

Constraint Satisfaction, Packet Routing, and the Lovász Local Lemma

David G. Harris^{*}

Aravind Srinivasan[†]

ABSTRACT

Constraint-satisfaction problems (CSPs) form a basic family of *NP*-hard optimization problems that includes satisfiability. Motivated by the sufficient condition for the satisfiability of SAT formulae that is offered by the Lovász Local Lemma, we seek such sufficient conditions for arbitrary CSPs. To this end, we identify a *variable-covering radius*-type parameter for the infeasible configurations of a given CSP, and also develop an extension of the Lovász Local Lemma in which many of the events to be avoided have probabilities *arbitrarily close to one*; these lead to a general sufficient condition for the satisfiability of arbitrary CSPs. One primary application is to packet-routing in the classical Leighton-Maggs-Rao setting, where we introduce several additional ideas in order to prove the existence of near-optimal schedules; further applications in combinatorial optimization are also shown.

1. INTRODUCTION

Constraint-satisfaction problems (CSPs) abstract a large variety of problems in combinatorial optimization. In their basic form, there are n variables where the i th variable is required to take values from some finite ordered set A_i . A CSP then is just a collection of *infeasible* configurations $\mathcal{F} \subseteq A_1 \times A_2 \times \dots \times A_n$; the decision problem is whether $\overline{\mathcal{F}}$ is nonempty, and the optimization problem is to find some element of $\overline{\mathcal{F}}$ (if this set is nonempty). The k -CNF-SAT problem is a well-known instance of CSPs, in which each element of \mathcal{F} is a conjunction of k literals formed by the un-

derlying Boolean variables or their complements (with the other $n - k$ being don't-cares). In addition to algorithms and complexity, CSPs have also spurred research in algebra, logic, model theory, probabilistic methods, phase transitions etc.; we just mention the algebraic dichotomy conjecture [7] as one of the central contributions to, and conjectures of, this field. Motivated by the well-known sufficient condition provided by the Lovász Local Lemma (LLL) for the satisfiability of k -CNF-SAT formulae [10], this work develops a general sufficient condition for the satisfiability of arbitrary CSPs – formulating a generalization of the LLL in the process.

We start by recalling how a key parameter works well with the LLL, for k -CNF-SAT instances (where each clause has exactly k literals, say). Let e denote the base of the natural logarithm. The basic (“symmetric”) version of the LLL states that if we have a collection of m “bad” events B_1, B_2, \dots, B_m such that $\Pr[B_i] \leq p$ with each B_i “depending” on at most d other B_j , then $e \cdot p \cdot (d + 1) \leq 1$ is a sufficient condition for $\Pr[\overline{B_1} \cap \overline{B_2} \cap \dots \cap \overline{B_m}] > 0$. The fact that $m \gg d$ is allowed, leads to a number of applications of the LLL [2]. The more general (“asymmetric”) version of the LLL allows heterogeneous values for the probability and amount of independence of each B_i . For nearly all applications of the LLL, there are now polynomial-time algorithms which can find configurations that avoid all bad events [22, 14, 18, 24]. An easy application of the symmetric version of the LLL shows that if the maximum number d of clauses in which any literal (X_i or \overline{X}_i) appears is at most $2^k/(ek) - 1/k$, then the formula is satisfiable (see [30, 12] for further improvements); furthermore, a satisfying assignment can be found in polynomial time [22]. We ask: *what is an analog of d for an arbitrary CSP \mathcal{P} , a suitable upper bound for which implies the satisfiability of \mathcal{P} ?*

To understand this, let us observe that any CSP can be written as a simple 0 – 1 integer linear program as follows. Denote the set $\{1, 2, \dots, t\}$ by $[t]$. Let $x_{i,j}$ be the indicator variable for the i th variable taking on the value $j \in A_i$; thus, $\forall i \in [n], \sum_{j \in A_i} x_{i,j} = 1$. Next, each forbidden configuration $\langle i_1, j_1 \rangle, \dots, \langle i_k, j_k \rangle$ can be avoided by writing the constraint “ $x_{i_1, j_1} + \dots + x_{i_k, j_k} \leq k - 1$ ”. Note that this linear packing constraint is (of course) monotone decreasing in the variables x . Allowing more general, nonlinear but monotone decreasing constraints could make the description succinct and/or sparse; we will see this in our packet-routing application.

With this in mind, we start with the following formulation of arbitrary CSPs:

^{*}Department of Applied Mathematics, University of Maryland, College Park, MD 20742. Research supported in part by NSF Award CNS-1010789. Email: davidgharris29@hotmail.com

[†]Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742. Research supported in part by NSF Awards CCR-0208005, ITR CNS-0426683, CNS-0626636, and CNS-1010789. Email: srin@cs.umd.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

STOC'13, June 16–18, 2013, Palo Alto, California, USA.
Copyright 2013 ACM 978-1-4503-2029-0/13/06 ...\$10.00.

Definition 1.1. (CSPs as Integer Programs with Decreasing Boolean Constraints.) Consider the following type of integer program, where $x_{i,j}$ here is the indicator variable for the i th variable of the CSP taking on the value $j \in A_i$, and $\mathcal{V} = \{x_{i,j} : i \in [n], j \in A_i\}$ denotes the set of all these indicator variables:

(Assignment Constraints) $\forall i \in [n], \sum_{j \in A_i} x_{i,j} = 1$.

(Decreasing Boolean Constraints) For all $k \in [\ell]$, a given Boolean function B_k , which is an **increasing** function of each of the variables in some given subset $S_k \subseteq \mathcal{V}$, should be **false**. (As our goal is to avoid B_k , we sometimes refer to these as “bad events”.)

(Integrality) $\forall (i, j), x_{i,j} \in \{0, 1\}$.

A technical remark. Our integer programs will have the above three types of constraints; a remark is in order. Suppose $|A_i| = 2$, say, and that we have variables $x_{i,1}$ and $x_{i,2}$ which are constrained to sum to 1, by the assignment constraint. Now if B_k is an increasing function of $x_{i,1}$, it is a decreasing function of $x_{i,2}$ on the set of feasible solutions. To avoid any confusion about this, the reader is asked to note that for each k , there is *some* $S_k \subseteq \mathcal{V}$ such that B_k is an increasing function of each variable in S_k (e.g., we will have $x_{i,1} \in S_k$ here).

Our new “assignment LLL” is given as Theorem 2.2; its proof departs from the standard inductive proof of the LLL to develop a different induction that combines the “Rödl Nibble” [?] and correlation inequalities. Intriguingly, our approach will set each $x_{i,j}$ to 1 with infinitesimally-small probability: note that each assignment constraint is now very likely to be violated. Nevertheless, this yields a pathway to proving the theorem. We apply the assignment LLL to improve bounds of [1, 3, 8, 15, 16, 19, 20, 25, 31]. However, our approach here does not appear to lead to an algorithmic counterpart of our assignment LLL. In many of our examples, this extended LLL framework can be applied as an almost automatic replacement for the original LLL. Hence, many of these applications will be very direct. In addition, we will examine the case of packet-routing in the classical setting of [20] in detail: we show here that there are many other improvements possible, using the standard LLL and further using our assignment LLL.

The following definition will be crucial:

Definition 1.2. $Y_{i,j} = \{k \mid x_{i,j} \in S_k\}$ indexes those B_k that increase explicitly as a function of $x_{i,j}$.

Informal discussion of the assignment LLL. The key analog of the parameter d from k -SAT, is $d = \max_{i,j} |Y_{i,j}|$: informally, we can view d as the “variable-covering radius” of the CSP – the maximum number of events that are influenced by any given variable. To understand the assignment LLL informally (albeit somewhat imprecisely), it roughly says that a sufficient condition for the CSP to be satisfiable, is the existence of a non-negative vector z (which is a relaxation of the vector x required by the three CSP constraints above) and a corresponding random vector Z – as well as some small $\delta > 0$ – with the following two properties:

1. the assignment constraints are slightly “over-satisfied”, i.e., for all i , $\sum_{j \in A_i} z_{ij} \geq e^\delta$, and

2. if $\vec{Z} = (Z_{i,j})$ is the vector of mutually independent Bernoulli random variables with $\mathbf{P}(Z_{i,j} = 1) = z_{i,j}$, then for all (i, j) and all $k \in Y_{i,j}$, $\mathbf{P}(B_k(\vec{Z}) \mid Z_{i,j} = 1) < \delta/d$.

(We stress that this is an approximate restatement, and is not to be taken literally.) The fact that the probability bound in the second item depends on d alone, is one of the key benefits brought by the assignment LLL. In contrast, a typical method of applying the LLL to this type of problem is to select *exactly one* value for each variable in the CSP – very often independently, with the i^{th} CSP variable chosen from some distribution on A_i (the distribution is often uniform or given by an LP or other convex program) – and to then treat the B_k as the bad events. The advantage of this method is that the assignment constraints are automatically satisfied. Unfortunately: (i) this introduces significant dependencies (e.g., between B_k and $B_{k'}$ that respectively depend on $x_{i,j}$ and $x_{i,j'}$), and more importantly, (ii) some B_k may depend on a large number of variables, even if d is small – thus making the dependencies among the B_k “large.” This is the difference between row-sparsity and column-sparsity of the dependency matrix.

We next discuss a variety of applications, and compare them to the best previous results (many proved using the standard LLL).

(a) Packet routing: the Leighton-Maggs-Rao framework. A fundamental packet-routing problem is as follows. Suppose we are given an undirected graph G with N packets, in which we need to route each packet i from vertex s_i to vertex t_i along a *given simple path* P_i . The constraints are that each edge can only carry one packet at a time, and each edge traversal takes unit time for a packet; nodes are allowed to queue packets. The goal is to conduct routings along the paths P_i , in order to minimize the *makespan* T (the time by which all packets have reached their destinations).

Two natural lower-bounds on T are the *congestion* C (the maximum number of the P_i that contain any given edge of the graph) and the *dilation* D (the length of the longest P_i); thus, $(C + D)/2$ is always a lower-bound, and there exist families of instances with $T \geq (1 + \Omega(1)) \cdot (C + D)$ [28]. A seminal result proven in [20] (via iterated application of the LLL, which has become a key tool in its own right [21]) is that in fact $T \leq O(C + D)$ for all input instances, using constant-sized queues at the edges; both this result and its approach, have been used in much work in networks and combinatorial optimization. This argument was refined and simplified in [29, 25], leading to a (constructive) bound of $23.4(C + D)$ [25]. By introducing several new ideas in scheduling packets along time intervals, we improve this to $7.26(C + D)$ (non-constructively) and $8.84(C + D)$ (constructively), thus approaching the simple lower bound for this fundamental problem.

The packet-routing application above involves increasing Boolean functions B_k that are non-linear. The rest of our applications have linear “packing” constraints; i.e., each bad event B_k is of the form “ $\sum_{(i,j)} a_k(i,j)x_{i,j} > b_k$ ”, where the given values $\{a_k(i,j)\}$ and b_k are non-negative.

(b) Integrality gap for column-sparse matrices. Consider the special case of our family of CSPs where each B_k is a linear packing constraint: of the form “ $\sum_{(i,j)} a_k(i,j)x_{i,j} > b_k$ ”, where $\{a_k(i,j)\} \in \{0, 1\}$ and $b_k \geq 0$. When does such an integer linear program have a feasible solution? In par-

ticular, suppose there exists a solution x' to the natural fractional relaxation¹ of our CSPs in which $\sum_{(i,j)} a_k(i,j)x'_{i,j} \leq b'_k$ holds for each $k \in [\ell]$; how small should the b'_k be for our CSP to be satisfiable? We work in the classical column-sparse setting [4, 17] where for each (i,j) , the number of k for which $a_k(i,j)$ is nonzero, is at most some parameter d .

Consider, for notational simplicity, the case where $b'_k \geq \Omega(\log d)$ for all k . There is a classical linear-algebraic approach to such problems [4, 17]; in particular, the work of [17] shows that “ $b_k \geq b'_k + d$ for each k ” suffices. Complementing this, our assignment LLL shows that, the condition “ $b_k \geq b'_k(1 + d^{-C}) + C' \cdot \sqrt{b'_k} \log d$ for each k ” is a sufficient condition, where C is any positive constant, and where C' is a constant that only depends on C . The related theorems of [31, 19] give a weaker result – which, furthermore, holds only when all the b_k are within $O(1)$ of each other. However, the results of [17, 19] are constructive, while ours is not. (Note again that these earlier results hold only when each B_k is linear, while we allow arbitrary increasing functions.)

(c) Independent transversals. Let n and Δ denote as usual the number of vertices and maximum degree respectively, of an undirected graph $G = (V, E)$. It is well-known that G has an independent set of size at least $n/(\Delta + 1)$. What if we want independent sets with much more structure? In particular, say that we have a partition of the vertices of G into blocks of size b . We wish to select exactly one vertex from each block such that the selected vertices form an independent set in G . This is known as an *independent transversal*.

A well-known result of Alon using a simple LLL-based proof [1, 2] shows that if $b \geq 2e\Delta$, then G has an independent transversal. Thus, at the cost of the loss of a small constant ($2e \sim 5.44$), we get much more structure as compared to the basic “ $n/(\Delta + 1)$ ” lower-bound. Using the algorithmic LLL of [24], this can be improved to show that $b \geq 4\Delta$ is sufficient. The existentially optimal, but non-constructive, result of [15, 16] states that $b \geq 2\Delta$ suffices.

Our approach shows that $G \geq 4\Delta + 1$ suffices, moreover, we are able to obtain two improvements compared to previous results. First, if the given partition of V has the property that the *average* degree of the vertices in each block is at most d , then the method of [1] shows that it suffices to have $b \geq 2e(d + 1)$, which does not appear to carry over to the corresponding $b \geq 2\Delta$ bound of [15, 16]; we show that $b > 4d$ suffices. (Pegden’s method [24] can also be used to derive this bound.) Second, our method extends to the weighted case, where the approach of [15, 16] does not hold to our knowledge. Suppose now that each vertex i has a weight $w_i \geq 0$; it is not hard to observe that G has an independent set of total weight at least $(\sum_i w_i)/(\Delta + 1)$, and that there does not always exist an independent transversal of weight more than $(\sum_i w_i)/b$. Here, we show that if our partition satisfies $b > 4\Delta$, then there exists an independent transversal of weight at least $(\sum_i w_i) \frac{\sqrt{b-4\Delta} + \sqrt{b}}{\sqrt{b-4\Delta} + \sqrt{b(2b-1)}}$ (which approaches $(\sum_i w_i)/b$ for large b).

We apply our result for independent transversals to the problem of *chromatic capacity*. Rödl has shown that for any acyclic digraph D , there exists an undirected graph D^* such that all of the latter’s acyclic orientations contain D as an induced subgraph [26]. Cochand & Duchet introduced

¹Wherein the constraint “ $\forall(i,j), x_{i,j} \in \{0,1\}$ ” is relaxed to let the $x_{i,j}$ lie in $[0,1]$.

the notion of *chromatic capacity* to explicitly construct D^* , using graphs that have a large value for this parameter [8]. We improve the degree-based upper-bound for the chromatic number from [3, 8], as well as a related upper-bound on the “ k^{th} upper chromatic number” of the reals due to [3].

We present the Assignment LLL and related notions in Sections 2, 3 and 4, and follow up with applications in the subsequent sections.

2. THE ASSIGNMENT LLL

We begin by fixing our notation. For a vector v and scalar a , we let $v = \bar{a}$ denote that each entry of v is a . Define $\mathcal{A} = [n]$ to be the family of all blocks, and let $Z_{i,j}$ be the indicator for selecting the j th element from block i . We are given some Boolean functions B_1, B_2, \dots, B_ℓ , such that each B_k is determined by, and is an increasing function of, a subset S_k of the Z variables.

We refer to any ordered pair $\langle i, j \rangle$ where $j \in A_i$ as an *element*. So, for example, we might refer to Y_x where $x = \langle i, j \rangle$ as an element. We suppose there are N elements, and sometimes identify the set of elements with the set $[N]$. When discussing blocks i and possible values $j \in A_i$, we will often omit to mention that $j \in A_i$ (e.g., as in $\sum_j p_{i,j}$). This should always be understood as j being restricted to A_i , or equivalently that $p_{i,j} = 0$ for $j \notin A_i$.

Unlike in the usual LLL, we cannot simply define a probability distribution and compute the probability of the bad event occurring. We will only have partial control over this probability distribution, and the following definition will be important:

Definition 2.1. (*UNC*) Given a probability distribution Ω on the underlying indicator variables Z , we say that Ω satisfies upper negative correlation with respect to the vector $q \in [0, \infty)^N$ – or simply “*UNC*(q)” – if for all k and for all distinct elements x_1, \dots, x_k , we have

$$\mathbf{P}_\Omega(Z_{x_1} = \dots = Z_{x_k} = 1) \leq q_{x_1} \dots q_{x_k}.$$

For an event E and vector q , we define $\mathbf{P}_q^*(E)$ to be the minimum, over all Ω satisfying *UNC*(q), of $\mathbf{P}_\Omega(E)$. (As the number of variables is finite, this minimum is achieved. Also, while we could restrict $q \in [0, 1]^N$ without loss of generality, by allowing $q \in [0, \infty)^N$ we simplify the notation considerably.)

Essentially, when computing $\mathbf{P}_q^*(E)$, we are not allowing the random variables Z to be positively correlated. For some types of events, such as large-deviation events, this allows us to control the probability very strongly; for other events, such as a union of many events, this is no better than the union bound.

Our main theorem is:

Theorem 2.2 (Assignment LLL). *Suppose we are given a CSP for which there exists $\lambda \in [0, \infty]^N$ such that*

$$\forall i \in [n], \sum_j \lambda_{i,j} \cdot \mathbf{P}_\lambda^* \left[\bigcap_{k \in Y_{i,j}} \bar{B}_k \mid Z_{i,j} = 1 \right] > 1.$$

Then, if no B_k is a tautology, the CSP is feasible.

To prove the theorem, we will study the following probabilistic process. We are given a vector $p \in [0, 1]^N$ of probabilities, one for each indicator Z_x . Each Z_x is drawn *independently* as Bernoulli- p , i.e., $P(Z_x = 1) = p_x$. (If for

some event x we have $p_x > 1$, then by an abuse of notation, we take this to mean that $Z_x = 1$ with certainty.) Let $Y = \{B_1, B_2, \dots, B_\ell\}$ denote the set of all bad events. Our goal is to satisfy all the assignment constraints and avoid all the events in Y . If $Y' \subseteq Y$, we use the notation $\exists Y'$ to denote the event that some $B_i \in Y'$ occurs. So in this case, we want to *avoid* the event $\exists Y$.

We recall a basic lemma concerning increasing and decreasing events, which follows from the FKG inequality [11]:

Lemma 2.3. *Let $X_0 \subseteq [N]$. Suppose B_1 is an event which depends solely on variables X_1 , where X_1 is disjoint from X_0 , and let E^- be an decreasing event. Then,*

$$\mathbf{P}(\forall x \in X_0 \ Z_x = 1 \mid B_1, E^-) \leq \prod_{x \in X_0} p_x.$$

Similarly, if E^+ is increasing, then $\mathbf{P}(\forall x \in X_0 \ Z_x = 1 \mid B_1, E^+) \geq \prod_{x \in X_0} p_x$.

Proof. We will only prove the first part of this lemma; the second is analogous.

We average over all assignments to the variables Z_x , for $x \in X_1$. For any such assignment-vector \vec{z} , the event $\bigwedge_{x \in X_0} Z_x = 1$ is an increasing function, while E^- is a decreasing function in the remaining variables. Hence, by FKG, the probability of this event conditional on $(Z_{X_1} = \vec{z} \wedge E^-)$ is at most its conditional on $Z_{X_1} = \vec{z}$ alone. But, the events $\bigwedge_{x \in X_0} Z_x = 1$ and $Z_{X_1} = \vec{z}$ involve disjoint sets of variables, so they are independent. Hence this probability is at most the unconditional probability of $\bigwedge_{x \in X_0} Z_x = 1$, namely $\prod_{x \in X_0} p_x$. \square

If $\mathcal{A}' \subseteq \mathcal{A}$ is any subset of the blocks, we define the event $\text{Assigned}(\mathcal{A}')$ to be the event that, for all $i \in \mathcal{A}'$, there is *at least one* value of j for which $Z_{i,j} = 1$. Our goal is to satisfy the constraint $\text{Assigned}(\mathcal{A})$. Because all the bad events are increasing Boolean functions, if we can find a configuration in which each block has at least one value assigned, we can easily alter it to a feasible configuration in which each block has *exactly one* value assigned.

We are now ready to state the first lemma concerning this probabilistic process. We want to show that there is a positive probability of satisfying all the assignment constraints and avoiding all bad events. We will show the following key lemma by induction:

Lemma 2.4. *Let $0 < \epsilon < 1$. Suppose $p \in [0, 1]^N$ is a vector such that for all blocks i ,*

$$\sum_j p_{i,j} (\mathbf{P}_{p/\epsilon}^*(\neg \exists Y_{i,j} \mid Z_{i,j} = 1)) - \sum_{j,j'} p_{i,j} p_{i,j'} \geq \epsilon.$$

Then for any block i , any $Y' \subseteq Y$ a set of bad events, and any $\mathcal{A}' \subseteq \mathcal{A}$ a set of blocks, we have

$$\mathbf{P}(\text{Assigned}(i) \mid \neg \exists Y', \text{Assigned}(\mathcal{A}')) \geq \epsilon.$$

Proof. We show this by induction on $|Y'| + |\mathcal{A}'|$. We may assume that $i \notin \mathcal{A}'$, as otherwise this is vacuous. First, suppose $|Y'| = 0$. Then, $\mathbf{P}(\text{Assigned}(i) \mid \neg \exists Y', \text{Assigned}(\mathcal{A}'))$ equals $\mathbf{P}(\text{Assigned}(i))$ as these events are independent. By Inclusion-Exclusion, the latter probability is at least

$$\mathbf{P}(\text{Assigned}(i)) \geq \sum_j p_{i,j} - \sum_{j,j'} p_{i,j} p_{i,j'}$$

and it is easy to see that the lemma's hypothesized constraint implies that this is at least ϵ .

Next suppose $|Y'| > 0$. We use Inclusion-Exclusion to estimate $\mathbf{P}(\text{Assigned}(i) \mid \neg \exists Y', \text{Assigned}(\mathcal{A}'))$. First, consider the probability that a distinct pair j, j' in block i are jointly chosen, conditional on all these events. For this, by Lemma 2.3 we have

$$\mathbf{P}(Z_{i,j} = Z_{i,j'} = 1 \mid \text{Assigned}(\mathcal{A}'), \neg \exists Y') \leq p_{i,j} p_{i,j'} \quad (1)$$

as $i \notin \mathcal{A}'$ and $\neg \exists Y'$ is decreasing.

Let us fix j . We next need to show a lower bound on $\mathbf{P}(Z_{i,j} = 1 \mid \text{Assigned}(\mathcal{A}'), \neg \exists Y')$. This is easily seen to equal $\mathbf{P}(Z_{i,j} = 1)$ if $Y_{i,j} \cap Y' = \emptyset$, so we can assume $Y_{i,j} \cap Y' \neq \emptyset$. Using Bayes' Theorem, we interchange the events $Z_{i,j} = 1$ and $\exists Y_{i,j}$ in the conditional probability to obtain

$$\begin{aligned} & \mathbf{P}(Z_{i,j} = 1 \mid \text{Assigned}(\mathcal{A}'), \neg \exists Y') \\ & \geq p_{i,j} \cdot \mathbf{P}(\neg \exists Y_{i,j} \mid Z_{i,j} = 1, \text{Assigned}(\mathcal{A}'), \neg \exists (Y' - Y_{i,j})). \end{aligned} \quad (2)$$

This approach to handling a conditioning was inspired by [5].

Consider the random variables Z **conditioned on** the events $\text{Assigned}(\mathcal{A}'), \neg(Y - Y_{i,j}), Z_{i,j} = 1$. Our *key claim* now is that these conditional random variables Z (apart from $Z_{i,j}$ itself) satisfy $\text{UNC}(p/\epsilon)$. To show this, we need to upper-bound $\mathbf{P}(\mathcal{E}_1 \mid \mathcal{E}_2)$, where

$$\begin{aligned} \mathcal{E}_1 & \equiv (Z_{i'_1, j'_1} = \dots = Z_{i'_k, j'_k} = 1) \text{ and} \\ \mathcal{E}_2 & \equiv (\text{Assigned}(\mathcal{A}'), \neg \exists (Y - Y_{i,j}), Z_{i,j} = 1), \end{aligned}$$

for arbitrary k and $i'_1, j'_1, \dots, i'_k, j'_k$. Let $I' = \{i'_1, \dots, i'_k\}$ and $\mathcal{E}_3 \equiv (Z_{i,j} = 1, \text{Assigned}(\mathcal{A}' - I'), \neg \exists (Y - Y_{i,j}))$. By simple manipulations, we obtain

$$\mathbf{P}(\mathcal{E}_1 \mid \mathcal{E}_2) \leq \frac{\mathbf{P}(\mathcal{E}_1 \mid \mathcal{E}_3)}{\mathbf{P}(\text{Assigned}(I') \mid \mathcal{E}_3)}. \quad (3)$$

Note that \mathcal{E}_1 does not share any variables with $(Z_{i,j} = 1, \text{Assigned}(\mathcal{A}' - I'))$, and that $\neg \exists (Y - Y_{i,j})$ is a decreasing event. Hence by Lemma 2.3 the numerator of (3) is at most $p_{i'_1, j'_1} \dots p_{i'_k, j'_k}$. Now let us examine the denominator. The variable $Z_{i,j}$ does not affect any of the events mentioned in the denominator, so we may remove it from the conditioning: i.e., $\mathbf{P}(\text{Assigned}(I') \mid \mathcal{E}_3)$ equals

$$\mathbf{P}(\text{Assigned}(I') \mid \text{Assigned}(\mathcal{A}' - I'), \neg \exists (Y - Y_{i,j})),$$

which in turn is at least $\epsilon^{|I'|}$ by iterated application of the induction hypothesis.

Putting all of this together,

$$\mathbf{P}(\mathcal{E}_1 \mid \mathcal{E}_2) \leq p_{i'_1, j'_1} \dots p_{i'_k, j'_k} / \epsilon^{|I'|} \leq p_{i'_1, j'_1} \dots p_{i'_k, j'_k} / \epsilon^k.$$

So the random variables Z conditionally satisfy $\text{UNC}(p/\epsilon)$ and we have

$$\begin{aligned} & \mathbf{P}(\neg \exists Y_{i,j} \mid Z_{i,j} = 1, \text{Assigned}(\mathcal{A}'), \neg \exists (Y' - Y_{i,j})) \\ & \geq \mathbf{P}_{p/\epsilon}^*(\neg \exists Y_{i,j} \mid Z_{i,j} = 1) \end{aligned}$$

The right-hand side is substantially simpler, as there is *no conditioning to link the variables*. Substituting this into (1) and (2), we get

$$\begin{aligned} & \mathbf{P}(\text{Assigned}(i) \mid \text{Assigned}(\mathcal{A}'), \neg \exists Y') \\ & \geq \sum_j p_{i,j} \mathbf{P}_{p/\epsilon}^*(\neg \exists Y_{i,j} \mid Z_{i,j} = 1) - \sum_{j,j'} p_{i,j} p_{i,j'} \end{aligned}$$

and by our hypothesis the right-hand side is at least ϵ . \square

Definition 2.5. (Functions h and H_i) Suppose we are given a CSP and a vector $\lambda \in [0, \infty)^N$. For any element $x \in [N]$, let $h_x(\lambda) = \mathbf{P}_\lambda^*(\neg \exists Y_x \mid Z_x = 1)$. Then, for any block A_i , we define $H_i(\lambda) = \sum_j \lambda_{i,j} h_{i,j}(\lambda)$.

Having defined the H_i , we can now return to Lemma 2.4 and allow all entries of p to tend to 0 at the same rate, which leads to the following proof of the Assignment LLL:

Proof. (Of Theorem 2.2.) Given a $\lambda \in [0, \infty)^N$ such that $H_i(\lambda) > 1$ for all i , we need to show that the CSP is feasible. Let $p = \alpha\lambda$ and let $\epsilon = \alpha$ for some $\alpha > 0$. For α sufficiently small, p is a probability vector. In order to use Lemma 2.4, it suffices to satisfy the constraint for all i

$$\sum_j p_{i,j} h_{i,j}(p/\epsilon) - \sum_{j,j'} p_{i,j} p_{i,j'} \geq \epsilon. \quad (4)$$

Let us fix a block A_i . Suppose we allow $\alpha \rightarrow 0$. In this case, (4) will be satisfied for some $\alpha > 0$ sufficiently small if we have

$$\sum_j \lambda_{i,j} \cdot \mathbf{P}_\lambda^*(\neg \exists Y_{i,j} \mid Z_{i,j} = 1) > 1$$

As there are only finitely many blocks, there is $\alpha > 0$ sufficiently small which satisfies all constraints simultaneously.

In this case, we claim that there is a positive probability of satisfying $\text{Assigned}(A_i)$, \bar{B}_k for all blocks A_i and all bad events B_k , when we assign variables Z independently Bernoulli- p . First, $\mathbf{P}(\neg \exists Y) \geq \prod_{x \in [N]} \mathbf{P}(Z_x = 0)$, since no B_k is a tautology; the latter product is clearly positive for small-enough α . Next, by Lemma 2.4 and Bayes' Theorem,

$$\mathbf{P}(\text{Assigned}(1) \wedge \cdots \wedge \text{Assigned}(n) \mid \neg \exists Y) \geq \prod_{i=1}^n \epsilon > 0.$$

In particular, there is a configuration of the Z_x which satisfies all the constraints simultaneously. \square

3. COMPUTING \mathbf{P}^*

In the usual LLL, one can fully specify the underlying random process, so one can compute the probability of a bad event fairly readily. In the assignment LLL, we know that the random variables must satisfy their UNC constraints, but we do not know the full distribution of these variables. This can make it much harder to bound the probability of a bad event.

Roughly speaking, the UNC constraints force the underlying variables to be negatively correlated (or independent). For some types of bad events, this is enough to give strong bounds:

Lemma 3.1. For random variables Z_{x_1}, \dots, Z_{x_k} , let $\mu = \lambda_{x_1} + \cdots + \lambda_{x_k}$. Then

$$\mathbf{P}_\lambda^*(Z_{x_1} + \cdots + Z_{x_k} \leq \mu(1 + \delta)) \geq 1 - \left(\frac{e^\delta}{(1 + \delta)^{1 + \delta}} \right)^\mu$$

Proof. The Chernoff upper-tail bound applies to negatively correlated random variables [23]. \square

Suppose we have an increasing bad event B which depends on Z_{x_1}, \dots, Z_{x_k} . We are given $\lambda \in [0, 1]^N$. Note that Ω is a probability distribution on Z_1, \dots, Z_N , but we

abuse notation to view it as a distribution on Z_{x_1}, \dots, Z_{x_k} as well. We describe a generic algorithm to compute $\mathbf{P}_\lambda^*(B)$ (we sometimes just denote $\mathbf{P}_\lambda^*(\cdot)$ as $\mathbf{P}^*(\cdot)$).

As B is an increasing event, we can write $B = a_1 \vee \cdots \vee a_n$, where each $a_i \in B$ is an atomic event, and is minimal in the sense that for all $a' < a_i$ we have $a' \notin B$. We assume $0 \notin B$, as otherwise B is a tautology and $\mathbf{P}^*(B) = 1$.

We say that a probability distribution Ω on the variables Z_{x_1}, \dots, Z_{x_k} is *worst-case* if $\mathbf{P}_\lambda^*(B) = \mathbf{P}_\Omega(B)$. By compactness, such an Ω exists. The basic idea of our algorithm is to view each $\mathbf{P}_\Omega(\omega)$ as an unknown quantity, where $\omega \in \Omega$ is an atomic event. We write $q_\omega = \mathbf{P}_\Omega(\omega)$ for simplicity. In this case, $\mathbf{P}_\Omega(B)$ is the sum

$$\mathbf{P}_\Omega(B) = \sum_{\omega \in B} q_\omega$$

Furthermore, the UNC constraints can also be viewed as linear constraints in the variable q_ω . For each x'_1, \dots, x'_k , we have the constraint

$$\sum_{\omega_{x'_1} = \cdots = \omega_{x'_k} = 1} q_\omega \leq \lambda_{x'_1} \cdots \lambda_{x'_k}$$

This defines an LP, in which we maximize the objective function $\mathbf{P}^*(B) = \sum_{\omega \in B} q_\omega$ subject to the UNC constraints. The size of this LP may be enormous, potentially including 2^k variables and 2^k constraints. However, in many applications k is a parameter of the problem which can be regarded as constant, so such a program may still be tractable. In general, we can reduce the number of variables and constraints of this LP with the following observations.

Proposition 3.2. There is a distribution Ω with $\mathbf{P}_\Omega(B) = \mathbf{P}^*(B)$ and such that Ω is only supported on the atomic events $\{0, a_1, \dots, a_n\}$.

Proof. Suppose Ω satisfies the UNC constraints. Then define Ω' as follows. For each atomic event ω , if $\omega \notin B$, then shift the probability mass of ω to 0; otherwise, shift the probability mass of ω to any minimal event a_i underneath it. This preserves all UNC constraints as well as the objective function. \square

For some types of bad events, there are certain symmetries among classes of variables. In this case, one can assume that the distribution Ω is symmetric in these variables; hence the probabilities of all such events can be collapsed into a single variable.

Proposition 3.3. Given a group $G \subseteq S_k$, where S_k is the symmetric group on k letters, define the group action of G on a probability distribution Ω by permutation of the indices and on λ by permutation of the coordinates. Suppose B, λ are closed under the action of G . Then there is a worst-case probability distribution Ω which is closed under G . For this probability distribution Ω , we only need to keep track of a single unknown quantity q' for each orbit of G .

Proof. Given a worst-case Ω , let $\Omega' = \frac{1}{|G|} \sum_{g \in G} g\Omega$. As B is closed under G , each of the distributions $g\Omega$ has the same probability of the bad event B . Since λ is closed under G , all the UNC constraints are preserved in each $g\Omega$. \square

We will show how to use these techniques to compute \mathbf{P}^* for various applications in later sections.

4. THE LLL DISTRIBUTION

If we are given a distribution λ satisfying $H_i(\lambda) > 1$, then we know that there exists a configuration which satisfies all constraints. Furthermore, such a configuration can be found by drawing the indicators Z as independent Bernoulli- $\alpha\lambda$, for some $\alpha > 0$ sufficiently small.

We may wish to learn more about such configurations, other than that they exist. We can use the probabilistic method, by defining an appropriate distribution on the set of feasible configurations. We define the LLL-distribution to be the distribution on feasible configurations Γ that is induced by the following process:

1. Draw each Z_x independently with probability $\alpha\lambda$.
2. If the resulting configuration Γ does not satisfy all constraints $\text{Assigned}(\mathcal{A}), \neg\exists Y$, return to step (1).
3. Now each block has at least one element selected. If any block has more than one element, arbitrarily discard all but one.
4. Return the resulting feasible configuration Γ' .

We refer to step (3) as the *alteration step* and step (2) as the *configuration step*.

For $\alpha > 0$ sufficiently small, this process terminates with positive probability, returning a feasible configuration Γ' . Such configuration has exactly one element selected from each block. We refer to this distribution as \mathcal{L}_α .

Note that there are finitely many configurations, and for each of these the probability of $\mathbf{P}(\Gamma \mid L_\alpha)$ can be viewed as a rational function of the parameter α . Hence as $\alpha \rightarrow 0$, $\mathbf{P}(\Gamma \mid L_\alpha)$ must converge. Furthermore, since for any $\alpha > 0$ sufficiently small, $\mathbf{P}(\Gamma \mid L_\alpha)$ defines a probability distribution, so must the limit as $\alpha \rightarrow 0$. We define the distribution \mathcal{L} to be the limiting distribution as $\alpha \rightarrow 0$, and refer to \mathcal{L} as the *LLL distribution*. The reason for this limiting step is that typically the distribution \mathcal{L} gives simpler and tighter bounds than any \mathcal{L}_α . Note however that there is no explicit process that samples from \mathcal{L} .

We show that in the LLL distribution, the probability that an element x is accepted does not differ too much from λ_x :

Theorem 4.1. *Suppose we are given real numbers $\lambda \in [0, 1]^N$ such that for all i we satisfy $H_i(\lambda) > 1$. Then, the distribution \mathcal{L} is well-defined. Also, for any i, j we have*

$$\lambda_{i,j} \frac{h_{i,j}(\lambda)}{h_{i,j}(\lambda)\lambda_{i,j} + \sum_{j' \neq j} \lambda_{i,j'}} \leq \mathbf{P}(Z_{i,j} = 1 \mid \mathcal{L}) \leq \lambda_{i,j}.$$

Proof. As in Theorem 2.2, we set $p = \alpha\lambda$ and $\epsilon = \alpha$ for some $\alpha > 0$ sufficiently small so that

$$\sum_j p_{i,j} h_{i,j}(p/\epsilon) - \sum_{j,j'} p_{i,j} p_{i,j'} \geq \epsilon$$

First, we show an easy upper bound on $\mathbf{P}(Z_{i,j} = 1)$:

$$\begin{aligned} \mathbf{P}(Z_{i,j} = 1 \mid \mathcal{L}_\alpha) &\leq \mathbf{P}(Z_{i,j} = 1 \mid \text{Assigned}(\mathcal{A}), \neg\exists Y) \\ &\leq p_{i,j}/\epsilon \quad \text{by Lemmas 2.3, 2.4} \\ &= \lambda_{i,j} \end{aligned}$$

Next, a lower bound on the probability of having $Z_{i,j} = 1$ after the alteration step is given by

$$\begin{aligned} \mathbf{P}(Z_{i,j} = 1 \mid \mathcal{L}_\alpha) &\geq P(Z_{i,j} = 1 \mid \text{Assigned}(\mathcal{A}), \neg\exists Y) \\ &\quad - 2 \sum_{j < j'} \mathbf{P}(Z_{i,j} = Z_{i,j'} = 1 \mid \text{Assigned}(\mathcal{A}), \neg\exists Y) \end{aligned}$$

Let us examine these in turn. We want to lower-bound $\mathbf{P}(Z_{i,j} = 1 \mid \text{Assigned}(\mathcal{A}), \neg\exists Y)$ and to upper-bound $\mathbf{P}(Z_{i,j} = Z_{i,j'} = 1 \mid \text{Assigned}(\mathcal{A}), \neg\exists Y)$. We define the event

$$\mathcal{E} \equiv \text{Assigned}(\mathcal{A} - i) \wedge \neg\exists Y.$$

We have

$$\begin{aligned} \mathbf{P}(Z_{i,j} = Z_{i,j'} = 1 \mid \text{Assigned}(\mathcal{A}), \neg\exists Y) &= \\ &= \frac{\mathbf{P}(Z_{i,j} = Z_{i,j'} = 1 \wedge \text{Assigned}(i) \mid \mathcal{E})}{\mathbf{P}(\text{Assigned}(i) \mid \mathcal{E})} \\ &\leq \frac{p_{i,j} p_{i,j'}}{\epsilon} \quad \text{by Lemmas 2.3, 2.4} \\ &= O(\alpha) \end{aligned}$$

Next, we lower-bound $P(Z_{i,j} = 1 \mid \text{Assigned}(\mathcal{A}), \neg\exists Y)$ as

$$\begin{aligned} \mathbf{P}(Z_{i,j} = 1 \mid \text{Assigned}(\mathcal{A}), \neg\exists Y) &= \\ &= \frac{\mathbf{P}(Z_{i,j} = 1 \mid \mathcal{E})}{\mathbf{P}(\text{Assigned}(i) \mid \mathcal{E})} \\ &= \frac{\mathbf{P}(Z_{i,j} = 1 \mid \mathcal{E})}{\sum_{j'} \mathbf{P}(Z_{i,j'} = 1 \mid \mathcal{E}) - \sum_{j',j''} \mathbf{P}(Z_{i,j'} = Z_{i,j''} = 1 \mid \mathcal{E})} \\ &\geq \frac{p_{i,j} \mathbf{P}_{p/\epsilon}(\neg\exists Y_{i,j} \mid Z_{i,j} = 1)}{p_{i,j} \mathbf{P}_{p/\epsilon}(\neg\exists Y_{i,j} \mid Z_{i,j} = 1) + \sum_{j' \neq j} p_{i,j'} - O(\alpha^2)} \\ &= \lambda_{i,j} \frac{h_{i,j}(\lambda)}{h_{i,j}(\lambda)\lambda_{i,j} + \sum_{j' \neq j} \lambda_{i,j'} - O(\alpha)} \end{aligned}$$

This implies that the probability of $Z_{i,j} = 1$ in the distribution \mathcal{L}_α is at least

$$\mathbf{P}(Z_{i,j} = 1 \mid \mathcal{L}_\alpha) \geq \frac{\lambda_{i,j} h_{i,j}(\lambda)}{\lambda_{i,j} h_{i,j}(\lambda) + \sum_{j' \neq j} \lambda_{i,j'} - O(\alpha)} - O(\alpha)$$

As $\alpha \rightarrow 0$, we have

$$\mathbf{P}(Z_{i,j} = 1 \mid \mathcal{L}) \geq P(Z_{i,j} = 1 \mid \mathcal{L}_\alpha) \geq \frac{\lambda_{i,j} h_{i,j}(\lambda)}{\lambda_{i,j} h_{i,j}(\lambda) + \sum_{j' \neq j} \lambda_{i,j'}} \quad \square$$

Suppose we are given non-negative weights $w_x \geq 0$. By drawing from the LLL distribution, we can show the existence of configurations with high or low weights:

Lemma 4.2. *Suppose we are given $\lambda \in [0, 1]^N$ such that for all blocks, $H_i(\lambda) > 1$.*

Fix a block i . To simplify notation, let $|A_i| = l$ with the elements sorted by weight so that $0 \leq w_{i,1} \leq w_{i,2} \leq \dots \leq w_{i,l}$. We define the upper and lower weights for block i as

$$\begin{aligned} W_i^- &= \min_{1 < k < l} \frac{\sum_{j=1}^k \lambda_j w_{i,j} + \sum_{j=k+1}^l \lambda_{i,j} h_{i,j} w_{i,j}}{\sum_{j=1}^k \lambda_{i,j} + \sum_{j=k+1}^l \lambda_{i,j} h_{i,j}} \\ W_i^+ &= \max_{1 < k < l} \frac{\sum_{j=1}^k \lambda_{i,j} w_{i,j} + \sum_{j=k+1}^l \lambda_{i,j} h_{i,j} w_{i,j}}{\sum_{j=1}^k \lambda_{i,j} + \sum_{j=k+1}^l \lambda_{i,j} h_{i,j}} \end{aligned}$$

Then the expected weight of block i in the LLL distribution lies in $[W_i^-, W_i^+]$.

Proof. The proof is very similar to Theorem 4.1. \square

Corollary 4.3. *Suppose for all blocks $H_i(\lambda) > 1$. Then there is a feasible configuration with weight at most $W^+ = \sum_{i=1}^n W_i^+$. There is also a feasible configuration with weight at least $W^- = \sum_{i=1}^n W_i^-$.*

5. DIRECT APPLICATIONS

5.1 Independent transversals

It is well-known that a graph G with n vertices and average degree d has an independent set of size at least $n/(d+1)$. What if we want independent sets with much more structure? Suppose we are given a partition of the vertices into sets $V = V_1 \sqcup \dots \sqcup V_k$, and we wish to select one vertex from each set so that the resulting set is an independent set. (Such a set is known as an *independent transversal*.)

One important parameter for the independent transversal problem is the size of the classes, i.e., requiring $|V_i| \geq b$ for all $i = 1, \dots, k$. Alon gives a short LLL-based proof that a sufficient condition for such an independent transversal to exist is to require $b \geq 2e\Delta$ [1], where Δ is the maximum degree of any vertex in the graph. Haxell provides an elegant topological proof that a sufficient condition is $b \geq 2\Delta$ [15]. The condition of [15] is existentially optimal, in the sense that no condition of the form $b \geq c\Delta$ is possible for $c < 2$.

These bounds are all given in terms of the *maximum degree* Δ , which can be a crude statistic. The proof of [15] adds vertices one-by-one to partial transversals, which depends very heavily on bounding the maximum degree of any vertex. Suppose we let d denote the maximum average degree of any class V_i . This is a more flexible statistic than Δ . Using Pegden's version of the algorithm LLL, we can show that $b \geq 4d$ suffices [24] (and this is constructive). We will prove this same result using our version of the LLL. This proof offers no advantages over Pegden's approach, but illustrates how our LLL can be applied in practice.

Theorem 5.1. *Suppose we have a graph $G = (V, E)$ whose vertex set is partitioned as $V = V_1 \sqcup \dots \sqcup V_k$, where for each i , (i) the average degree of the vertices in V_i is at most d , and (ii) $|V_i| \geq b > 4d$. Then G has an independent transversal.*

Proof. If any $|V_i|$ is greater than b , we may discard its highest-degree vertices to bring it down to size b exactly, while decreasing its average degree. In this way, we may assume that $|V_i| = b$ for all i . Our constraint is that for each edge $f \in E$, at most one incident vertex is selected. To use Theorem 2.2, we set $\lambda = \vec{\beta}$ for some $\beta > 0$ to be chosen shortly. Fix any $v \in V$ with degree d_v , from a class V_i . Then the event that all bad events Y_v are avoided, given that v itself is selected, is just the event that no neighbor of v is selected. This occurs with probability $h_v \geq 1 - d_v\beta$. So we have

$$H_i(\lambda) = \sum_{v \in V_i} \beta(1 - d_v\beta) \geq b\beta(1 - d\beta).$$

Setting $\beta = \frac{1}{2d}$ shows that $b > 4d$ is a sufficient condition for an independent transversal to exist. \square

We can also use our weighting condition to bound the *weight* of such an independent transversal, which is not possible using [15]. Suppose that we are given weights w_v for each vertex of G . There is a simple argument that G has an

independent set of weight at least $\frac{w(V)}{\Delta+1}$. We can also give a similar bound on the weight of an independent transversal:

Theorem 5.2. *Suppose we have a graph $G = (V, E)$ with V partitioned into classes of size $b > 4\Delta$. Then, G has an independent transversal of weight at least $w(V) \frac{\sqrt{b-4\Delta} + \sqrt{b}}{\sqrt{b-4\Delta} + \sqrt{b(2b-1)}}$, which approaches $w(V)/b$ for large b .*

Proof. Define

$$\alpha^* = \frac{1 - \sqrt{\frac{b-4\Delta}{b}}}{2\Delta}$$

Suppose we set $\vec{\lambda} = \alpha$, where $\alpha = \alpha^* + \epsilon$. Then, for $\epsilon > 0$ sufficiently small, we satisfy the constraint of Corollary 4.3. Hence there exists an independent transversal of weight at least W^- .

Our bound for W_i^- depends in a rather complicated way on the distribution of weights within a block. If our goal is to show a lower bound on the form $cw(V)$ for some constant c , then the worst-possible distribution of weights is that a single vertex in each block has all the weight. In this case, our bound W^- gives us that

$$\begin{aligned} W^- &\geq \sum_i \frac{\alpha h w(V_i)}{(b-1)\alpha + \alpha h} \\ &= w(V) \frac{1 - \alpha\Delta}{b - \alpha\Delta} \end{aligned}$$

Summing over all blocks, we see that the expected weight approaches arbitrarily close to $\frac{w(V)(1-\alpha^*\Delta)}{b-\alpha^*\Delta}$ as $\epsilon \rightarrow 0$. As the set of all independent transversals is finite, this implies the existence of an independent transversal of weight at least

$$w(V) \cdot \left(\frac{\sqrt{b-4\Delta} + \sqrt{b}}{\sqrt{b-4\Delta} + \sqrt{b(2b-1)}} \right).$$

\square

5.2 Chromatic capacity

Rödl has shown that for any acyclic digraph D , there exists an undirected graph D^* such that all of the latter's acyclic orientations contain D as an induced subgraph [26]. Cochand & Duchet introduced the notion of *chromatic capacity* to explicitly construct D^* , using graphs that have a large value for this parameter [8]. Given an undirected multi-graph G , the *chromatic capacity* $\chi_{\text{cap}}(G)$ is the largest integer t such that there is an edge t -labeling such that, for any vertex t -labeling, some edge shares the same color as both its endpoints. That is,

$$\exists f : E \rightarrow [t] \forall g : V \rightarrow [t] \exists \langle u, v \rangle \in E \quad g(u) = g(v) = f(u, v)$$

See [6, 9] for further study of this parameter.

It is shown in [3, 8] that $\chi_{\text{cap}}(G) \leq \sqrt{2e\Delta} - e - 1$. We can use our results for independent transversals to show that the chromatic capacity is at most $2\sqrt{\Delta}$. (A value proportional to $\sqrt{\Delta}$ is sometimes of the correct order, as shown by the complete graph [8, 9].)

Theorem 5.3. *Let Δ denote the maximum degree of a graph G . Then for any graph G , its chromatic capacity is at most $2\sqrt{\Delta}$.*

Proof. Given graph G with an edge-labeling f on $t+1$ colors, construct the graph H as follows. For each vertex $v \in G$ and

each color $j \in [t+1]$, construct a vertex $(v, j) \in H$. For each edge $e = \langle u, v \rangle \in G$ with color j , construct an edge in H between (u, j) and (v, j) . Now partition the vertices of H by grouping all the vertices $(v, 1), \dots, (v, t+1)$ into a single class.

We can now observe the following simple facts:

1. Any independent transversal of H corresponds to a unique vertex-labeling of G which avoids $f(u, v) = g(u) = g(v)$, and vice-versa.
2. The classes in H contain exactly $t+1$ vertices.
3. For each vertex $v \in G$, the *average degree* of the corresponding class in H is $\frac{d_v}{t+1}$.

By Theorem 5.1, as long as $t+1 > 4 \frac{\Delta}{t+1}$, there exists an independent transversal of H and hence a valid vertex-labeling of G . This is satisfied when $t = 2\sqrt{\Delta}$. \square

We can use this result almost immediately to improve a related bound of Archer [3] on the *upper chromatic number* of the reals. The upper chromatic number of a metric space S is related to distance-based color-constraints on S [13]. The argument of [3] uses the chromatic capacity to show that the “ k^{th} upper chromatic number” $\hat{\chi}^{(k)}$ of the reals under the Euclidean distance, is at most $\lceil 4ek \rceil$. Our bound on the chromatic capacity above, shows that $\hat{\chi}^{(k)}(\mathbb{R}) \leq 8k$.

5.3 Integrality gap of column-sparse packing problems

Consider our family of CSPs where we have a series of linear packing constraints: indexed by k and of the form “ $\sum_{i,j} a_{k,i,j} x_{i,j} \leq b_k$ ”, with non-negative $a_{k,i,j}, b_k$. In addition, there is the usual assignment constraint, namely a series of sets A_1, \dots, A_n with the constraints $\sum_{j \in A_i} x_{i,j} = 1$. When does such an integer linear program have a feasible solution? Suppose we wish to solve this via LP relaxation. One technique is to start with the LP where the integrality constraints on $x_{i,j} \in \{0, 1\}$ are relaxed to $x'_{i,j} \in [0, 1]$, and with the packing constraints tightened to $\sum_{i,j} a_{k,i,j} x'_{i,j} \leq b'_k$ for some $b'_k \leq b_k$.

The notion of “covering radius” d here is obvious: as in [4, 17], we assume that for each (i, j) , there are at most d constraints k with $a_{k,i,j} \neq 0$. Consider, for notational simplicity, the case where $a_{k,i,j} \in \{0, 1\}$. We will show that a bound of

$$b_k \geq \left(1 + \frac{1}{\text{poly}(d)}\right) b'_k + \Omega(\sqrt{b'_k \log d})$$

suffices to guarantee the existence of an integral solution.

We state this theorem (with notation such as d etc. as above) very carefully, as the various parameters depend on each other delicately:

Theorem 5.4. *Let $c_1, c_2 > 0$ be any constants. There is some constant $c_3 \geq 0$ with the following property. If the LP*

$$\forall i, \sum_{j \in A_i} y_{i,j} \geq 1; \quad \forall k, \sum_{i,j} a_{k,i,j} y_{i,j} \leq b'_k; \quad \forall (i, j), y_{i,j} \in [0, 1]$$

*is satisfiable, then so is the **integer program***

$$\forall i, \sum_{j \in A_i} x_{i,j} = 1; \quad \forall k, \sum_{i,j} a_{k,i,j} x_{i,j} \leq b_k; \quad \forall (i, j), x_{i,j} \in [0, 1]$$

where for all k we have $b_k \geq b'_k(1 + d^{-c_1}) + c_3 \sqrt{b'_k \log d}$ and $b'_k \geq c_2 \log d$.

Proof. We suppose $d > 1$, as when $d = 1$ the IP is trivial. Let us fix $c_1, c_2 > 0$; we will show how to choose c_3 appropriately. Define $\epsilon = d^{-c_1}$. Suppose we have a feasible solution y to the relaxed linear program. In the notation of Theorem 2.2, set $\lambda_{i,j} = (1 + \epsilon)y_{i,j}$ and let Z be the associated random vector. (In case $\lambda_{i,j} > 1$, we set $x_{i,j} = 1$; this only helps our analysis. Although we do not discuss the details here, the reader can assume that $\lambda_{i,j} \leq 1$ for simplicity.)

For any given pair (i, j) , there are most d constraints that involve $Z_{i,j}$; consider any such constraint k . In order to avoid the bad event conditional on $Z_{i,j} = 1$, we must have the remaining variables in that constraint sum to at most $b_k - 1$. The UNC constraints apply to these variables. So the sum of all such variables has mean $\mu = \sum_{(i',j') \neq (i,j)} a_{k,i',j'} (1 + \epsilon) y_{i',j'} \leq (1 + \epsilon) b'_k$. We can bound the probability that such a sum deviates from its mean by a Chernoff bound under negative correlation [23], where $\exp(x)$ denotes e^x :

$$\begin{aligned} P_\lambda^*(\text{bad event } k \mid Z_{i,j} = 1) &\leq \exp\left(-\frac{(b_k - \mu)^2}{(2 + (b_k/\mu - 1))\mu}\right) \\ &\leq \exp\left(-\frac{(b_k - (1 + \epsilon)b'_k)^2}{(2 + (\frac{b_k}{(1+\epsilon)b'_k} - 1))b'_k}\right) \\ &\leq \exp\left(-\frac{c_3^2 \log d}{(2 + c_2^2 \sqrt{b'_k \log d})(1 + d^{-c_1})}\right) \end{aligned}$$

By our assumption on $b'_k \geq c_2 \log d$ and our assumption $d > 1$, we can simplify this as $\exp(-c \log d)$, where c is a constant which increases unboundedly with c_3 . Hence, in the notation of Definition 2.5,

$$h_{i,j}(\lambda) \geq 1 - \sum_{k: a_{k,i} = 1} \exp(-c \log d) \geq 1 - d^{1-c}.$$

Now, summing over all $j \in A_i$, we have

$$\begin{aligned} H_i(\lambda) &\geq \sum_j \lambda_{i,j} (1 - d^{1-c}) \\ &\geq \sum_j (1 + d^{-c_1}) y_{i,j} (1 - d^{1-c}) \\ &\geq (1 + d^{-c_1}) (1 - d^{1-c}), \end{aligned}$$

which can be made larger than 1 by choosing c_3 sufficiently large, for any value $d > 1$. Hence, by Theorem 2.2, there is a configuration x that satisfies all the IP constraints. \square

6. PACKET ROUTING

6.1 Overview

We will examine the packet routing problem in much more detail than the other problems we have considered. This analysis will use our assignment LLL. However, many of the improvements we make will be much more problem-specific. We will improve some choices of parameters, as well as examine more closely instances in which the congestion is controlled on very small scales. So, we will first examine the packet-routing problem using the standard LLL; this extends the analysis of [29] and [25]. We will then show how to enhance this to handle the nonlinear increasing functions that arise, by using our assignment LLL.

We begin by reviewing the basic strategy of [29], and its improvement by [25]. The former has a very readable overview of our basic strategy, and we will not include all the details which are covered there. Our choice of parameters will be slightly improved from [29] and [25]. We note that [25] studied a more general version of the packet-routing problem, so their choice of parameters was not (and could not be) optimized.

We are given a graph G with N packets. Each packet has a simple path, of length at most D , to reach its endpoint vertex (we refer to D as the *dilation*). In any timestep, a packet may wait at its current position, or move along the next edge on its path. Our goal is to find a schedule of smallest makespan in which, in any given timestep, an edge carries at most a single packet. We define the *congestion* C to be the maximum, over all edges, of the number of packets scheduled to traverse that edge. It is clear that D and C are both lower bounds for the makespan, and [20] has known that in fact a schedule of makespan $O(C + D)$ is possible. [29] provided an explicit constant bound of $39(C + D)$, as well as describing an algorithm to find such a schedule.

While the final schedule only allows one packet to cross an edge at a time, we will relax this constraint during our construction. We consider “infeasible” schedules, in which each packet follows its path but where arbitrarily many packets may pass through each edge at each timestep.² We define an *interval* to be a consecutive set of times in our schedule, and the *congestion* of an edge in a given interval to be the number of packets crossing that edge. If we are referring to intervals of length i , then we define a *frame* to be an interval which starts at an integer multiple of i .

One can easily form an (infeasible) schedule with delay D and overall congestion C . Initially, this congestion may “bunch up” in time, that is, certain edges may have very high congestion in some timesteps and very low congestion in others. So the congestion is not bounded on any smaller interval than the trivial interval of length D . During our construction, we will “even out” the schedule, bounding the congestion on successively smaller intervals.

Ideally, one would eventually finish by showing that on each individual timestep (i.e., interval of length 1), the congestion is roughly C/D . In this case, one could turn such an infeasible schedule into a feasible schedule, by simply expanding each timestep into C/D separate timesteps.

As [25] showed, it suffices to control the congestion on intervals of length 2. Given our infeasible schedule, we can view each interval of length 2 as defining a new subproblem. In this subproblem, our packets start at a given vertex and have paths of length 2. The congestion of this subproblem is exactly the congestion of the schedule. Hence, if we can schedule problems of length 2, then we can also schedule the 2-intervals of our expanded schedule.

Proposition 6.1. ([25]) *Suppose there is a instance G with delay $D = 2$ and congestion C . Then there is a feasible schedule of makespan $C + 1$. Furthermore, such a schedule can be formed in polynomial time.*

The work of [25] used this to improve the bound on the makespan to $23.4(C + D)$; it also speculated that by examining the scheduling for longer, but still small, delays, one

²Unless specified as “feasible”, a “schedule” will refer to a possibly infeasible schedule.

could further improve the general packet routing. Unfortunately, we are not able to show a general result for small delays such as $D = 3$. However, as we will see, the schedules that are produced in the larger construction of [29] are far from generic; rather they have relatively balanced congestion on small scales. We will see how to take advantage of this balanced structure to improve the scheduling.

We next review the general construction of [29].

6.2 Using the LLL to find a schedule

The general strategy for this construction is to add random delays to each packet, and then allowing the packet to move through each of its edges in turn without hesitation. This effectively homogenizes the congestion across time. We have the following lemma:

Lemma 6.2. *Let $i' < i$, and let m, C' be non-negative integers. Suppose there is a (possibly infeasible) schedule S of length L such that every interval of length i has congestion at most C . Suppose that we have*

$$e \times P(\text{Binomial}(C, \frac{i'}{i-i'}) > C') \times (Cmi^2 + 1) < 1.$$

Then there is a (possibly infeasible) schedule S' of length $L' = L(1 + 1/m) + i$, in which every interval of length i' has congestion $\leq C'$. Furthermore, S' can be constructed in polynomial time.

Proof. We break the schedule S into frames of length $F = mi$, and refine each separately. Within each frame, we add a random delay of length $i - i'$ to each packet separately.

Let us fix an F -frame for the moment. Associate a bad event to each edge f and i' -interval I , that the congestion in that interval and edge exceeds C' .

For each I, e , there are at most C possible packets that could cross, and each does so with probability $p = \frac{i'}{i-i'}$. Hence the probability of the bad event is at most the probability that a Binomial random variable with C trials and probability p exceeds C' .

Next consider the dependency. Given an edge f and i' -interval I , there are at most C packets crossing it, each of which may pass through up to mi other edges in the frame. We refer to the combination of a specific packet passing through a specific edge as a *transit*. Now, for each transit, there are (depending on the delay assigned to that packet) at most i other i' -intervals in which this transit could have been scheduled. Hence the dependency is at most Cmi^2 .

By the LLL, the condition in the hypothesis guarantees that there is a positive probability that the delays avoid all bad events. In this case, we refine each frame of S to obtain a new schedule S' as desired. We can use the algorithmic LLL to actually find such schedules S' in polynomial time.

So far, this ensures that *within each frame*, the congestion within any interval of length i' is at most C' . In the refined schedule S' there may be intervals that cross frames. To ensure that these do not pose any problems, we insert a delay of length i' between successive frames, during which no packets move at all. This step means that the schedule S' may have length up to $L(1 + 1/m) + i$. \square

We note one important distinction between this analysis and that of Scheideler [29]. In Scheideler, one associates a bad event to each edge; we have a bad event corresponding to each edge and interval.

Using Lemma 6.2, we can transform the original problem instance (in which C, D may be unbounded), into one in which C, D are small. In order to carry out this analysis properly, one would need to develop a series of separate bounds depending on the sizes of C, D . To simplify the exposition, we will assume that C, D are very large, in which case certain rounding effects can be disregarded. When C, D are smaller, we can show stronger bounds but doing this completely requires an extensive case analysis of the parameters.

Lemma 6.3. *Assume $C + D \geq 2^{896}$. There is a schedule of length at most $1.004(C + D)$ and in which the congestion on any interval of length 2^{24} is at most 17,040,600. Furthermore, this schedule can be produced in polynomial time.*

Proof. Define the sequence a_k recursively as follows.

$$a_0 = 256 \quad a_{k+1} = 2^{a_k}$$

There is a unique k such that $a_k^{3.5} \leq (C + D) < a_{k+1}^{3.5}$. By a slight variant on Lemma 6.2, one can add delays to obtain a schedule of length $C + D$, in which the congestion on any interval of length $i' = a_k^3$ is at most $C' = i'(1 + 4/a_k)$.

At this point, we use Lemma 6.2 repeatedly to ensure to control the congestion intervals of length a_j^3 , for $j = k - 1, \dots, 0$. At each step, this increases the length of the resulting schedule from L_j to $L_j(1 + 1/a_{j+1}) + a_j$, and increases the congestion on the relevant interval from $i(1 + 4/a_k)$ to

$$i(1 + 4/a_k) \prod_{j=0}^{k-1} (1 + 4/a_j) \left(\frac{1}{1 - (a_j/a_{j+1})^3} \right)$$

(We use the Chernoff bound to estimate the binomial tail in Lemma 6.2.)

For $C + D \geq a_k^{3.5}$, it is a simple calculation to see that the increase in length is from $C + D$ (after the original refinement) to at most $1.004(C + D)$. In the final step of this analysis, we are bounding the congestion of intervals of length $a_0^3 = 2^{24}$, and the congestion on such an interval is at most 17040600.

Furthermore, since all of these steps use the LLL, one can form all such schedules in polynomial time.

See [29] for a much more thorough explanation of this process. \square

Now that we have reduced to constant-sized intervals, we are no longer interested in asymptotic arguments (which use generic bounds such as the Chernoff bound), and we come down to specific numbers.

Lemma 6.4. *There is a feasible schedule of length at most $10.92(C + D)$, which can be constructed in polynomial time.*

Proof. For simplicity, we assume $C + D \geq 2^{896}$.

By Lemma 6.3, we form a schedule S_1 , of length $L_1 \leq 1.004(C + D)$, in which each interval of length 2^{24} has congestion at most 17040600.

Now apply Lemma 6.2 to S_1 , with $m = 64, i' = 1024, C' = 1385$ to obtain a schedule S_2 , of length $L_2 \leq 1.0157L_1 + 2^{24}$, in which each interval of length 1024 has congestion at most 1385.

Now apply Lemma 6.2 to S_2 with $m = 64, i' = 2, C' = 20$, to obtain a schedule S_3 of length $L_3 \leq 1.0157L_2 + 1024$, in which each frame of length 2 has congestion at most 20.

Now apply Proposition 6.1 to S_3 , expanding each 2-frame to a *feasible* schedule of length 21. The total length of the resulting schedule is at most $\frac{21}{2}L_3 \leq 10.92(C + D)$. \square

6.3 Better scheduling of the final 2-frame

Let us examine the last stage in the construction more closely. In this phase, we are dividing the schedule into intervals of length 2, and we want to control the congestion of each edge in each 2-frame.

For a given edge f and time t , we let $c_t(f)$ denote the number of packets scheduled to cross that edge in the four time steps of the original (infeasible) schedule.

Suppose we have two consecutive 2-frames starting at time t . The reason for the high value of C' in the final step of the above construction is that it is quite likely that $c_t + c_{t+1}$ or $c_{t+2} + c_{t+3}$ are much larger than their mean. However, it would be quite rare for *both* these bad events to happen simultaneously. We will construct a schedule in which we insert an “overflow” time between the 2-frames. This overflow handles cases in which either $c_t + c_{t+1}$ is too large or $c_{t+2} + c_{t+3}$ is too large.

Our goal will be to modify either of the 2-frames so as to ensure that the congestion is at most T . In order to describe our modification strategy, we first fix, for every packet and frame, a “first edge” and “second edge” in this frame. Some packets may only transit a single edge, which we will arbitrarily label as first or second. As we modify the schedule, some packets that initially had two transits scheduled will be left with only one; in this case, we retain the label for that edge. So, we may assume that every edge is marked as first or second and this label does not change.

We do this by shifting transits into the overflow time. For each 2-frame, there are two overflow times, respectively earlier and later. If we want to shift an edge to the later overflow time, we choose any packet that uses that edge as a second edge (if any), and reschedule the second transit to the later overflow time; similarly if we shift an edge to the earlier overflow time. See Figure 1.

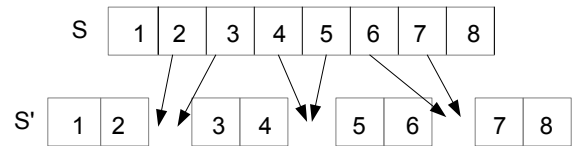


Figure 1: The packets in the original schedule S are shifted into overflow times in the schedule S' .

Note that in the analysis of [25], the only thing that matters is the *total* congestion of an edge in each 2-frame. In deciding how to shift packets into the overflow times, we need to be careful to account for how often the edge appears as the first or second transit. If an edge appears exclusively as a “first edge”, we will only be able to shift it into the earlier overflow, and similarly if an edge appears exclusively as a “second edge”.

Keeping this constraint in mind, our goal is to equalize as far as possible the distribution of edges into earlier and later overflows. We do this by the following scheme:

For each edge f and every odd integer $t = 1, 3, 5, \dots, L$, repeat while $c_t(f) + c_{t+1}(f) > T$:

1. If $c_t(f) = 0, c_{t+1}(f) > T$, then shift one packet into the later overflow time.
2. Else if $c_t(f) > T, c_{t+1}(f) = 0$, then shift one packet into the earlier overflow time.
3. Else if $c_t(f) + c_{t+1}(f) > T, c_t(f) > 0, c_{t+1}(f) > 0$, and $c_t(f) + c_{t+1}(f)$ is odd, then shift one packet into the earlier overflow time.
4. Else if $c_t(f) + c_{t+1}(f) > T, c_t(f) > 0, c_{t+1}(f) > 0$, and $c_t(f) + c_{t+1}(f)$ is even, then shift one packet into the later overflow time.

Suppose we fix t to be some odd integer. If we let c' denote the congestions at the end of this overflow-shifting process, then we have $c'_t(f) + c'_{t+1}(f) \leq T$, and the number of packets shifted into the earlier (respectively later) overflow time can be viewed as a function of the original values of the congestions c_t, c_{t+1} . We denote these functions by $O^-(c_t, c_{t+1}; T)$ and $O^+(c_t, c_{t+1}; T)$ respectively.

Specifically we have the following condition:

Proposition 6.5. *Suppose that we have a schedule of even length L , and let $c_t(f)$ for $t = 1, \dots, L$ denote the number of times f is scheduled as the t th edge of a packet. Suppose that for all edges $f \in E$ and all $t = 1, 3, 5, \dots$ we satisfy the constraint*

$$O^+(c_t(f), c_{t+1}(f); T) + O^-(c_{t+2}(f), c_{t+3}(f); T) \leq T'$$

as well as the boundary constraints

$$O^-(c_1(f), c_2(f)) \leq T' \text{ and } O^+(c_{L-1}(f), c_L(f)) \leq T'.$$

Then there is a feasible schedule of makespan $L \cdot (\frac{T+T'+2}{2}) + T'$, which can be constructed in polynomial time.

Proof. After the modification, each 2-frame has congestion at most T , while each overflow time has congestion at T' . Each overflow time has delay at most 2, since for any packet x , there may be at most two edges scheduled into that overflow time, namely the edge that had been originally marked as the second edge of the earlier 2-frame, and the edge that had been originally marked as the first edge of the latter 2-frame. Hence each 2-frame can be scheduled in time $T + 1$ and each overflow can be scheduled in time $T' + 1$. As there are $L/2$ 2-frames in the original schedule, there are $L/2 + 1$ overflow periods. Hence the total cost is at most $L \cdot (\frac{T+T'+2}{2}) + T'$. \square

Note that the conditions required by this Proposition 6.5 are local, in the sense that any violation is any event which affects an individual edge and a 4-interval which starts at an odd time t . We refer to such an interval for simplicity as an *aligned 4-interval*. We refer to the conditions required by this Proposition as the *4-conditions*; these conditions can be viewed as either pertaining to an entire schedule, or to an individual aligned 4-interval. We also note that the 4-conditions are *monotone*, in the sense that if a configuration violates them, then it will continue to do so if the congestion of any edge at any time is increased. We can use this to improve our bound:

Theorem 6.6. *There is a feasible schedule of length $8.84(C + D)$, which can be found in polynomial time.*

Proof. We assume $C + D \geq 2^{896}$ for simplicity.

As in the proof of Lemma 6.4, one obtains a schedule S of length $L = 1.0158 \times 1.004 \times 1.004(C + D)$ in which the congestion on any interval of length $i = 1024$ is at most $C = 1385$.

We divide S into frames of length $F = 1024m$ where $m = 64$, and add a random delay of length up to 1020 to each packet separately. This increases the length of the schedule up to $L' \leq 1.0157L + 1024$. We will first show that each frame individually satisfies the 4-conditions. So we may concentrate on a single such frame.

We associate a bad event to each aligned 4-interval I and edge f , that it violates the 4-conditions. It is not hard to see that the dependence of each such bad event is at most $Cmi^2/2$.

Now consider the probability of a bad event, that a given edge and interval has $O^+(c_t, c_{t+1}; T) + O^-(c_{t+2}, c_{t+3}; T) > T'$. (The boundary cases are similar and are omitted.) There are up to C packets which could affect the given f, I . For each such packet and each position within the interval, there is a probability of at most $1/(i - 4)$ that the packet x is scheduled at that time (conditional on any allocation to the other 4 positions). As the bad event is an increasing event, it suffices to suppose that the distribution of each c_t, \dots, c_{t+3} is iid Binomial with C trials and probability $1/(i - 4)$.

Now, one can simply enumerate over all possible values and count the total probability of satisfying Proposition 6.5. This is possible because we are dealing with a fixed, finite and relatively small choice of C . A computer program calculates this probability to be 3.9×10^{-12} for the choice $T = 8, T' = 7$.

One can verify that these parameters satisfy the LLL condition. In particular, such a schedule S' exists and can be found in polynomial time.

In order to ensure that the entire schedule satisfies the 4-frame conditions, one may insert a delay of length 2 between consecutive frames. This ensures that the overflow at the boundaries of the separate frames do not interfere with each other. Doing this inflates the schedule from length L' to length $L'(1 + 2/F) + 2 \leq 1.0158L + 1027$.

By Proposition 6.5, this schedule S' can be scheduled in makespan $8.5L' + 7 \leq 8.84(C + D)$. Note that all the constructions used here can be implemented in polynomial time. \square

6.4 The assignment LLL applied to packet routing

So far, all of the improvements we have made to the packet routing problem used nothing more than the conventional LLL. We now show how to modify this construction to use the assignment LLL in the appropriate places.

Let us examine more closely the process used to refine a schedule in which each interval of length C has congestion at most i . We break the schedule S into frames of length $F = mi$, and refine each separately. Within each frame, we add a random delay of length $b = i - i'$ to each packet separately. Let us fix an F -frame for the moment. We have an assignment problem, in which we must assign a delay to each packet. Our bad events here correspond to an edge receiving an excessive congestion in some time interval. The precise

meaning of an excessive congestion depends on which stage of the construction we are at. In the intermediate stages, all we care about is the total number of packets passing through that interval. In the final stage, we must satisfy the 4-conditions, which are more delicate conditions depending on the exact location of the packets within 4-intervals. In either case, these are local and increasing constraints.

We can modify Theorem 6.6 to use Theorem 2.2 instead of the LLL:

Proposition 6.7. *Let $i' < i$, and let m, C', k be non-negative integers. Suppose there is a schedule S of length L such that every interval of length i has congestion at most C for some C . Suppose also that for some $\lambda \in [0, 1]$ we have*

$$i \left(1 - mi' \left(\frac{e^\delta}{(1+\delta)^{1+\delta}} \right)^\mu \right) > 1$$

where $\mu = Ci'\lambda$ and $\delta = \frac{C'}{Ci'\lambda} - 1 \geq 0$

Then there is a schedule S' of length $L' = L(1+1/m) + i$, in which every interval of length i' has congestion $\leq C'$.

Proof. Suppose we add delays in the range $b = i - i'$ uniformly to each packet within each frame of length $F = mi$. For each edge f and i' -interval I , we introduce a bad event y that the congestion in the interval exceeds C' . We introduce a block corresponding to each packet, and an indicator $Z_{x,t}$ corresponding to the packet x having delay t . We select these with probability λ .

Fix a packet x and delay t . This packet x may pass through up to mi edges in the F -frame. If the delay t is adopted, then each transit of x occurs at a fixed time and edge, and will affect up to i' other intervals. Hence, x, t can affect at most mi' congestion events.

Next, fix a bad event $y = \langle f, I \rangle$ affected by delay x, t . We want to estimate the probability of violating the congestion constraint for y , given that $Z_{x,t}$ is set. This congestion constraint is violated if there are at least C' other packet delays (in addition to x, t) scheduled for f in interval I . There are at most C packets which could be scheduled to pass through the given edge, and there are i' possible delays which would contribute to the congestion of the given edge-interval. So, in all, there are at most Ci' delay variables which are relevant to y . Hence the probability of this event is $\mathbf{P}^*(\sum Z_{x,t} \geq C')$, where the random variables Z in the sum have mean at most $Ci'\lambda$. As shown in Proposition 3.1, the Chernoff bound applies to such sums. Hence $h_{x,t} \geq mi' \left(1 - \frac{e^\delta}{(1+\delta)^{1+\delta}} \right)^\mu$.

Thus,

$$H_x(\lambda) \geq i\lambda \times \left(1 - mi' \left(\frac{e^\delta}{(1+\delta)^{1+\delta}} \right)^\mu \right).$$

By Theorem 2.2, this suffices to show that a good configuration exists. \square

The final schedule is the most difficult to bound.

Proposition 6.8. *Let $m, C, i > 4, \lambda, T, T'$ be given. Suppose there is a schedule S of length L such that every interval of length i has congestion at most C . Suppose we choose delay variables independently with probability λ in the range $i - 4$, for each packet within each frame of length $F = mi$. Then within each frame we have*

$$H \geq (i-4)\lambda \left(1 - \frac{mi}{2} (p_1^* + p_2^* + p_3^* + p_4^*) \right)$$

where p_1^* is the probability \mathbf{P}^* of having

$$O^+(c_1 + 1, c_2) + O^-(c_3, c_4) > T'$$

where the congestions c_1, \dots, c_4 are the sums of random variables which obey UNC constraints with respect to λ , and similarly for p_2^*, p_3^*, p_4^* .

Then there is a schedule of length $L' \leq (1+1/m)L + i$, which satisfies the 4-conditions with T, T' .

Furthermore, there is an algorithm to compute p_1^*, \dots, p_4^* . Although this is exponential-time in general, it is tractable for parameters in the range of interest.

Proof. Fix a frame F , and fix a packet and delay x, t within that frame. Let us examine the bad events that are affected by x, t . Once we adopt this delay for the given packet, all transits of the packet are determined. There are at most $mi/2$ positions in which the packet occurs as the first transit in a pair of aligned 2-frames, and so forth.

Let us fix such a pair of aligned 2-frames and an edge f , and we want to estimate the probability \mathbf{P}^* of the event that the 4-conditions fail, given that the packet is scheduled to transit in a given time step relative to the start of the 2-frames. To simplify the notation, we assume $\langle x, t \rangle$ is scheduled for the first time-step.

This bad event depends solely on the number of packets traversing the edge in the four relevant time-steps. Let Ω^* represent the worst-case probability distribution on the variables $Z_{x',t'}$ for other packets x' and delays t' which could affect these four time-steps. Each variable $Z_{x',t'}$ can cause the packet x' to be scheduled in exactly one of these four positions. We say a random variable $Z_{x',t'}$ has *type* j if it affects the j th position (for $j = 1, \dots, 4$). The total number of variables of each type is $K_j \leq C$.

We define random variables Y_j to be the sums of the random variable of type j . The bad event can now be written as

$$O^+(Y_1 + 1, Y_2) + O^-(Y_3, Y_4) > T';$$

here, we are adding +1 to Y_1 because the event $\langle x, t \rangle$ is forced to occur. These random variables Z (and consequently Y) are not independent, and Ω^* may represent a quite complicated joint distribution. However by Proposition 3.3 we may assume, without loss of generality, that Ω^* is *symmetric* in the sense that all atomic events have equal probability, if they lead to the same value of $\vec{Y} = \langle Y_1, \dots, Y_4 \rangle$. Now the entire probability distribution Ω^* is determined by the probability distribution on \vec{Y} .

We now rephrase the UNC-constraints in terms of \vec{Y} . If we fix a subset of the $Z_{x',t'}$ variables and compute the probability that all such variables are one simultaneously, then this depends solely on the number of $Z_{x',t'}$ variables of each type. Suppose we choose k_1, \dots, k_4 such variables. Then, given a fixed value for $\vec{Y} = \vec{y}$, the probability that all such variables

are set simultaneously is $\prod_{j=1}^4 \frac{\binom{K_j}{y_j - k_j}}{\binom{K_j}{y_j}}$. Hence, summing over values of y , we rephrase the UNC-constraint as: for all k_1, k_2, k_3, k_4 ,

$$\sum_{\vec{y}} \mathbf{P}(\vec{Y} = \vec{y}) \frac{\binom{K_1}{y_1 - k_1} \binom{K_2}{y_2 - k_2} \binom{K_3}{y_3 - k_3} \binom{K_4}{y_4 - k_4}}{\binom{K_1}{y_1} \binom{K_2}{y_2} \binom{K_3}{y_3} \binom{K_4}{y_4}} \leq \lambda^{k_1 + k_2 + k_3 + k_4}.$$

We relax this to the weaker constraint: for all k_1, k_2, k_3, k_4 ,

$$\sum_{\vec{y}} \mathbf{P}(\vec{Y} = \vec{y}) \frac{\binom{C}{y_1 - k_1} \binom{C}{y_2 - k_2} \binom{C}{y_3 - k_3} \binom{C}{y_4 - k_4}}{\binom{C}{y_1} \binom{C}{y_2} \binom{C}{y_3} \binom{C}{y_4}} \leq \lambda^{k_1 + k_2 + k_3 + k_4}.$$

By Proposition 3.2, we may assume that Ω^* is supported only at $Y = \langle 0, 0, 0, 0 \rangle$ and at minimal bad $\langle y_1, \dots, y_4 \rangle$, that is, values such that $\langle y_1, \dots, y_4 \rangle$ is a bad event but decreasing any coordinate is not a bad event. The probability of the bad event is the sum of all such probabilities, and all the UNC constraints are also positive linear constraints with respect to these probabilities. Hence, in order to determine the largest possible value of the bad event, we have an LP over a relatively tractable number of variables: it can be solved using standard LP libraries. \square

We now apply this construction to replace the two final steps in the construction of Section 6.3.

Theorem 6.9. *There is a feasible schedule of makespan at most $7.26(C + D)$.*

Proof. For simplicity, we assume $C + D \geq 2^{896}$.

By Lemma 6.3, we form a schedule S_1 , of length $L_1 \leq 1.004(C + D)$, in which each interval of length 2^{24} has congestion at most 17,040,600. (This step can be executed in polynomial time.)

We apply Proposition 6.7 with $\lambda = 5.984 \times 10^{-8}$, $C' = 1319$, and $m = 64$ to obtain a schedule S_2 of length $L_2 \leq 1.0157L_1 + 2^{24}$, in which each interval of length 1024 has congestion at most 1319. Next apply the algorithm of Proposition 6.8 with $\lambda = 0.001052$, $T = 6$, $T' = 6$. This gives the bounds

$$\begin{aligned} p_1^* &\leq 2.314 \times 10^{-7} \\ p_2^* &\leq 3.518 \times 10^{-7} \\ p_3^* &\leq 4.270 \times 10^{-7} \\ p_4^* &\leq 1.943 \times 10^{-7} \end{aligned}$$

and hence one can obtain a schedule S_3 of length $L_3 \leq 1.0158L_2 + 1024$ satisfying the 4-conditions with $T = 6$, $T' = 6$. By Proposition 6.5, this yields a schedule whose makespan is $7L_3 + 6 \leq 7.26(C + D)$. \square

We cannot construct this schedule in polynomial time using the assignment LLL of this paper. However, we are only using the assignment LLL for the final stages of this construction; we can use the ordinary algorithmic LLL for the earlier stages. Hence, one can show that there is an algorithm running in time $\exp(O(N))$ to find a schedule of length $7.26(C + D)$. This does not seem very efficient, but note that there is no obvious algorithm to solve the packet-routing problem optimally in time $\exp(O(N))$; rather, the exhaustive search would cost something like $\exp(\Omega(N \log N))$.

Acknowledgments. We thank Tom Leighton and Satish Rao for valuable discussions long ago, which served as the foundation for this work; but for their declining, they would be coauthors of this paper. We are thankful to Noga Alon, Venkatesan Guruswami, Bernhard Haeupler, Penny Haxell, and Jeff Kahn for helpful discussions, and to the STOC 2013 referees for their valuable comments.

Aravind Srinivasan dedicates this work to the memory of his late grandmother Mrs. V. Chellammal (a.k.a. Rajalakshmi): memories of her remain a great inspiration.

7. REFERENCES

- [1] N. Alon. The linear arboricity of graphs. *Israel Journal of Mathematics*, 62:311–325, 1988.
- [2] N. Alon and J. H. Spencer. *The Probabilistic Method, Third Edition*. John Wiley & Sons, Inc., 2008.
- [3] A. Archer. On the upper chromatic numbers of the reals. *Discrete Mathematics*, 214:65–75, 2000.
- [4] J. Beck and T. Fiala. “Integer-making” theorems. *Discrete Applied Mathematics*, 3:1–8, 1981.
- [5] R. B. Boppana and J. H. Spencer. A useful elementary correlation inequality. *Journal of Combinatorial Theory, Ser. A*, 50:305–307, 1989.
- [6] G. Brightwell and Y. Kohayakawa. Ramsey properties of orientations of graphs. *Random Structures and Algorithms*, 4:413–428, 1993.
- [7] A. Bulatov, P. Jeavons, and A. A. Krokhin. Constraint satisfaction problems and finite algebras. In *Proc. International Conference on Automata, Languages and Programming*, pages 272–282, 2002.
- [8] M. Cochand and P. Duchet. A few remarks on orientations of graphs and Ramsey theory. In G. Halász and V. T. Sós, editors, *Irregularities of Partitions, Algorithms and Combinatorics*, volume 8, pages 39–46. 1989.
- [9] P. Erdős and A. Gyárfás. Split and balanced colorings of complete graphs. *Discrete Mathematics*, 200:79–86, 1999.
- [10] P. Erdős and L. Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. In *Infinite and Finite Sets*, volume 11 of *Colloq. Math. Soc. J. Bolyai*, pages 609–627. North-Holland, 1975.
- [11] C. M. Fortuin, J. Ginibre, and P. N. Kasteleyn. Correlational inequalities for partially ordered sets. *Communications of Mathematical Physics*, 22:89–103, 1971.
- [12] H. Gebauer, T. Szabó, and G. Tardos. The Local Lemma is Tight for SAT. In *SODA*, pages 664–674, 2011.
- [13] D. Greenwell and P. Johnson Jr. Forbidding prescribed distances for designated colors. *Geombinatorics*, 2:13–16, 1992.
- [14] B. Haeupler, B. Saha, and A. Srinivasan. New Constructive Aspects of the Lovász Local Lemma. *Journal of the ACM*, 58, 2011.
- [15] P. E. Haxell. A note on vertex list colouring. *Combinatorics, Probability, and Computing*, 10:345–348, 2001.
- [16] P. E. Haxell, T. Szabó, and G. Tardos. Bounded size components – partitions and transversals. *Journal of Combinatorial Theory, Series B*, 88:281–297, 2003.
- [17] R. M. Karp, F. T. Leighton, R. L. Rivest, C. D. Thompson, U. V. Vazirani, and V. V. Vazirani. Global wire routing in two-dimensional arrays. *Algorithmica*, 2:113–129, 1987.
- [18] K. Kolipaka and M. Szegedy. Moser and Tardos meet Lovász. In *Proceedings of ACM STOC*, pages 235–244, 2011.
- [19] F. T. Leighton, C.-J. Lu, S. B. Rao, and A. Srinivasan. New Algorithmic Aspects of the Local Lemma with Applications to Routing and Partitioning. *SIAM Journal on Computing*, 31:626–641, 2001.

- [20] F. T. Leighton, B. M. Maggs, and S. B. Rao. Packet routing and jobshop scheduling in $O(\text{congestion} + \text{dilation})$ steps. *Combinatorica*, 14:167–186, 1994.
- [21] M. Molloy and B. Reed. *Graph Colouring and the Probabilistic Method*. Springer-Verlag, 2001.
- [22] R. Moser and G. Tardos. A constructive proof of the general Lovász Local Lemma. *Journal of the ACM*, 57(2):1–15, 2010.
- [23] A. Panconesi and A. Srinivasan. Randomized Distributed Edge Coloring via an Extension of the Chernoff-Hoeffding Bounds. *SIAM Journal of Computing*, 26(2):350–368, 1997.
- [24] W. Pegden. An extension of the Moser-Tardos algorithmic Local Lemma. *Arxiv 1102.2583*, 2011.
- [25] B. Peis and A. Wiese. Universal packet routing with arbitrary bandwidths and transit times. In *IPCO*, pages 362–375, 2011.
- [26] V. Rödl. A generalization of Ramsey theorem. In *Graphs, Hypergraphs, and Block Systems*, pages 211–220. Zielona Gora, 1976.
- [27] V. Rödl. On a packing and covering problem. *European Journal of Combinatorics*, 6:69–78, 1985.
- [28] T. Rothvoss. A simpler proof for $O(\text{congestion} + \text{dilation})$ packet routing. In *Proc. Conference on Integer Programming and Combinatorial Optimization*, 2013. URL: <http://arxiv.org/pdf/1206.3718.pdf>.
- [29] C. Scheideler. Universal routing strategies for interconnection networks. In *Lecture Notes in Computer Science*, volume 1390. Springer, 1998.
- [30] J. Shearer. On a problem of Spencer. *Combinatorica*, 5:241–245, 1985.
- [31] A. Srinivasan. An extension of the Lovász Local Lemma, and its applications to integer programming. *SIAM Journal on Computing*, 36:609–634, 2006.