a fixed state $\tau \in \Omega$, and notice that applying Theorem 2.9 with the parameter ϵ instantiated to δ and our choice of t imply that

$$\mu(e \in M_i \mid Q_v^i(t,\sigma) = Q_v^i(t,\tau)) \geq (1-\delta)\frac{1-\delta}{\chi_e^*(G)} \geq \frac{1-\frac{\epsilon}{2}}{\chi_e^*(G)} \ ,$$

for any vertex v, any edge e adjacent to v and any $i \in [N]$. This implies

$$\mathbb{E}[d_{G_\sigma}(v) \mid Q_v^i(t,\sigma) = Q_v^i(t,\tau)] \leq \Delta\left(1 - \frac{1-\frac{\epsilon}{2}}{\chi_e^*(G)}\right)^N \leq \chi_e^*(G)\left(1 - \frac{1-\frac{\epsilon}{2}}{\chi_e^*(G)}\right)^N \ . \tag{43}$$

Now, since $N = o(\chi_e^*(G))$, we have

$$\mathbb{E}[d_{G_\sigma}(v) \mid Q_v^i(t,\sigma) = Q_v^i(t,\tau)] \leq \chi_e^*(G)\left(1 - (1+o(1))\frac{(1-\frac{\epsilon}{2})N}{\chi_e^*(G)}\right) \leq \chi_e^*(G) - \left(1 - \frac{9\epsilon}{17}\right)N \ . \tag{44}$$

Further, since $c^* = \chi_e^*(G) - (1+\epsilon)^{-1}N$ and $\epsilon \leq \frac{1}{10}$, (44) yields

$$\mathbb{E}[d_{G_\sigma}(v) \mid Q_v^i(t,\sigma) = Q_v^i(t,\tau)] \leq c^* - \left(1 - \frac{9\epsilon}{17} - (1+\epsilon)^{-1}\right)N \leq c^* - \frac{\epsilon}{3}N \ . \tag{45}$$

As the choices of the M_i are independent and each affects the degree of v in G' by at most 1, we can apply Lemma A.1 with $\lambda = (\frac{\epsilon}{3} - \frac{\epsilon}{4})N = \frac{\epsilon}{12}N$ to prove part (a). In particular, recalling that $N = \lfloor \chi_e^*(G)^{\frac{3}{4}} \rfloor \sim \Delta^{3/4}$ we have that

$$\mu\left(d_{G_\sigma(v)} > c^* - \frac{\epsilon}{4}N \;\middle|\; Q_v^i(t,\sigma) = Q_v^i(t,\tau)\right) \leq 2\mathrm{e}^{-\frac{\lambda^2}{2N}} \leq \frac{1}{\Delta^{C+\Delta^{\frac{1}{3}}}} \ ,$$

for any constant C for sufficiently large Δ. □

Proof of Part (b) . The proof of part (b) is similar. Consider again a random state σ distributed according to μ and fix a state $\tau \in \Omega$. Theorem 2.9 implies that for each $i \in [N]$, the probability that an edge e with both endpoints in H is in M_i, conditional on $Q_H^i(t,\sigma) = Q_H^i(t,\tau)$, is at least $(1-\delta)\frac{1-\delta}{\chi_e^*(G)} \geq \frac{1-\frac{\epsilon}{2}}{\chi_e^*(G)}$. Moreover, Edmonds' characterization of the matching polytope (which we have already seen in the the proof of Lemma 4.2) implies that the number of edges in G with both endpoints in H is at most $\chi_e^*(G)\lfloor \frac{V(H)-1}{2} \rfloor$. Similar calculations to the ones in part (a) reveal that

$$\mathbb{E}[|E_\sigma(H)| \mid Q_H^i(t,\sigma) = Q_H^i(t,\tau)] \leq \left(\frac{V(H)-1}{2}\right)(c^* - \frac{\epsilon}{3}N) \ ,$$

where $E_\sigma(H)$ is the set of edges of G_σ induced by H. Since the choices of matchings M_i are independent and each affects $|E_\sigma(H)|$ by at most $\frac{|V(H)|-1}{2}$, we can again apply Lemma A.1 to prove part (b). □