Balanced Allocation on Dynamic Hypergraphs*

Catherine Greenhill† Bernard Mans‡ Ali Pourmiri§

April 24, 2020

Abstract

The balls-into-bins model randomly allocates n sequential balls into n bins, as follows: each ball selects a set D of $d \geqslant 2$ bins, independently and uniformly at random, then the ball is allocated to a least-loaded bin from D (ties broken randomly). The *maximum load* is the maximum number of balls in any bin. In 1999, Azar et al. showed that, provided ties are broken randomly, after n balls have been placed the *maximum load*, is $\log_d \log n + \mathcal{O}(1)$, with high probability. We consider this popular paradigm in a dynamic environment where the bins are structured as a *dynamic hypergraph*. A dynamic hypergraph is a sequence of hypergraphs, say $\mathcal{H}^{(t)}$, arriving over discrete times $t = 1, 2, \ldots$, such that the vertex set of $\mathcal{H}^{(t)}$'s is the set of n bins, but (hyper)edges may change over time. In our model, the t-th ball chooses an edge from $\mathcal{H}^{(t)}$ uniformly at random, and then chooses a set D of $d \geqslant 2$ random bins from the selected edge. The ball is allocated to a least-loaded bin from D, with ties broken randomly. We quantify the dynamicity of the model by introducing the notion of *pair visibility*, which measures the number of rounds in which a pair of bins appears within a (hyper)edge. We prove that if, for some $\varepsilon > 0$, a dynamic hypergraph has pair visibility at most $n^{1-\varepsilon}$, and some mild additional conditions hold, then with high probability the process has maximum load $\mathcal{O}(\log_d \log n)$. Our proof is based on a variation of the witness tree technique, which is of independent interest. The model can also be seen as an adversarial model where an adversary decides the structure of the possible sets of d bins available to each ball.

1 Introduction

The standard balls-into-bins model is a process that randomly allocates m sequential balls into n bins, where each ball chooses a set D of d bins, independently and uniformly at random, then the ball is allocated to a least-loaded bin from D (with ties broken randomly). When $m = n$ and $d = 1$, it is well known that at the end of process the maximum number of balls at any bin, the *maximum load*, is $(1 + o(1))\frac{\log n}{\log \log n}$, with high probability. Surprisingly, Azar et al. [2] showed that for this d-choice process with $d \geqslant 2$, provided ties are broken randomly, the maximum load exponentially decreases to $\log_d \log n + \mathcal{O}(1)$. This phenomenon is known as the *power of d choices*. The multiple-choice paradigm has been successfully applied in a wide range of problems from nearby server selection, and load-balanced file placement in the distributed hash table, to the performance analysis of dictionary data structures (e.g., see [21]). In the classical setting, all $\binom{n}{d}$ sets of d bins are available to each ball. However, in many realistic scenarios such as cache networks, peer-to-peer or cloud-based systems, the balls (requested files, jobs, items,..) have to be allocated to bins

*The first author is supported by the Australian Research Council Discovery Project DP190100977. The second and third authors are supported by the Australian Research Council Discovery Project DP170102794.

†UNSW Sydney, Australia, c.greenhill@unsw.edu.au
‡Macquarie University, Sydney, Australia, bernard.mans@mq.edu.au
§Macquarie University, Sydney, Australia, ali.pourmiri@mq.edu.au

(servers, processors,...) that are close to them, in order to minimize the access latencies. On the other hand, the lack of perfect randomness stimulates the de-randomization of the d-choice process, which also requires the study of non-uniform distributions over choices (e.g. [1, 6, 7, 11]). Hence in many settings, allowing all possibilities for the set D of d bins is costly, and may not be practical. This motivates the investigation of the effect of distributions of the set D on the maximum load. In this regard, Kenthapadi and Panigrahy [13] proposed *balanced allocation* on graphs, where bins form the vertices of a Δ-regular graph and each ball chooses an edge of the graph uniformly at random. The ball is then placed in an endpoint of the selected edge with smaller load (ties are broken randomly). Kenthapadi and Panigrahy showed that the maximum load is $\Theta(\log\log n)$ if and only if $\Delta = n^{\Omega(1/\log\log n)}$. Here, one may see that the possibilities for the set D (the two chosen bins) is restricted to the set of $n\Delta/2$ edges of the graph. In the standard balls-into-bins model with $d = 2$, the underlying graph is a complete graph (all $\binom{n}{2}$ edges present). Following the study of balls-into-bins with related choices, Godfrey [12] utilized hypergraphs to model the structure of bins. In this model, each ball picks a random edge of a given hypergraph that contain $\Omega(\log n)$ bins and the hypergraph satisfies some mild conditions. Then, the ball is allocated to a least-loaded bin contained in the edge, with ties broken randomly. Godfrey showed that the maximum load is constant. Balanced allocation on graphs and hypergraphs has been further studied in [3, 4, 18, 19]. In the aforementioned works, either the underlying graph is fixed during the process or, in the hypergraph setting, the number d of choices satisfies $d = \Omega(\log n)$. However, in many real-world systems the structure may change over time, and probing the load of $\Omega(\log n)$ bins might be a costly task. Seeking a more realistic model, this paper studies the d-choice process in dynamic graphs and hypergraphs, where $2 \leqslant d = o(\log n)$.

Balanced allocation on dynamic hypergraphs can also be seen as an adversarial model, where the set D of potential choices is proposed by an adversary (environment) whose goal is to increase the maximum load. Here we want to understand the conditions under which the balanced allocation on dynamic (hyper)graphs still benefits from the effect of the power of d choices.

1.1 Our Results

We propose balanced allocation algorithms on different dynamic environments, namely dynamic graph and hypergraph models. In order to measure the dynamicity, we introduce the notion of *pair visibility*. For a pair $\{i, j\}$ of distinct vertices, the *visibility* of $\{i, j\}$, denoted by $\mathtt{vis}(i, j)$, is the number of rounds $t \in \{1, \dots, n\}$ such $\{i, j\}$ is contained in the edge chosen at round t. (A more formal definition is given below.) When ball i is placed into a bin, the *height* of ball i is the number of balls that were allocated to the bin before ball i. We say that event E_n holds with high probability (w.h.p.) if $\mathbf{Pr}[\mathsf{E}_n] \geqslant 1 - n^{-c}$ for every constant $c > 0$.

Balanced Allocation on Dynamic Hypergraphs

Write $[n] = \{1, \dots, n\}$ to be the set of n bins. A hypergraph $\mathcal{H} = ([n], \mathcal{E})$ is *s-uniform* if $|H| = s$ for every $H \in \mathcal{E}$. For every integer $n \geqslant 1$, let $s = s(n)$ be an integer such $2 \leqslant s \leqslant n$. A *dynamic s-uniform hypergraph*, denoted by $(\mathcal{H}^{(1)}, \mathcal{H}^{(2)}, \dots, \mathcal{H}^{(n)})$, is a sequence of s-uniform hypergraphs $\mathcal{H}^{(t)} = ([n], \mathcal{E}_t)$ with vertex set $[n]$. The edge sets $\mathcal{E}_t$ may change with t. A hypergraph is *regular* if every vertex is contained in the same number of edges.

In this paper, we are interested in the following properties which dynamic hypergraphs may satisfy. We refer to these properties as the *balancedness*, *visibility*, and *size* properties. The balancedness property is adapted from [3, 12].

> Balancedness: Let H_t denote a randomly chosen edge from $\mathcal{E}_t$. If there exists a constant $\beta \geqslant 1$ such that $\mathbf{Pr}[i \in H_t] \leqslant \beta s/n$ for every $1 \leqslant t \leqslant n$ and each bin $i \in [n]$, then the

dynamic hypergraph $(\mathcal{H}^{(1)}, \dots \mathcal{H}^{(n)})$ is β-balanced. A dynamic hypergraph is balanced if it is β-balanced for some constant $\beta \geqslant 1$. Every regular hypergraph is 1-balanced.

Visibility: For every pair of distinct vertices $\{i, j\} \subset [n]$, the visibility of $\{i, j\}$ is

$$\mathtt{vis}(i, j) = |\{t \in \{1, 2, \dots, n\} \mid \{i, j\} \subset H \in \mathcal{E}_t\}|.$$

If there exists a constant $\varepsilon \in (0, 1)$ such that $\mathtt{vis}(i, j) \leqslant sn^{1-\varepsilon}$ for all pairs $\{i, j\} \subseteq [n]$ of distinct bins then the dynamic hypergraph $(\mathcal{H}^{(1)}, \dots, \mathcal{H}^{(n)})$ is ε-visible. A dynamic hypergraph satisfies the visibility property if it is ε-visible for some constant $\varepsilon \in (0, 1)$.

Size: If $s = \Omega(\log n)$ and there exists a positive constant $c_0 \geqslant 1$ such that $|\mathcal{E}_t| \leqslant n^{c_0}$ for every $t \geqslant 1$, then the dynamic hypergraph $(\mathcal{H}^{(1)}, \dots, \mathcal{H}^{(n)})$ satisfies the c_0-size property. A dynamic hypergraph satisfies the size property if it satisfies the c_0-size property for some constant $c_0 \geqslant 1$.

Definition 1 (Balanced Allocation on Dynamic Hypergraphs)**.** Suppose that $(\mathcal{H}^{(1)}, \dots, \mathcal{H}^{(n)})$ is an s-uniform hypergraph and fix $d = d(n)$ with $2 \leqslant d = o(\log n)$ and $d \leqslant s$. The balanced allocation algorithm on $(\mathcal{H}^{(1)}, \dots, \mathcal{H}^{(n)})$ proceeds in rounds $(t = 1, 2, \dots, n)$, sequentially allocating n balls to n bins. In round t, the t-th ball chooses an edge H_t uniformly at random from $\mathcal{E}_t$, then it randomly chooses a set D_t of d bins from H_t (without repetition) and allocates itself to a least-loaded bin from D_t, with ties broken randomly.

Theorem 2. *Let $(\mathcal{H}^{(1)}, \dots, \mathcal{H}^{(n)})$ be a dynamic s-uniform hypergraph which satisfies the balancedness, ε-visibility and size properties. Fix $d = d(n)$ such that $2 \leqslant d = o(\log n)$. There exists $\Theta(n) \leqslant m \leqslant n$ such that after the balanced allocation process on $(\mathcal{H}^{(1)}, \dots, \mathcal{H}^{(n)})$ has allocated m balls, the maximum load is $\log_d \log n + \mathcal{O}(1/\varepsilon)$ with high probability. Moreover, for every fixed positive integer γ with $\gamma m \leqslant n$, after allocating γm balls the maximum load is at most $\gamma(\log_d \log n + \mathcal{O}(1/\varepsilon))$, w.h.p..*

Remark 3. In our result we only consider the case where $d = o(\log n)$, because when $d = \Omega(\log n)$, a constant upper bound is obtained by [12]. The size property is mainly assumed for technical reasons. For instance, $|\mathcal{E}_t| \leqslant \text{poly}(n)$ is not necessary. Roughly speaking, balanced allocation on a dynamic hypergraph with large $|\mathcal{E}_t|$ resembles the standard balls-into-bins process. So it might be possible that having more structural information about a dynamic hypergraph would enable us to extend our result to allow an arbitrary number of edges $|\mathcal{E}_t|$. Another possible extension of Theorem 2 would be to allow s to be a function of d.

The proof of Theorem 2 is based on the witness tree technique (see [1, 11, 16, 20] for example). First, we define a certain structure corresponding to the allocation process and claim that the structure exists with very small probability (i.e., $n^{-\mathcal{O}(1)}$). Second, we will deterministically show that if the maximum load is higher than a certain threshold, then this structure must exist. Putting these together, we obtain an upper bound for the maximum load, with high probability. This approach is of independent interest and might be applied for the study of random hypergraphs. The proof is given in Section 2.

Finally, in the following theorem we show that ε-visibility can also lead to a lower bound on the maximum load achieved by the balanced allocation process on hypergraphs. This theorem is proved in Appendix A.

Theorem 4. *Let $s = s(n) = n^{\varepsilon}$, where $\varepsilon \in (0, 1)$ is an arbitrary small real number. There exists a dynamic s-uniform hypergraph, say $(\mathcal{H}^{(1)}, \dots, \mathcal{H}^{(n)})$, which satisfies the balancedness condition and (trivially) satisfies the ε-visibility condition. Let $2 \leqslant d \leqslant s$ be any integer which is constant. Suppose that the balanced allocation process on $(\mathcal{H}^{(1)}, \dots, \mathcal{H}^{(n)})$ has allocated n balls, then the maximum load is at least $\min\{\Omega(1/\varepsilon), \Omega(\log n/ \log \log n)\}$ with high probability.*

Balanced Allocation on Dynamic Graphs

A dynamic graph is a special case of a dynamic hypergraph, where $s = s(n) = 2$ for all n. Write $(G^{(1)}, \ldots, G^{(n)})$ to denote a dynamic graph, where $G^{(t)} = ([n], E_t)$ for $t = 1, 2, \ldots, n$. Theorem 2 does not cover the case of graphs ($s = 2$), due to the size property. We will prove a result on balanced allocation for regular dynamic graphs.

Definition 5 (Balanced Allocation on Dynamic Graphs)**.** Suppose that $(G^{(1)}, \ldots, G^{(n)})$ is a regular dynamic graph on vertex set $[n]$. The balanced allocation algorithm on $(G^{(1)}, \ldots, G^{(n)})$ proceeds in rounds ($t = 1, \ldots, n$). In each round t, the t-th ball chooses an edge of $G^{(t)}$ uniformly at random, and the ball is then placed in one of the bins incident to the edge with a lesser load, with ties broken randomly.

Say that the dynamic graph is *regular* if $G^{(t)}$ is Δ_t-regular for some positive integer Δ_t and all $t = 1, 2, \ldots, n$. For every pair of distinct bins $\{i, j\} \subset [n]$, we will assume that the visibility $\mathtt{vis}(i, j)$ satisfies

$$\mathtt{vis}(i, j) = |\{t \in \{1, 2, \ldots, n\} \mid \{i, j\} \in E_t\}| \leqslant 2n^{1-\varepsilon}$$

for some constant $\varepsilon \in (0, 1)$. This property is called ε-visibility.

Theorem 6. *Let $(G^{(1)}, \ldots, G^{(n)})$ be a regular dynamic graph which satisfies the ε-visibility condition, for some $\varepsilon \in (0, 1)$. Suppose that the balanced allocation process on $(G^{(1)}, \ldots, G^{(n)})$ has allocated n balls. Then the maximum load is at most $\log_2 \log n + \mathcal{O}(1/\varepsilon)$, with high probability.*

The proof, which can be found in Section 3, is again based on the witness tree technique. We remark that Theorem 6 can be extended to the case where the dynamic graph is *almost regular*, meaning that the ratio of the minimum and maximum degree of $G^{(t)}$ is bounded above by an absolute constant for $t = 1, \ldots, n$.

Dynamic Graphs and Hypergraphs with Low Pair Visibility

In order to show the ubiquity of the visibility condition, we will describe some dynamic graphs with low pair visibility. One can easily construct a dynamic hypergraph from a dynamic graph by considering the r-neighborhood of each vertex of the t-th graph as a hyperedge in the t-th hypergraph, for $t = 1, \ldots, n$.

- Dynamic Cycle. For $t = 1, \ldots, n$ define the edge set

  $$E_t = \{\{i, j\} \subset \{0, \ldots, n-1\} \mid j = i + \lceil t/\sqrt{n} \rceil \pmod{n} \text{ or } i = j + \lceil t/\sqrt{n} \rceil \pmod{n}\},$$

  where calculations are performed modulo n (that is, in the additive group $\mathbb{Z}_n$). In modular addition, for every pair $\{i, j\} \subset \{0, \ldots, n-1\}$, the equation $i = j + k \pmod{n}$ has at most one solution $1 \leqslant k \leqslant \sqrt{n}$ and hence

  $$\mathtt{vis}(i, j) = |\{t \in \{1, 2, \ldots, n\} \mid \{i, j\} \in E_t\}| \leqslant \sqrt{n}.$$

  Now $C^{(t)} = (\{0, 1, \ldots, n-1\}, E_t)$ is 2-regular, so it is either a Hamilton cycle or a union of two or more disjoint cycles (depending on whether t and n are coprime). By Theorem 6, the maximum load attained by the algorithm on $\{C^{(t)}, t = 1, \ldots, n\}$ is at most $\log_2 \log n + \mathcal{O}(1)$. The analysis of the balanced allocation algorithm on Δ-regular graphs given by Kenthapadi and Panigrahy [13] showed that the balanced allocation process on arbitrary Δ-regular graphs has maximum load $\Theta(\log \log n)$ only when $\Delta = n^{\Omega(1/\log \log n)}$. By contrast, here each $C^{(t)}$ has degree at most 2, but the visibility condition keeps the maximum load as low as the standard two-choice process.

Remark 7. By Theorem 6, w.h.p., the balanced allocation process on the dynamic cycle achieves the maximum load at most $\log_2 \log n + \mathcal{O}(1)$. Since $|E_t| = n$ for $t = 1, \ldots, n$, each ball requires $\log_2 n$ random bits. However, in the standard power-of-two-choices process, each ball chooses two independent and random bins, which requires $2 \log n$ random bits. Therefore, the dynamic cycle can be used to reduce (by half) the number of random bits required in the standard two-choice process.

- Dynamic Modular Hypergraph. Suppose that n is a prime number and fix $s = s(n)$ such that $\log n \leqslant s \leqslant n^{1/5}$. (Here n is large enough so that this range is non-empty.) For $t = 1, \ldots, n$, let $k_t = \lceil \sqrt{n} \rceil + \lceil \frac{t}{n^{3/4}} \rceil$ and for each $\alpha \in \mathbb{Z}_n$ define

$$H_t(\alpha) = \{\, \alpha + j k_t \,(\mathrm{mod}\ n) \mid j = 0, 1, \ldots, s-1 \,\}.$$

Then $H_t(\alpha)$ is a subset of $\mathbb{Z}_n$ of size s, as n is prime. Now for each $t = 1, \ldots, n$ we define the dynamic s-uniform hypergraph $\mathcal{H}^{(t)} = (\mathbb{Z}_n, \mathcal{E}_t)$, where $\mathcal{E}_t = \{H_t(\alpha) \mid \alpha \in \mathbb{Z}_n\}$. Then $\mathcal{H}^{(t)}$ is s-regular, and hence 1-balanced, and it satisfies the 1-size property as $|\mathcal{E}_t| = n$. Suppose that $\{\beta_1, \beta_2\} \subset H_t(\alpha)$ for some $\alpha \in \mathbb{Z}_n$, with $\beta_1 \neq \beta_2$. Then there exists $j_1, j_2 \in \{0, \ldots, s-1\}$ such that $\beta_1 = \alpha + j_1 k_t \,(\mathrm{mod}\ n)$ and $\beta_2 = \alpha + j_2 k_t \,(\mathrm{mod}\ n)$. Thus, $\beta_2 - \beta_1 = (j_2 - j_1) k_t \,(\mathrm{mod}\ n)$. Note that j_1, j_2 must be distinct as β_1, β_2 are distinct. Next suppose that $k_{t_1} \neq k_{t_2}$ for some $t_1, t_2 \in \{1, \ldots, n\}$, and take any $j_1, j_2 \in \{1, \ldots, s-1\}$. By definition of k_t and working in $\mathbb{Z}$, we see that

$$1 \leqslant |j_2 k_{t_2} - j_1 k_{t_1}| \leqslant (s-1)\big(\lceil \sqrt{n} \rceil + \lceil n^{1/4} \rceil\big) < n,$$

and it follows that

$$j_1 k_{t_1} \neq j_2 k_{t_2} \,(\mathrm{mod}\ n). \tag{1}$$

Finally, suppose that some distinct β_1, β_2 satisfy $\{\beta_1, \beta_2\} \subset H_{t_1}(\alpha) \cap H_{t_2}(\alpha)$ where $k_{t_1} \neq k_{t_2}$. Then $\beta_2 - \beta_1 = j k_{t_1} \,(\mathrm{mod}\ n)$ for some $j_1 \in \{1, \ldots, s-1\}$, and $\beta_2 - \beta_1 = j_2 k_{t_2} \,(\mathrm{mod}\ n)$ for some $j_2 \in \{1, \ldots, s-1\}$, but this contradicts (1). Therefore, by definition of k_t, for every $\{\beta_1, \beta_2\} \subset \mathbb{Z}_n$, we have

$$\mathtt{vis}(\beta_1, \beta_2) = |\{t \in \{1, 2, \ldots, n\} \mid \{\beta_1, \beta_2\} \subset H_t(\alpha) \text{ for some } \alpha \in \mathbb{Z}_n\}| \leqslant \mathcal{O}(n^{3/4}).$$

- Stationary Geometric Mobile Network. Consider an R-dimensional torus $\Gamma(n, R)$, which is a graph whose vertex set is the Cartesian product of $\mathbb{Z}_\ell^R = \mathbb{Z}_\ell \times \ldots \times \mathbb{Z}_\ell$, where $\ell = n^{1/R} \in \mathbb{Z}$, and two vertices $(x_1, \ldots, x_R)$ and $(y_1, \ldots, y_R)$ are connected if for some $j \in \{1, 2 \ldots, R\}$ $x_j = y_j \pm 1 \bmod n$ and for all $i \neq j$ we have $x_i = y_i$. Let π be the stationary distribution of the following random walk on $\Gamma(n, R)$: at each step, the walker stays at the current vertex with probability p, and otherwise chooses a neighbour randomly and moves to that neighbour. The transition probability from vertex u to a neighbouring vertex w is $(1-p)/(2R)$, where $2R$ is the degree of vertex u in $\Gamma(n, R)$. Now place n agents on vertices of $\Gamma(n, R)$ independently, each according to the distribution π. At each time step, each agent independently performs a step of the random walk described above (For random walks on a torus we refer the interested reader to [15]). For every pair of distinct agents a and b, let $d_t(a, b)$ denote the Manhattan distance (in Γ) of the locations of a and b at time t. For a given $r \geqslant 1$, we define the *communication graph process* $\{G_r^{(t)} \mid t = 0, 1, \ldots\}$ over the set of agents, say A, so that for every $t \geqslant 0$, agents a and b are connected if and only if $d_t(a, b) \leqslant r$. The model has been thoroughly studied when $R = 2$ in the context of information spreading [9]. We present the following result regarding the pair visibility of the communication graph process, proved in Appendix B.

Proposition 8. *Fix $r = r(n) = n^{o(1)}$. Also let $\{G_r^{(t)} = (A, E_t) \mid 1 \leqslant t \leqslant n\}$ be the communication graph process defined on an R-dimensional torus $\Gamma(n, R)$. Then there exists constant $\varepsilon > 0$ such that for every pair of agents, say $\{a, b\} \subset A$,*

$$\texttt{vis}(a, b) = |\{t \in \{1, 2, \ldots, n\} \mid \{a, b\} \in E_t\}| = \mathcal{O}(n^{1-\varepsilon}).$$

1.2 Related Works

As we discussed, in the standard balls-into-bins, each ball picks a set of d choices from n bins, independently and uniformly at random. One of the first algorithms considering a different distribution over the bins is called *always-go-left* proposed by Vöcking [20]. In this algorithm, the bins are partitioned into d groups of size n/d and each ball picks one random bin from each group. The ball is then allocated to a least-loaded bin among the chosen bins, with ties broken in favor of the bin from the least-indexed group. The algorithm uses exponentially smaller number of choices and achieve a maximum load of $\frac{\log \log n}{d\phi_d} + \mathcal{O}(1)$, where $1 \leqslant \phi_d \leqslant 2$ is an specified constant. Byers et al. [5] studied a model, where n bins are uniformly at random placed on a geometric space. Then each ball, in turn, picks d locations in the space. Corresponding to these d locations, the ball probes the load of d bins that have the minimum distance from the locations. The ball then allocates itself to one of the d bins with minimum load. In this scenario, the probability that a location close to a bin is chosen depends on the distribution of other bins in the space and hence there is not a uniform distribution over the potential choices. Here, the authors showed the maximum load is $\log_d \log n + \mathcal{O}(1)$. Later on, Kenthapadi and Panigrahy [13] proposed a graphical balanced allocation in which bins are interconnected as a s-regular graph and each ball picks a random edge of the graph. It is then placed in one of its endpoints with a smaller load. This allocation algorithm results in a maximum load of $\log \log n + \mathcal{O}\left(\frac{\log n}{\log(s/\log^4 n)}\right) + \mathcal{O}(1)$. Godfrey [12] studied balanced allocation on hypergraphs where each ball probes the bins contained in a random edge of size $\Omega(\log n)$. In [3, 12], the balanced allocation process on hypergraphs was studied where number of choices is $d = \Omega(\log n)$. The analysis involves the second moment method (Chernoff bounds), and lower bound on d is needed in order to achieve concentration. Hence it is unlikely that the techniques of [3,12] can be extended to the range $d = o(\log n)$. Peres et al. [18] also considered balanced allocation on graphs where the number of balls m can be much larger than n (i.e., $m \gg n$) and the graph is not necessarily regular and dense. Then, they established upper bound $\mathcal{O}(\log n/\sigma)$ for the gap between the maximum and the minimum loaded bin after allocating m balls, where σ is the edge expansion of the graph. Bogdan et al. [4] studied a model where each ball picks a random vertex and performs a local search from the vertex to find a vertex with local minimum load, where it is finally placed. They showed that when the graph is a constant degree expander, the local search guarantees a maximum load of $\Theta(\log \log n)$. Pourmiri [19] substitutes the local search by non-backtracking random walks of length $\ell = o(\log n)$ to sample the choices and then the ball is allocated to a least-loaded bin. Provided the underlying graph has sufficiently large girth and ℓ, he showed the maximum load is a constant. In the context of hashing (e.g., [1, 11]), authors apply the witness graph techniques to analyze the maximum load in the balls-into-bins process where the bins are picked based on tabulation.

2 Balanced Allocation on Dynamic Hypergraphs

In this section we establish an upper bound for the maximum load attained by the balanced allocation on hypergraphs (i.e., Theorem 2). In order to analyze the process let us first define a *conflict graph*. We write D_t for the set of d bins chosen by the t-th ball, and sometimes refer to D_t as the *d-choice* of the t-th ball. We will slightly abuse the notation and write $D_u \cap D_t$, $D_u \cup D_t$ to denote the set of common bins, and the union of bins, chosen by balls u and t, respectively.

Definition 9 (Conflict Graph)**.** For $m = 1, \ldots, n$, the conflict graph $\mathcal{C}_m$ is a simple graph with vertex set $\{D_1, D_2, \ldots, D_m\}$. Vertices D_u and D_t are connected by an edge in $\mathcal{C}_m$ if and only if $D_u \cap D_t \neq \emptyset$ (that is, the d-choices of the t-th ball and the u-th ball contain a common bin).

We say a subgraph of $\mathcal{C}_m$ with vertex set $\{D_{t_1}, \ldots, D_{t_k}\}$ is *c-loaded* if every bin in $D_{t_1} \cup D_{t_2} \cup \cdots \cup D_{t_k}$ has at least c balls.

Our analysis will involve a useful combinatorial object, called an *ordered tree*. An ordered tree is a rooted tree, together with a specified ordering of the children of every vertex. Recall that $\frac{1}{k+1}\binom{2k}{k}$ is the k-th Catalan number, which counts numerous combinatorial objects, including the number of ways to form k balanced parentheses. It is well known [17] that ordered trees with $k-1$ edges are counted by the $(k-1)$-th Catalan number, leading easily to the following proposition.

Proposition 10. *The number of k-vertex ordered trees is $\frac{1}{k}\binom{2k-2}{k-1} \leqslant 4^{k-1}$.*

More information regarding the enumeration of trees can be found in [14].

The following blue-red coloring will be very helpful in our analysis.

Definition 11 (Blue-red coloring)**.** Given $m \in \{1, 2, \ldots, n\}$, suppose that $T \subset \mathcal{C}_m$ is a rooted and ordered k-vertex tree contained in $\mathcal{C}_m$. Let the vertex set of T be $\{D_{t_1} \ldots, D_{t_k}\}$, where D_{t_1} is the root. Perform depth-first search starting from the root, respecting the specified order of each vertex. For $i = 1, \ldots, k$, let $D(i) \in \{D_{t_1} \ldots, D_{t_k}\}$ be the vertex which is the i-th visited vertex in the depth-first search. Then $D(1) = D_{t_1}$ is the root. for $j = 1, \ldots, k$. We now define a blue-red coloring $col : \{D(2), \ldots, D(k)\} \to \{\text{blue, red}\}$ as follows. For $i = 2, \ldots, k$,

$$col(D(i)) = \begin{cases} \text{blue} & \text{if } \left|(\cup_{j=1}^{i-1} D(j)) \cap D(i)\right| = 1, \\ \text{red} & \text{if } \left|(\cup_{j=1}^{i-1} D(j)) \cap D(i)\right| \geqslant 2. \end{cases}$$

The following key lemma presents a upper bound for the probability that a certain tree can be found as a subgraph of $\mathcal{C}_m$.

Lemma 12 (Key Lemma)**.** *Let $(\mathcal{H}^{(1)}, \ldots, \mathcal{H}^{(n)})$ be a dynamic s-uniform hypergraph which satisfies the β-balanced, ε-visibility and c_0-size properties. Suppose that $c > 0$ is an arbitrary constant and $k = C \log n$ for some constant $C \geqslant 1$. There exists $\Theta(n) \leqslant m \leqslant n$ such that the probability that $\mathcal{C}_m$ contains a c-loaded k-vertex tree with r red vertices in its blue-red colouring is at most*

$$n^{c_0+3} \exp\{4k \log(2\beta d) - r\varepsilon \log(n)/2 - c(d-1)(k-r-1)\}.$$

Moreover, with high probability, $r = \mathcal{O}(1/\varepsilon)$.

The proof, presented in Appendix C, involves an extension of the witness tree technique. This method might be of independent interest in the study of random hypergraphs.

We now explain how to recursively build a witness graph if there exists a bin whose load is higher than a certain threshold. The *minimum load* of D_t is the number of balls in the least-loaded bin in D_t (the set of d choices of D_t). Clearly, if ball t is placed at height h then D_t has minimum load at least h.

Construction of the Witness Graph Suppose that there exists a bin with load $\ell + c + 1$. Let R be the d-choice corresponding to the ball at height $\ell + c$ in this bin. Then the minimum load of R is $\ell + c$. We start building the witness tree in $\mathcal{C}_m$ whose root is R. For every bin $i \in R$, consider the ℓ balls in bin i at height $\ell + c - j$, for $j = 1, \ldots, \ell$, and let $D^i_{t_j}$ be the d-choice corresponding to the ball in bin i with height $\ell + c - j$. These ℓ balls exist as the minimum load of R is $\ell + c$. We refer to set $\{D^i_{t_j} \mid i \in R,\ 1 \leqslant j \leqslant \ell\}$ as the set of *children* of R, where the minimum load of $D^i_{t_j}$

is $\ell + c - j - 1$. All children of R are connected to R in $\mathcal{C}_m$. Order the children of R arbitrarily, then blue-red colour the first level of the tree (the children of R). Recall that a vertex is colored by blue if it only shares one bin with its predecessors in the ordering. So a blue d-choice contains $d-1$ bins that have not appeared in previous d-choices (with respect to depth-first search, respecting the fixed ordering). We call these $d-1$ bins *fresh*.

Next, consider each blue vertex of the tree (if any), and recover the d-choices corresponding to balls that are placed in fresh bins with height at least c. Then, blue-red color the children of those d-choices, with respect to an arbitrary ordering. This recursion will continue until either there are no balls remaining with height at least c, or there are no blue vertices. For $j = 1, \dots, \ell$, let $f(\ell - j)$ denote the number of d-choices that the recursive construction gives, when the d-choice for the root has minimum load $\ell + c - j - 1$. Provided all vertices are colored blue, the recursive construction continues until no ball remains with height at least c. Therefore, a simple calculation shows that

$$f(\ell) \geqslant (d-1)(f(\ell-1) + f(\ell-2) + \cdots + f(0) + 1),$$

where $f(0) = 1$. Solving the above recursive formula shows that $f(\ell) \geqslant 2(d-1)d^{\ell-1} \geqslant d^\ell$.

Proof of Theorem 2. Let $(\mathcal{H}^{(1)}, \dots, \mathcal{H}^{(n)})$ be a dynamic hypergraph which satisfies the β-balanced, ε-visibility and c_0-size properties. By Lemma 12, there exists $\Theta(n) = m \leqslant n$ such that the following holds with high probability: after m balls have been allocated by the balanced allocation process, if $T \subseteq \mathcal{C}_m$ is a c-loaded tree with k vertices and T is blue-red coloured according to some arbitrary ordering of the children of each vertex, then the number r of red vertices satisfies $r = \mathcal{O}(1/\varepsilon)$. So we are able to find a constant $c_2 \geqslant 0$ such that, with high probability, $r < c_2 \cdot d$.

Now suppose that after allocating m balls, there is a ball at height $\ell + c_1 + c_2 + 1$. This implies that there is a d-choice, denoted by R, whose minimum load is at least $\ell + c_1 + c_2 + 1$. Let us consider all balls placed in the bins contained in R with height at least $\ell + c_1 + 1$. Recover the corresponding d-choices for these balls, say $D_1, D_2, \dots, D_w$, then colour them blue-red with respect to the root R and an arbitrary ordering of the children of each vertex. Since $w \geqslant c_2 \cdot d$, w.h.p., there are $b \geqslant 1$ blue vertices and $w - b$ red vertices. We now consider every blue vertex $D_t \in \{D_1, D_2, \dots, D_w\}$ as a root and start the recursive construction of the witness graph. Assuming that the number of red vertices is strictly less than $c_2 \cdot d < w$, it follows that at least one recursive construction (with root D_i) does not produce any red vertex. Moreover, the recursion from D_i gives a c_1-loaded tree with at least $k = d^\ell$ vertices. We take $\ell = \log_d \log n$, so that $k = \log n$. Another application of Lemma 12 implies that a c_1-loaded k-vertex tree with no red vertices exists with probability at most

$$\begin{aligned} n^{c_0+3} \exp\{4k \log(2\beta d) - c_1(d-1)(k-1)\} &\leqslant \exp\left\{\left(c_0 + 4 + 4\log(2\beta d) - c_1(d-1)\right)\log n\right\} \\ &\leqslant \exp\left\{\left(c_0 + 4 + 4\log(4\beta) - c_1\right)\log n\right\}, \end{aligned}$$

using the fact that $2 \leqslant d = o(\log n)$ and $k = \log n$. Setting c_1 to be a large enough positive constant, we conclude that with high probability the maximum load is at most

$$\log_d \log n + \mathcal{O}(1) + c_2 = \log_d \log n + \mathcal{O}(1/\varepsilon),$$

where $c_2 = \mathcal{O}(1/\varepsilon)$. This proves the first statement of Theorem 2. The proof of the second statement is presented in Appendix D. $\square$

3 Balanced Allocation on Dynamic Graphs

In this section we show an upper bound for maximum load attained by the balanced allocation on regular dynamic graphs (i.e., Theorem 6). Suppose that the balanced allocation process has allocated n balls to the dynamic regular graph $(G^{(1)}, \dots, G^{(n)})$. Define the *conflict graph* $\mathcal{C}_n$ formed by the edges selected by the n balls. The vertex set of $\mathcal{C}_n$ is the set $[n]$ of bins, and the loads of these bins are updated during the process.

Given a tree T which is a subgraph of $\mathcal{C}_n$, and vertices u, v of the tree, if $\{u, v\}$ is an edge of $\mathcal{C}_n$ then we say it is a *cycle-producing edge* with respect to the tree T. The name arises as adding this edge to the tree would produce a cycle, which may be a 2-cycle if the edge $\{u, v\}$ is already present in T. For a positive integer $c > 0$, a subgraph of $\mathcal{C}_n$ is called *c-loaded* if each vertex (bin) contained in the subgraph has load at least c. The following proposition presents some properties of connected components of $\mathcal{C}_n$.

Proposition 13. *Let $(G^{(1)}, \dots, G^{(n)})$ be a regular dynamic graph on vertex set $[n]$ which is ε-visible. Let that $\mathcal{C}_n$ be the conflict graph obtained after allocating n balls using the balanced allocation process. Then for every given constant $c > 0$, with probability at least $1 - n^{-c}$, every $12(c+1)$-loaded connected component of $\mathcal{C}_n$ contains strictly fewer than $\log n$ vertices. Moreover, the number of cycle-producing edges in the component is at most $2(c+1)/\varepsilon$.*

We will prove the proposition in Appendix F. We now explain how to recursively build a witness graph, provided there exists a bin whose load is higher than a certain threshold.

Construction of the Witness Graph Let us start with a bin, say r, with $\ell + c$ balls. Clearly, if a ball is in bin r at height h then the other bin it chose, as part of the balanced allocation procedure, had load at least h. Starting from bin (vertex) r, let us recover all ℓ edges corresponding to the balls that were placed in r with height at least c. Thus, the alternative bin choices have loads at least $\ell + c - 1, \dots, c$, respectively. These ℓ bins are all neighbours of r in $\mathcal{C}_n$, and we refer to them as the children of r. Next, we recover the edges corresponding to balls placed in the children of r at height at least c. Recursively, we continue until there is no ball remaining at height c or more. For every $i = 1, \dots, \ell$, let $f(\ell - i)$ denote the number of vertices generated by the recursive construction, starting with a bin which contains $\ell - i + c$ balls. Assume for the moment that, for each vertex with load at least c, the recursive procedure always gives produces distinct children. Then

$$f(\ell) \geqslant f(\ell - 1) + f(\ell - 2) + \ldots + f(0) + 1,$$

where $f(0) = 1$. A simple calculation shows that $f(\ell) \geqslant 2^{\ell}$. Thus, the recursive procedure gives a c-loaded tree with at least 2^{ℓ} vertices, under the assumption that the children of each vertex considered by the recursion are all distinct.

We may now prove our main result on dynamic regular graphs.

Proof of Theorem 6. We want to show that after n balls have been allocated to the dynamic regular graph $(G^{(1)}, \dots, G^{(n)})$, which satisfies the ε-visibility property, the maximum load is at most $\log_2 \log n + \mathcal{O}(1/\varepsilon)$ with high probability.

Let $c > 0$ be a given constant. By the second statement of Proposition 13, with probability at least $1 - n^{-c}$, the number of cycle-producing edges in a given component of $\mathcal{C}_n$ is at most $c_2 = 2(c+1)/\varepsilon$. For a contradiction, suppose that there exists a bin, say r, which has at least $\ell + c_1 + c_2 + 1$ balls, where $c_1 = 12(c+1)$. Consider $c_2 + 1$ balls in r at height at least $\ell + c_1$. The children of r in $\mathcal{C}_n$ are the bins $r_1, r_2, \dots, r_{c_2+1}$ (which might not be distinct), which were the alternative choice of these $c_2 + 1$ balls. Each of these children r_i has load at least $\ell + c_1$. We start the recursive construction at each child r_i of r. Assuming that this component of $\mathcal{C}_n$ contains at most c_2 cycle-producing edges, it follows that for at least one child r_i of r, the recursive procedure gives distinct children for each vertex which is a descendent of r_i. Hence we obtain a c_1-loaded tree which has 2^{ℓ} vertices. Substituting $\ell = \log_2 \log n$ and applying the first statement of Proposition 13, we conclude that with probability at least $1 - n^{-c}$ such a structure does not exist in $\mathcal{C}_n$. This contradiction shows that with high probability, the maximum load after n balls have been allocated is at most $\log_2 \log n + \mathcal{O}(1/\varepsilon)$. $\square$

References

[1] Anders Aamand, Mathias Bæk Tejs Knudsen, and Mikkel Thorup. Power of d choices with simple tabulation. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPIcs*, pages 5:1–5:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ICALP.2018.5.

[2] Yossi Azar, Andrei Z. Broder, Anna R. Karlin, and Eli Upfal. Balanced allocations. *SIAM J. Comput.*, 29(1):180–200, 1999.

[3] Petra Berenbrink, André Brinkmann, Tom Friedetzky, and Lars Nagel. Balls into bins with related random choices. *J. Parallel Distrib. Comput.*, 72(2):246–253, 2012.

[4] Paul Bogdan, Thomas Sauerwald, Alexandre Stauffer, and He Sun. Balls into bins via local search. In *Proc. 24th Symp. Discrete Algorithms (SODA)*, pages 16–34, 2013.

[5] John W. Byers, Jeffrey Considine, and Michael Mitzenmacher. Geometric generalizations of the power of two choices. In *Proc. 16th Symp. Parallelism in Algorithms and Architectures (SPAA)*, pages 54–63, 2004.

[6] L. Elisa Celis, Omer Reingold, Gil Segev, and Udi Wieder. Balls and bins: Smaller hash families and faster evaluation. *SIAM J. Comput.*, 42(3):1030–1050, 2013. doi:10.1137/120871626.

[7] Xue Chen. Derandomized balanced allocation. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2513–2526, 2019. doi:10.1137/1.9781611975482.154.

[8] Kai-Min Chung, Henry Lam, Zhenming Liu, and Michael Mitzenmacher. Chernoff–Hoeffding bounds for Markov chains: Generalized and simplified. In *29th International Symposium on Theoretical Aspects of Computer Science, STACS 2012, February 29th - March 3rd, 2012, Paris, France*, pages 124–135, 2012. doi:10.4230/LIPIcs.STACS.2012.124.

[9] Andrea E. F. Clementi, Angelo Monti, Francesco Pasquale, and Riccardo Silvestri. Information spreading in stationary markovian evolving graphs. *IEEE Trans. Parallel Distrib. Syst.*, 22(9):1425–1432, 2011. doi:10.1109/TPDS.2011.33.

[10] Xavier Dahan. Regular graphs of large girth and arbitrary degree. *Combinatorica*, 34(4):407–426, 2014.

[11] Søren Dahlgaard, Mathias Bæk Tejs Knudsen, Eva Rotenberg, and Mikkel Thorup. The power of two choices with simple tabulation. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA 16, page 16311642, USA, 2016. Society for Industrial and Applied Mathematics.

[12] Brighten Godfrey. Balls and bins with structure: balanced allocations on hypergraphs. In *Proc. 19th Symp. Discrete Algorithms (SODA)*, pages 511–517, 2008.

[13] Krishnaram Kenthapadi and Rina Panigrahy. Balanced allocation on graphs. In *Proc. 17th Symp. Discrete Algorithms (SODA)*, pages 434–443, 2006.

[14] Donald Knuth. *The Art of Computer Programming, Vol. 1: Fundamental Algorithms*. Adison-Wesley, third edition, 1997.

[15] David A. Levin, Yuval Peres and Elizabeth L. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, 2006.

[16] Michael Mitzenmacher, Andréa W. Richa, and Ramesh Sitaraman. The power of two random choices: A survey of techniques and results. In *in Handbook of Randomized Computing*, pages 255–312. Kluwer, 2000.

[17] OEIS Foundation Inc. (2020). The On-Line Encyclopedia of Integer Sequences. *http://oeis.org/A000108*.

[18] Yuval Peres, Kunal Talwar, and Udi Wieder. Graphical balanced allocations and the $(1+\beta)$-choice process. *Random Struct. Algorithms*, DOI: 10.1002/rsa.20558, 2014.

[19] Ali Pourmiri. Balanced allocation on graphs: A random walk approach. *Random Struct. Algorithms*, 55(4):980–1009, 2019. doi:10.1002/rsa.20875.

[20] Berthold Vöcking. How asymmetry helps load balancing. *J. ACM*, 50(4):568–589, 2003.

[21] Udi Wieder. Hashing, load balancing and multiple choice. *Foundations and Trends in Theoretical Computer Science*, 12(3-4):275–379, 2017. doi:10.1561/0400000070.

A Proof of Theorem 4

Proof. Let $G=([n],E)$ denote a s-regular graph that does not contain any 4-cycle, where $s=n^{\varepsilon}$. It is worth mentioning that there are several explicit families of s-regular graphs with girth $\log_s n$ (e.g., see [10]). For each $i\in[n]$, let $N(i)$ be set of vertices adjacent to i. Also, let $\mathcal{H}=([n],\{N(i), i=1,\ldots,n\})$ denote a hypergraph obtained from G. We consider the s-uniform dynamic hypergraph $(\mathcal{H},\mathcal{H},\ldots,\mathcal{H})$. Clearly, for every $\{i,j\}\subset[n]$ we have that

$$\mathtt{vis}(i,j)\leqslant n\leqslant sn^{1-\varepsilon}$$

Therefore, the dynamic hypergraph is ε-visible. Fix an integer d such that $2\leqslant d\leqslant s$ and d is constant. Since G does not contain any 4-cycle, we deduce that every d-subset of vertices only appears in at most one hyperedge of $\mathcal{H}$. Therefore, the probability that a d-subset is chosen by any ball is $1/(n\binom{s}{d})$. Let $D=\{i_1,i_2,\ldots,i_d\}\subset[n]$ be an arbitrary set of d vertices contained in some hyperedge of $\mathcal{H}$. Let $X(D,k)$ be an indicator random variable taking one if at least k balls choose D and zero otherwise. Then we have that

$$\mathbf{Pr}\left[X(D,k)=1\right]=\binom{n}{k}\left(\frac{1}{n\binom{s}{d}}\right)^k$$

Also let $Y_k=\sum_D X(D,k)$ denote the number of d-subsets that are chosen by at least k balls. By linearity of expectation we have that

$$\mathbf{E}\left[Y_k\right]=\sum_D \mathbf{E}\left[X(D,k)\right]=n\binom{s}{d}\binom{n}{k}\left(\frac{1}{n\binom{s}{d}}\right)^k\geqslant n\left(\frac{s^{-d}}{k}\right)^k=n\left(\frac{n^{-d\varepsilon}}{k}\right)^k, \tag{2}$$

where the last inequality follows from $\binom{n}{k}\geqslant(\frac{n}{k})^k$ and $\binom{s}{d}<s^d$. In what follows we show that with high probability there exists k such that $Y_k\geqslant 1$. Suppose that $d\varepsilon=\Theta(1)$, then if we set $k=1$, then

there is a d-subset which is picked by at least one ball and hence $Y_1 \geqslant 1$. If $(\log\log n)/(3\log n) < d\varepsilon$ and $d\varepsilon = o(1)$, then by setting $k = 1/(6d\varepsilon)$ we have $k < (\log n)/(2\log\log n) < \log n$ and

$$\mathbf{E}[Y_k] \geqslant nk^{-k}n^{-kd\varepsilon} \geqslant n(\log n)^{-\log n/(2\log\log n)}n^{-1/6} = n^{1/3} = \omega(\log n).$$

Moreover, if $d\varepsilon \leqslant \log\log n/(3\log n)$, then by letting $k = \log n/(2\log\log n)$ we get that

$$\mathbf{E}[Y_k] \geqslant nk^{-k}n^{-kd\varepsilon} \geqslant n(\log n)^{-\log n/(2\log\log n)}n^{-1/6} = n^{1/3} = \omega(\log n).$$

Therefore, there exists $k = \min\{\Omega(1/\varepsilon), \Omega(\log n/\log\log n)\}$ so that $\mathbf{E}[Y_k] = \omega(\log n)$. As the number of balls is n, it is easy to observe that for a given k, the random variables $X(D,k)$ are negatively correlated. Application of the Chernoff bound for negatively correlated random variable implies that

$$\mathbf{Pr}[Y_k \leqslant \mathbf{E}[Y_k]/2] \leqslant \exp(-\mathbf{E}[Y_k]/8) = \exp(-\omega(\log n)).$$

It follows that there exists a d-subset D which is chosen by at least k balls and hence there is at least one bin in D whose load is at least k/d. □

B Proof of Proposition 8

In this section we prove Proposition 8. First we restate a useful theorem from [8].

Theorem 14. [8, Theorem 3] *Let M be an ergodic Markov chain with finite state space Ω and stationary distribution π. Let $T = T(\varepsilon)$ be its ε-mixing time for $\varepsilon < 1/8$. Let $(Z_1, \ldots, Z_t)$ denote a t-step random walk on M starting from an initial distribution ρ on Ω (that is, Z_1 is distributed according to ρ). For some positive constant μ and every $i \in [t]$, let $f_i : \Omega \to [0,1]$ be a weight function at step i such that the expected weight $\mathbf{E}_\pi[f_i(v)] = \sum_{v\in\Omega}\pi(v)f_i(v)$ satisfies $\mathbf{E}_\pi[f_i(v)] = \mu$ for all i. Define the total weight of the walk $(Z_1, ..., Z_t)$ by $X = \sum_{i=1}^{t} f_i(Z_i)$. Write $||\rho||_\pi = \sqrt{\sum_{x\in\Omega}\rho_x^2/\pi_x}$. Then there exists some positive constant c (independent of μ and ε) such that for all $\alpha \geqslant 0$,*

1. $\mathbf{Pr}[X \geqslant (1+\alpha)\mu t] \leqslant c||\rho||_\pi\, \mathrm{e}^{-\alpha^2\mu t/72T}$ *for* $0 \leqslant \alpha \leqslant 1$.
2. $\mathbf{Pr}[X \geqslant (1+\alpha)\mu t] \leqslant c||\rho||_\pi\, \mathrm{e}^{-\alpha\mu t/72T}$ *for* $\alpha > 1$.
3. $\mathbf{Pr}[X \leqslant (1-\alpha)\mu t] \leqslant c||\rho||_\pi\, \mathrm{e}^{-\alpha^2\mu t/72T}$ *for* $0 \leqslant \alpha \leqslant 1$.

Proof of Proposition 8. Let Ω be the vertex set of the R-dimensional torus $\Gamma(n,R)$ and let a and b denote two arbitrary agents. By definition of the communication graph process, agents a and b are initially placed on two randomly chosen vertices of Γ, say u_0 and v_0. Note that u_0 and v_0 are independently chosen according to the stationary distribution π of the random walk on $\Gamma(n,R)$. Now consider the trajectory of agents a and b, which give two independent random walks $u_0, u_1, \ldots$ and $v_0, v_1, \ldots$ on $\Gamma(n,R)$. Defining $X_t = (u_t, v_t)$ for $t = 0, 1, \ldots$ gives a finite, ergodic Markov chain with stationary distribution (π,π) on $\Omega\times\Omega$. For every $t \geqslant 0$, define

$$f(X_t) = f(u_t, v_t) = \begin{cases} 1 & \text{if } d(u_t, v_t) \leqslant r\ , \\ 0 & \text{otherwise.} \end{cases}$$

where $d(\cdot,\cdot)$ is the Manhattan distance for the given grid. Let u_t^1 and v_t^1 denote the projection of the random walks u_t and v_t onto the 1-dimensional torus $\Gamma(n^{1/R}, 1)$, respectively, defined by taking the first component of each of the random walks on $\Gamma(n,R)$. Then $X_t^1=(u_t^1,v_t^1)$ is an ergodic Markov chain on $\Gamma(n^{1/R},1)$, and its initial distribution is stationary. We may also define

$$f(u_t^1,v_t^1)=\begin{cases}1 & \text{if } d(u_t^1,v_t^1)\leqslant r\ ,\\ 0 & \text{otherwise.}\end{cases}$$

By the Manhattan distance property, if $f(u_t,v_t)=1$ then $f(u_t^1,v_t^1)=1$. Therefore,

$$\mathtt{vis}(a,b)=\sum_{t=0}^{n} f(X_t)\leqslant\sum_{t=0}^{n} f(X_t^1).$$

Set $\delta=\min\{1/4,1/R\}$. Let t_0 be the first time when $d(u_{t_0}^1,v_{t_0}^1)\leqslant n^\delta$. Consider a moving window W of length $2n^\delta+1$, which contains the locations of $u_{t_0}^1$ and $v_{t_0}^1$. At time t_0, the vertices covered by W are labelled in increasing order, with the leftmost vertex labelled $-n^\delta$ and the rightmost vertex labelled $n^\delta-1$. The window W stays at its initial location as long as no agent hits a border of W (vertices labelled $-n^\delta$ or n^δ), or the middle vertex of W (labelled 0). Let b be the first agent that hits a border or the centre of W. From this time on, b and W are coupled so that they both move and/or stay, simultaneously. (If b moves left then W also moves left, for example.) Each time the window W moves, a vertex $u\in\Gamma_1$ is no longer covered by W and a new vertex, $w\in\Gamma_1$, becomes covered by w. The new vertex w is assigned the label of vertex u. This process always labels the vertices covered by W by $\{-n^{\delta-1},\ldots,n^\delta-1\}$, and the movement of agent b over these labeled vertices simulates a random walk on the additive group $\mathbb{Z}_{2n^\delta+1}$. Define

$$S=\{1\leqslant t\leqslant n \mid u_t^1 \text{ and } v_t^1\in W\}.$$

Assume that $S\neq\emptyset$ and define the chain $Y_t=(u_t^1,v_t^1),t\in S$. Then Y_t can be considered as an ergodic Markov chain of length $|S|\leqslant n$ over $\mathbb{Z}_{2n^\delta-1}$, or equivalently, as a Markov chain on a $(2n^\delta+1)$-cycle. By the proposition assumption we have $r=\mathcal{O}(n^{o(1)})<n^\delta$, and so

$$\mathtt{vis}(a,b)=\sum_{t=0}^{n} f(X_t)\leqslant\sum_{t=0}^{n} f(X_t^1)\leqslant\sum_{t\in S} f(Y_t)\leqslant\sum_{t=0}^{n} f(Y_t).$$

The chain Y_t converges to stationary distribution (π,π), where π is the uniform distribution of a random walk on a $(2n^\delta+1)$-cycle. It follows that for all $t=0,1,\ldots$ we have $\mathbf{E}_{(\pi,\pi)}\left[f(Y_t)\right]=\mu=\Theta(r/n^\delta)$, independently of t. It is well-known [15] that the ε-mixing time of the random walk on a $(2n^\delta+1)$-cycle is $\mathcal{O}(n^{2\delta}\log(1/\varepsilon))$. If ρ is the initial distribution Y_0, then we have that $||\rho||_\pi\leqslant\mathcal{O}(n^\delta)$. Applying Theorem 14 implies that

$$\mathbf{Pr}\left[\sum_{t=1}^{n} f(Y_t)\geqslant\mu\cdot n\right]=\mathcal{O}(n^\delta)\mathrm{e}^{-\Theta(rn^{1-3\delta})}=n^{-\omega(1)}.$$

Therefore, with probability $1-n^{-\omega(1)}$,

$$\mathtt{vis}(a,b)\leqslant\sum_{t=0}^{n} f(Y_t)=\mathcal{O}(rn^{1-\delta})=\mathcal{O}(n^{1-\delta+o(1)})=\mathcal{O}(n^{1-\varepsilon}),$$

taking $\varepsilon=\delta/2$, say. Taking the union bound over all pairs of agents completes the proof.

□

C Appearance Probability of a Certain Structure

In this subsection we work towards a proof of Lemma 12. First we will give some useful definition and prove some helpful results. The definition was introduced in [19].

Definition 15. Suppose that $\mathcal{A}$ is an allocation algorithm that sequentially allocates n balls into n bins according to some mechanism. For a given constant $\alpha > 0$, and for $\Theta(n) = m \leqslant n$, we say that $\mathcal{A}$ is (α, m)-uniform if for every ball $1 \leqslant t \leqslant m = \Theta(n)$ and every bin $i \in [n]$,

$$\mathbf{Pr}\left[\, \text{ball } t \text{ is allocated to bin } i \text{ by } \mathcal{A} \mid \text{balls } 1, 2, \ldots, t-1 \text{ have been allocated by } \mathcal{A} \,\right] \leqslant \frac{\alpha}{n}.$$

In the above definition, we condition on the allocations of balls $1, \ldots, t-1$ into bins made by $\mathcal{A}$.

The following result, proved in Appendix E, states that the balanced allocation process is uniform on dynamic hypergraphs.

Lemma 16 (Uniformity Lemma)**.** *Fix $d = d(n)$ with $2 \leqslant d = o(\log n)$ and suppose that for some constant $\beta \geqslant 1$, the s-uniform dynamic hypergraph $(\mathcal{H}^{(1)}, \ldots, \mathcal{H}^{(n)})$ satisfies the β-balanced and size properties, with $d \leqslant s$. Then there exists a constant $\alpha = \alpha(\beta)$, which depends only on β, and there exists $m = \Theta(n)$ with $m < n$, such that the balanced allocation process on $(\mathcal{H}^{(1)}, \ldots, \mathcal{H}^{(n)})$ is (α, m)-uniform. Specifically, we may take $\alpha = 44\beta$.*

We are ready to prove Lemma 12.

Lemma 17 (Restatement of Lemma 12)**.** *Fix $d = d(n)$ with $2 \leqslant d = o(\log n)$. Let $(\mathcal{H}^{(1)}, \ldots, \mathcal{H}^{(n)})$ be a dynamic hypergraph which satisfies the β-balanced, ε-visibility and c_0-size properties. Suppose that $c \geqslant 44\beta e^2$ is a sufficiently large constant, and let $k = C \log n$ for some constant $C \geqslant 1$. There exists $\Theta(n) \leqslant m \leqslant n$ such that the probability that $\mathcal{C}_m$ contains a c-loaded k-vertex tree is at most*

$$\exp\Big\{4k \log(2\beta d) - c(d-1)(k-r-1) + \big(c_0 + 3 - r\varepsilon/2\big)\log(n)\Big\}$$

where r is the number of red vertices in the blue-red coloring of the tree. Moreover, with high probability, if $\mathcal{C}_m$ contains any such tree then $r = \mathcal{O}(1/\varepsilon)$.

Proof. Fix $m = m(n)$ to equal the m provided by Lemma 16. There are at most 4^k ordered trees with k vertices. (Proposition 10). Fix such a tree, say T, and label the vertices $\{1, 2, \ldots, k\}$ such that vertex i is the i-th new vertex visited when performing depth-first search in T starting from the root, and respecting the given ordering. In particular, the root of T is vertex 1. Next, we will assign a d-choice to the root vertex of T, as a first step in describing trees which may be present in the witness graph $\mathcal{C}_m$. Let x count the number of possible d-choices that can be assigned to the root of T. Then

$$x \leqslant \binom{s}{d} \cdot \left| \bigcup_{t=1}^{m} \mathcal{E}_t \right| \cdot m \leqslant \binom{s}{d} \cdot n^{c_0+2},$$

where the last inequality follows from the size property and the inequality $m \leqslant n$. Therefore, there are x possibilities for the root and hence there are at most $4^k \cdot \binom{s}{d} \cdot n^{c_0+2}$ ordered trees with the specified root. Fix an arbitrary d-choice D_t as the root for T.

Next we fix an arbitrary function $\mathrm{col} : \{2, \ldots, k\} \to \{\mathrm{blue}, \mathrm{red}\}$, that gives a blue-red coloring of $2, \ldots, k$. In what follows we establish an upper bound for the probability that $\mathcal{C}_m$ contains the blue-red colored tree $T \subset \mathcal{C}_m$, (according to Definition 11). Let $q_1(t)$ be the probability that the t-th ball chooses the root of T (that is, that the d-choice made by the t-th ball corresponds to the root of T). Then

$$\sum_{t=1}^{m} q_1(t) \leqslant \sum_{t=1}^{m} \frac{1}{\binom{s}{d}} \leqslant \frac{n}{\binom{s}{d}}, \tag{3}$$

because H contains $\binom{s}{d}$ distinct d-element sets for for every $H \in \mathcal{E}_t$. For every $t = 2, \ldots, k$, define $q_i(t, \mathrm{col}(i))$ to be the probability that the t-th ball chooses the i-th vertex of the tree (i.e., i) with $\mathrm{col}(i)$. If $\mathrm{col}(i)$ is red then D_t must share at least two bins with $\cup_{j=1}^{i-1} D_{t_j}$, while if $\mathrm{col}(i)$ is blue then D_t only shares one bin with its parent. For every $i = 2, \ldots, k$, let us derive an upper bound on $q_i(t, \mathrm{blue})$. Here, the i-th vertex share one bin with its parent in T, say D_{t_j}. Now D_{t_j} has d bins and by the balancedness property we get

$$\mathbf{Pr}\left[D_{t_j} \cap H_t \neq \emptyset\right] \leqslant \sum_{i \in D_{t_j}} \mathbf{Pr}\left[i \in H_t\right] \leqslant \frac{\beta d s}{n},$$

where H_t is the edge chosen by ball t from $\mathcal{H}^{(t)}$, uniformly at random. Suppose that for some $a \geqslant 1$ we have $|D_{t_j} \cap H_t| = a \leqslant d$. Then the total number of d-element subsets of H_t which share only one bin with D_{t_j} is $a\binom{s-a}{d-1} \leqslant d\binom{s-1}{d-1}$. Thus, we get

$$\sum_{t=1}^{m} q_i(t, \mathrm{blue}) \leqslant \sum_{t=1}^{m} \frac{\beta d s}{n} \cdot d \frac{\binom{s-1}{d-1}}{\binom{s}{d}} = \sum_{t=1}^{m} \frac{\beta d^3}{n} \leqslant \beta d^3, \tag{4}$$

because $m \leqslant n$.

Next, for every $i = 2, \ldots, k$, and every $t = 2, \ldots, m$, we need an upper bound on $q_i(t, \mathrm{red})$. If the i-th vertex of the tree is the set D_t and is coloured red, then D_t is a d-element set of bins which shares at least two bins with $\cup_{j=1}^{i-1} D_{t_j}$. One of these bins belongs to the (known) parent, and the other belongs to $D_{t_1} \ldots, D_{t_{i-1}}$. So if U is the number of choices for this pair of bins, then

$$U \leqslant d \cdot (i-1)d \leqslant kd^2. \tag{5}$$

Let $\{p_1, p_2 \ldots, p_U\}$ be the set of such pairs of bins. For $J = 1, \ldots, U$, write $A(p_J, t)$ for the event that the pair p_J is contained in a randomly chosen edge of $\mathcal{E}_t$. Observe that if $p_J \subset D_t$ then $A(p_J, t)$ holds. Then, by the balancedness property we have

$$\begin{aligned}
&\mathbf{Pr}\left[p_J \subset D_t\right] \\
&= \mathbf{Pr}\left[p_J \subset D_t \mid A(p_J, t)\right] \cdot \mathbf{Pr}\left[A(p_J, t)\right] \\
&\leqslant \mathbf{Pr}\left[p_J \subset H_t\right] \cdot \frac{\binom{s-2}{d-2}}{\binom{s}{d}} \cdot \mathbf{Pr}\left[A(p_J, t)\right] \\
&\leqslant \mathbf{Pr}\left[p_J \cap H_t \neq \emptyset\right] \cdot \frac{\binom{s-2}{d-2}}{\binom{s}{d}} \cdot \mathbf{Pr}\left[A(p_J, t)\right] \\
&\leqslant \frac{2\beta s}{n} \cdot \frac{\binom{s-2}{d-2}}{\binom{s}{d}} \cdot \mathbf{Pr}\left[A(p_J, t)\right] = \frac{2\beta d(d-1)}{(s-1)n}\ \mathbf{Pr}\left[A(p_J, t)\right],
\end{aligned}$$

as $\binom{s-2}{d-2}$ is the number of d-element subsets of H_t which contain the pair p_J. Then

$$q_i(t, \mathrm{red}) \leqslant \sum_{J=1}^{U} \frac{2\beta d(d-1)}{(s-1)n}\ \mathbf{Pr}\left[A(p_J, t)\right].$$

Note that by (5) we have $U \leqslant kd^2$ and hence,

$$\sum_{t=1}^{m} q_i(t, \mathrm{red}) \leqslant \sum_{J=1}^{U} \sum_{t=1}^{n} \frac{2\beta d(d-1)}{(s-1)n}\ \mathbf{Pr}\left[A(p_J, t)\right] \leqslant \sum_{J=1}^{U} \frac{2\beta d(d-1)}{(s-1)n}\ \mathtt{vis}(p_J) \leqslant \frac{2\beta k d^4}{n^{\varepsilon}}. \tag{6}$$

The final inequality follows from the visibility property, using the fact that $d < s$.

Write $\mathrm{col}^{-1}(\mathrm{blue})$ for the set of blue vertices in T, and similarly for $\mathrm{col}^{-1}(\mathrm{red})$. Then

$$|\mathrm{col}^{-1}(\mathrm{red})| + |\mathrm{col}^{-1}(\mathrm{blue})| = k - 1.$$

Suppose that $(t_1, \dots, t_k)$ is the sequence of balls that are going to select vertices $1, 2, \dots, k$ of T. By applying (3), (4) and (6), we find that the probability that the edges of the colored tree T appears in $\mathcal{C}_m$ at times $(t_1, \dots, t_k)$, and the corresponding sets $D_{t_1}, \dots, D_{t_k}$ consistent with chosen blue-red coloring scheme, is at most

$$\begin{aligned}
&\sum_{(t_1,\dots,t_k)} \left\{ q_1(t_1) \prod_{i=2}^{k} q_i(t_i, \mathrm{col}(i)) \right\} \leqslant \left(\sum_{t=1}^{m} q_1(t) \right) \prod_{2=1}^{k} \left(\sum_{t=1}^{m} q_i(t, \mathrm{col}(i)) \right) \\
&\leqslant \frac{n}{\binom{s}{d}} \left(\prod_{i \in \mathrm{col}^{-1}(\mathrm{blue})} \sum_{t=1}^{m} q_i(t, \mathrm{blue}) \right) \left(\prod_{i \in \mathrm{col}^{-1}(\mathrm{red})} \sum_{t=1}^{m} q_i(t, \mathrm{red}) \right) \\
&\leqslant \frac{n}{\binom{s}{d}} \left(\beta d^3\right)^{|\mathrm{col}^{-1}(\mathrm{blue})|} \left(\frac{2\beta k d^4}{n^{\varepsilon}} \right)^{|\mathrm{col}^{-1}(\mathrm{red})|} \leqslant \frac{n \beta^k d^{4k}}{\binom{s}{d}} \left(\frac{2k}{n^{\varepsilon}} \right)^{|\mathrm{col}^{-1}(\mathrm{red})|}. \qquad (7)
\end{aligned}$$

There are at most 2^{k-1} coloring functions and $4^k \mathrm{poly}(n)\binom{s}{d}$ rooted and ordered trees. So by the upper bound (7), together with the union bound over all colored ordered trees, we obtain

$$\begin{aligned}
&\mathbf{Pr}\left[\, \mathcal{C}_m \text{ contains a valid blue-red colored } k\text{-vertex tree with } r \text{ red vertices} \,\right] \\
&\leqslant 4^k 2^{k-1} \cdot n^{c_0+2} \binom{s}{d} \cdot \frac{n\, \beta^k d^{4k}}{\binom{s}{d}} \left(\frac{2k}{n^{\varepsilon}} \right)^r \\
&\leqslant n^{c_0+3} \cdot (2\beta d)^{4k} \cdot n^{-r\varepsilon/2} \leqslant \exp\big(4k \log(2\beta d) + (c_0 + 3 - r\varepsilon/2) \log n\big), \qquad (8)
\end{aligned}$$

using $k = \mathcal{O}(\log n)$ for the penultimate inequality.

Let $b = k - r - 1$ be the number of blue vertices and let $D_{s_1}, \dots, D_{s_b}$ be the sorted list of blue vertices such that $s_1 < s_2 < \cdots < s_b$. Then, by the definition of blue-red coloring, for every $j = 1, \dots, b$ we have $|(\cup_{g=1}^{j-1} D_{s_g}) \cap D_{s_j}| \leqslant 1$. This implies that

$$y = |\cup_{j=1}^{k} D_{t_j}| \geqslant |\cup_{j=1}^{b} D_{s_j}| \geqslant (d-1)b = (d-1)(k-1-r),$$

since $\{s_1, \dots, s_b\} \subseteq \{t_1, \dots, t_k\}$. Applying Lemma 16 implies that the balanced allocation is (α, m)-uniform, where $\alpha = 44\beta$, say. Hence for any $c \geqslant 44\beta \mathrm{e}^2$, the probability that each bin in $\cup_{j=1}^{k} D_{t_j}$ is allocated at least c balls (that is, the tree T is c-loaded) is at most

$$\binom{m}{cy} \left(\frac{\alpha y}{n} \right)^{cy} \leqslant \left(\frac{\mathrm{e} m}{cy} \right)^{cy} \left(\frac{\alpha y}{n} \right)^{cy} \leqslant \left(\frac{\mathrm{e}\alpha}{c} \right)^{cy} \leqslant \mathrm{e}^{-c(d-1)(k-r-1)},$$

where the last inequality follows from $m \leqslant n$ and the fact that $c > \alpha \mathrm{e}^2$. Since balls are independent from each other, we can multiply the above inequality by (8) to show that the probability that $\mathcal{C}_m$ contains a c-loaded k-vertex tree with r red vertices is at most

$$\exp\Big\{ 4k \log(2\beta d) - c(d-1)(k-r-1) + \big(c_0 + 3 - r\varepsilon/2\big) \log n \Big\}, \qquad (9)$$

proving the first statement of the lemma. Finally, suppose that $r\varepsilon \to \infty$ as $n \to \infty$. Then the upper bound in (9) can be written as

$$\begin{aligned}
&\exp\Big\{ \big(4 \log(2\beta d) - c(d-1)\big)k + \mathcal{O}(\log n) + o(r \cdot \log n) - (r\varepsilon/2) \log n \Big\} \\
&\leqslant \exp\Big\{ \mathcal{O}(\log n) + o(r \cdot \log n) - (r\varepsilon/2) \log n \Big\}.
\end{aligned}$$

Since $r\varepsilon \to \infty$, this term dominates and the probability that $\mathcal{C}_m$ contains a blue-red coloured tree with r red vertices tends to zero. Therefore, if such a tree is present in $\mathcal{C}_m$ then $r = \mathcal{O}(1/\varepsilon)$ with high probability. This completes the proof. □

D Missing Part of Proof of Theorem 2

In order to prove the second statement of Theorem 2 we show the sub-additivity of the balanced allocation algorithm. We want to prove that for every constant integer $\gamma \geqslant 1$ with $\gamma m \leqslant n$, after allocating γm balls, the maximum load is at most $\gamma(\log_d \log n + \mathcal{O}(1/\varepsilon))$, with high probability. First assume that $2m \leqslant n$ and suppose that the algorithm has allocated m balls to $\mathcal{H}^{(t)}, t = 1, \ldots, m$ and let $\ell^* \leqslant \log_d \log n + \mathcal{O}(1)$ denote its maximum load. We now consider two independent balanced allocation algorithms, say $\mathcal{A}$ and $\mathcal{A}_0$, on two dynamic hypergraphs starting from step m. These dynamic hypergraphs are $(\mathcal{H}^{(m)}, \ldots, \mathcal{H}^{(n)})$ and $(\mathcal{H}_0^{(m)}, \ldots, \mathcal{H}_0^{(n)})$, where $\mathcal{H}_0^{(t)}$ is an identical copy of $\mathcal{H}^{(t)}$ for $t = m, \ldots, n$. Moreover, we assume that in round m, all bins contained in $\mathcal{H}_0^{(m)}$ have exactly ℓ^* balls. Let us couple algorithm $\mathcal{A}$ on $\mathcal{H}^{(t)}$ and algorithm $\mathcal{A}_0$ on $\mathcal{H}_0^{(t)}$. Write $V = [n]$ for the set of n bins. To do so, the coupled process allocates a pair of balls to bins as follows: for $t = m + 1, \ldots, 2m$, the coupling chooses a one-to-one labeling function $\sigma_t : V \to \{1, 2, \ldots, n\}$ uniformly at random, where V is the ground set of both hypergraphs (i.e, set of n bins) and $\{1, 2, \ldots, n\}$ is a set of labels. Next, the coupling chooses D_t randomly from $\mathcal{H}^{(t)}$. Let D'_t denote the same set of d bins as D_t in $\mathcal{H}_0^{(t)}$. Algorithm $\mathcal{A}$ allocates ball $t+1$ to a least-loaded vertex of D_t, and algorithm $\mathcal{A}_0$ allocates ball $t+1$ to a least-loaded vertex of D'_t, with both algorithms breaking ties in favour of the vertex v with the smallest load and minimum label $\sigma_t(v)$. Note that algorithm $\mathcal{A}$ is a faithful copy of the balanced allocation process on $(\mathcal{H}^{(m)}, \ldots, \mathcal{H}^{(n)})$, and algorithm $\mathcal{A}_0$ is a faithful copy of the balanced allocation process on $(\mathcal{H}_0^{(m)}, \ldots, \mathcal{H}_0^{(n)})$, respectively. (This follows as σ_t is chosen uniformly at random.) Let X_i^t and Y_i^t, $m+1 \leqslant t \leqslant 2m$, denote the load of bin i in $\mathcal{H}^{(t)}$ and $\mathcal{H}_0^{(t)}$, respectively. We prove by induction that for every integer $m \leqslant t \leqslant 2m$ and $i \in V$ we have

$$X_i^t \leqslant Y_i^t. \tag{10}$$

The inequality holds when by the assumption that $Y_i^m = \ell^*$ for every $i \in V$. Let us assume that for every t', $t' \leqslant t \leqslant 2m$, Inequality (10) holds, then we will show it for $t+1$. Let $i \in D_{t+1}$ and $j \in D'_{t+1}$ denote the vertices (bins) that receive a ball in step $t+1$. We now consider two cases:

- **Case 1:** $X_i^t < Y_i^t$. Since algorithm $\mathcal{A}$ allocated ball $t+1$ to bin i, it follows that

$$X_i^t + 1 = X_i^{t+1} \leqslant Y_i^t \leqslant Y_i^{t+1}.$$

  So, Inequality (10) holds for $t+1$ and every bin $i \in V$.

- **Case 2:** $X_i^t = Y_i^t$. Since D'_{t+1} is a copy of D_{t+1}, we have $j \in D_{t+1}$ and $i \in D'_{t+1}$. We know that no vertex (bin) in D_{t+1} has smaller load than i, and no vertex (bin) in D'_{t+1} has smaller load than j. Hence

$$X_i^t \leqslant X_j^t \leqslant Y_j^t \leqslant Y_i^t,$$

  where the middle inequality follows from the inductive hypothesis (10) for bin j. So by assumption of this case we obtain $X_i^t = X_j^t = Y_j^t = Y_i^t$. If $i \neq j$ and $\sigma_{t+1}(j) < \sigma_{t+1}(i)$, then it contradicts the fact that ball $t+1$ is allocated to bin i by algorithm $\mathcal{A}$. Similarly, if $\sigma_{t+1}(j) > \sigma_{t+1}(i)$, then it contradicts the fact that algorithm $\mathcal{A}_0$ allocated ball t to bin j. Therefore $i = j$ and hence

$$X_i^{t+1} = X_i^t + 1 = Y_i^t + 1 = Y_i^{t+1}.$$

Thus, in both cases, Inequality (10) holds for every $t \geqslant 0$. By applying the first part of the theorem, with high probability, using algorithm $\mathcal{A}_0$ to allocate m balls to the dynamic hypergraph $(\mathcal{H}_0^{(m)}, \ldots, \mathcal{H}_0^{(2m)})$ results in maximum load

$$\ell^* + \log_d \log n + \mathcal{O}(1/\varepsilon) \leqslant 2(\log_d \log n + \mathcal{O}(1/\varepsilon))$$

in $\mathcal{H}_0^{(2m)}$. Therefore, by Inequality (10), after using algorithm $\mathcal{A}$ to allocate m balls to the dynamic hypergraph $(\mathcal{H}^{(m)}, \ldots, H^{(n)})$, with high probability the maximum load in $\mathcal{H}^{(2m)}$ is at most $2(\log_d \log n + \mathcal{O}(1/\varepsilon))$. Applying the union bound, we conclude that after allocating γm balls, where $\gamma m \leqslant n$, the maximum load is at most $\gamma(\log_d \log n + \mathcal{O}(1/\varepsilon))$, with high probability.

E Proof of Lemma 16

Berenbrink et al. [3] proposed an allocation algorithm $\mathcal{B}$ such that for $t = 1, 2, \ldots$, the t-th ball chooses an edge of $\mathcal{H}^{(t)} = ([n], \mathcal{E}_t), t = 1, \ldots$, uniformly at random, say H_t. The ball is then allocated to an empty vertex (bin) of H_t, with ties broken randomly. If H_t does not contain an empty bin then the process fails. The next lemma follows directly from [3, Lemmas 4, 5].

Lemma 18. *Suppose that the dynamic s-uniform hypergraph $(\mathcal{H}^{(1)}, \ldots, \mathcal{H}^{(n)})$ satisfies the balancedness and size properties. There exists $m = \Theta(n)$ such that with probability at least $1-n^{-2}$, algorithm $\mathcal{B}$ successfully allocates m balls and there are at least $s/2$ empty vertices in H_t for $t = 1, \ldots, m$.*

We now apply the above result to show the same property holds for the balanced allocation on any dynamic hypergraph.

Lemma 19. *Fix $d = d(n)$ with $2 \leqslant d = o(\log n)$. Suppose that the dynamic s-uniform hypergraph $(\mathcal{H}^{(1)}, \ldots, \mathcal{H}^{(n)})$ satisfies the balancedness and size properties. There exists $m = \Theta(n)$ with $m < n$ such that with probability at least $1 - n^{-2}$, the edge H_t chosen by the t-th ball contains at least $s/2$ empty vertices for $t = 1, \ldots, m$.*

Proof. We apply a coupling technique between the balanced allocation process on a dynamic hypergraph and $\mathcal{B}$.

Let us first consider an identical copy of the set of bins, called B. The coupled process sequentially allocates a ball to a pair of bins. In round $t = 1, \ldots, m$, the t-th ball chooses an edge of $\mathcal{H}^{(t)}$ uniformly at random, say H_t. Let H'_t be the corresponding set of bins, chosen from B. Then the first ball is allocated to a bin, say i, contained in H_t according the balanced allocation. If $i \in H'_t$ is empty then the second ball is allocated to bin $i \in H'_t$ as well. If $i \in H'_t$ is not empty then the second ball is allocated to an empty bin from H'_t, with ties are broken randomly. If there is no empty bin in H'_t then the coupling fails. Note that H_t and H'_t have the same set of bins but may have different loads. Observe that the coupled process allocates balls to bins from B according to $\mathcal{B}$. Next we show that for $t = 1, \ldots, m$,

$$\textsc{Empty}(H_t) \geqslant \textsc{Empty}(H'_t), \tag{11}$$

where $\textsc{Empty}(H)$ denotes the number of empty bins contained in H. For a contradiction, assume that there is a first time t_1 such that $\textsc{Empty}(H'_{t_1}) > \textsc{Empty}(H_{t_1})$. Then there is vertex $i \in H'_{t_1}$ which is empty, while $i \in H_{t_1}$ has a ball at height zero: this is ball t_0, say, where $1 \leqslant t_0 \leqslant t_1$. This implies that the coupled process has allocated ball t_0 to bin $i \in H_{t_1}$, but it has not allocated any ball to bin $i \in H'_{t_1}$, since i was empty until round t_1. This contradicts the definition of the coupled process. So Inequality (11) holds for $t = 1, \ldots, m$. Applying Lemma 18 yields that there exists $m = \Theta(n)$ such that for $t = 1, \ldots, m$,

$$\textsc{Empty}(H_t) \geqslant \textsc{Empty}(H'_t) \geqslant s/2.$$

$\square$

Proof of Lemma 16. Fix $m = m(n)$ to equal the m provided by Lemma 19. For $t = 1, \ldots, m$, let D_t be the d-element subset of H_t that is chosen by the t-th ball. Define the indicator random variable $\mathbb{I}_t$ as follows:

$$\mathbb{I}_t := \begin{cases} 1 & \text{if } D_t \text{ contains at least } d/6 \text{ empty vertices,} \\ 0 & \text{otherwise.} \end{cases}$$

Let us fix an arbitrary bin i and then define $A(t,i)$ to be the event that the t-th ball is allocated to vertex i. (The first $t-1$ balls have already been allocated, as the balanced allocation process never fails.) Observe that if $i \notin D_t$ then $\mathbf{Pr}[A(t,i)] = 0$. It follows that

$$\begin{aligned} \mathbf{Pr}[A(t,i)] &= \mathbf{Pr}[A(t,i) \mid i \in D_t \text{ and } \mathbb{I}_t = 1] \cdot \mathbf{Pr}[i \in D_t \text{ and } \mathbb{I}_t = 1] \\ &\quad + \mathbf{Pr}[A(t,i) \mid i \in D_t \text{ and } \mathbb{I}_t = 0] \cdot \mathbf{Pr}[i \in D_t \text{ and } \mathbb{I}_t = 0]. \end{aligned}$$

Now there are at least $d/6$ empty vertices on D_t when $\mathbb{I}_t = 1$, so

$$\mathbf{Pr}[A(t,i) \mid i \in D_t \text{ and } \mathbb{I}_t = 1] \leqslant 6/d.$$

It follows that

$$\begin{aligned} \mathbf{Pr}[A(t,i)] &\leqslant (6/d)\ \mathbf{Pr}[i \in D_t \text{ and } \mathbb{I}_t = 1] + \mathbf{Pr}[i \in D_t \text{ and } I_t = 0] \\ &\leqslant (6/d)\ \mathbf{Pr}[i \in D_t] + \mathbf{Pr}[\mathbb{I}_t = 0 \mid i \in D_t] \cdot \mathbf{Pr}[i \in D_t]. \end{aligned} \tag{12}$$

In order to have $i \in D_t$, first an edge containing i must be selected, and then the chosen d-element subset of that edge must contain i. By the β-balancedness property,

$$\mathbf{Pr}[i \in D_t] \leqslant \frac{\beta s}{n} \cdot \frac{\binom{s-1}{d-1}}{\binom{s}{d}} \leqslant \frac{\beta}{n}.$$

Using the above inequality, we simplify Inequality (12) as follows:

$$\mathbf{Pr}[A(t,i)] \leqslant \frac{6\beta}{n} + \frac{\beta d}{n}\ \mathbf{Pr}[\mathbb{I}_t = 0 \mid i \in D_t].$$

If $d \leqslant 6$ then the above inequality immediately implies that $\mathbf{Pr}[A(t,i)] \leqslant 12\beta/n$. This completes the proof when $d \leqslant 6$. For the remainder of the proof we assume that $d \geqslant 7$, and prove that

$$\mathbf{Pr}[\mathbb{I}_t = 0 \mid i \in D_t] \leqslant \hat{c}/d \tag{13}$$

for some absolute constant $\hat{c} > 0$. From this, we see that $\mathbf{Pr}[A(t,i)] \leqslant \alpha/n$ where $\alpha = \beta(6 + \hat{c})$. As i was an arbitrary bin, this proves that the process is (α, m)-uniform.

Let $\mathcal{F}$ be the event that H_t contains at least $s/2$ empty vertices for all $t = 1, \ldots, m$. By Lemma 19, we have $\mathbf{Pr}[\mathcal{F}] \geqslant 1 - n^{-2}$. Then

$$\begin{aligned} &\mathbf{Pr}[\mathbb{I}_t = 0 \mid i \in D_t] \\ &= \mathbf{Pr}[\mathbb{I}_t = 0 \mid (i \in D_t) \text{ and } \mathcal{F}] \cdot \mathbf{Pr}[\mathcal{F}] + \mathbf{Pr}[\mathbb{I}_t = 0 \mid (i \in D_t) \text{ and } \neg\mathcal{F}] \cdot \mathbf{Pr}[\neg\mathcal{F}] \\ &\leqslant \mathbf{Pr}[\mathbb{I}_t = 0 \mid (i \in D_t) \text{ and } \mathcal{F}] + n^{-2} \leqslant \mathbf{Pr}[\mathbb{I}_t = 0 \mid (i \in D_t) \text{ and } \mathcal{F}] + 1/d. \end{aligned} \tag{14}$$

Let X be the random variable that counts the number of empty bins of a random $(d-1)$-element subset of $H_t \backslash \{i\}$, conditioned on the event that "$(i \in D_t)$ and $\mathcal{F}$" holds. Then X is a hypergeometric random variable with parameters $(s-1, K, d-1)$, where K is the number of empty bins contained in $H_t \setminus \{i\}$. Thus

$$\mathbf{E}[X] = \frac{(d-1)K}{s-1} \quad \text{and} \quad \mathbf{Var}[X] \leqslant \frac{(d-1)K}{s-1} \leqslant d.$$

Then $\mathbf{E}[X] > d/3$, since $K \geqslant s/2 - 1$ when $i \in D_t$ and $\mathcal{F}$ holds (and using the size property $s = \Omega(\log n)$) and the fact that $d \geqslant 7$). Therefore

$$\begin{aligned}
&\mathbf{Pr}[\mathbb{I}_t = 0 \mid (i \in D_t) \text{ and } \mathcal{F}] \\
&\leqslant \mathbf{Pr}[X < d/6] \leqslant \mathbf{Pr}[|X - \mathbf{E}[X]| \leqslant \mathbf{E}[X]/2] < \frac{4\,\mathbf{Var}[X]}{\mathbf{E}[X]^2} \leqslant \frac{36d}{d^2} = \frac{36}{d},
\end{aligned}$$

using Chebychev's inequality. Substituting the above upper bound in Inequality (14) establishes (13) with $\hat{c} = 38$, which completes the proof. □

F Proof of Proposition 13

In this subsection we will prove two lemmas and then combine them to establish the proposition. The lemmas and their proofs are inspired by [13, Lemma 2.1 and 2.2]. Recall that a subgraph of $\mathcal{C}_n$ is c-loaded if every vertex (bin) in the subgraph has load at least c.

Lemma 20. *Let k be a positive integer and let $c_1 > 0$. The probability that conflict graph $\mathcal{C}_n$ contains a c_1-loaded connected component with k vertices is at most*

$$n \cdot 8^k \cdot \left(\frac{2\mathrm{e}}{c_1}\right)^{c_1 k}.$$

Moreover, by setting $c_1 = 12(c+1)$, we conclude that with probability at least $1 - n^{-c}$, the conflict graph $\mathcal{C}_n$ does not contain a c_1-loaded tree with at least $\log n$ vertices.

Proof. A connected component in $\mathcal{C}_n$ with k vertices contains a spanning tree with k vertices. By Proposition 10, there are at most 4^{k-1} ordered trees with k vertices. For every ordered tree, we can choose its root in n ways, as we have n bins (vertices). Hence there are at most $n \cdot 4^{k-1}$ rooted and ordered trees. Let us fix an arbitrary ordered tree T with a specified root. Also let $(t_1, \ldots, t_{k-1})$ denote an arbitrary sequence of rounds, where $t_i \in \{1, \ldots, n\}$ is the round when the i-th edge of the ordered tree T is chosen. Notice that in an ordered tree with specified root, the i-th edge always connects the i-th child to its parent, and the parent is already known to us. Therefore, to build the tree, the i-th edge of the tree must be chosen from edges of $G^{(t_i)}$ that are adjacent to the known parent. This implies that the algorithm chooses the i-th edge of T in round t_i with probability $\frac{\Delta_{t_i}}{n\Delta_{t_i}/2} = \frac{2}{n}$. Since balls are independent from each other, the tree T is constructed at the given times $(t_1, \ldots, t_{k-1})$ with probability

$$\left(\frac{2}{n}\right)^{k-1}. \tag{15}$$

On the other hand, ball t is allocated to a given bin with probability at most $\Delta_t/(n\Delta_t/2) = 2/n$. Therefore, the probability that T is c_1-loaded is at most

$$\binom{n}{ck}\left(\frac{2k}{n}\right)^{c_1 k} \leqslant \left(\frac{\mathrm{e}n}{c_1 k}\right)^{c_1 k}\left(\frac{2k}{n}\right)^{c_1 k} = \left(\frac{2\mathrm{e}}{c_1}\right)^{c_1 k}, \tag{16}$$

where we used the fact that $\binom{n}{c_1 k} \leqslant \left(\frac{en}{c_1 k}\right)^{c_1 k}$. Since balls are independent, one can multiply (15) by (16) and derive an upper bound for the probability that T is constructed at the given times and is c-loaded. Taking the union bound over all rooted ordered trees and time sequences gives

$$
\begin{aligned}
n4^{k-1} \sum_{(t_1,\ldots,t_{k-1})} \left\{ \left(\frac{2}{n}\right)^{k-1} \left(\frac{2\mathrm{e}}{c_1}\right)^{c_1 k} \right\} &\leqslant n4^{k-1} n^{k-1} \cdot \left\{ \left(\frac{2}{n}\right)^{k-1} \left(\frac{2\mathrm{e}}{c_1}\right)^{c_1 k} \right\} \\
&= n8^{k-1} \left(\frac{2\mathrm{e}}{c_1}\right)^{c_1 k},
\end{aligned}
$$

proving the first statement of the lemma. By setting $c_1 = 12(c+1)$ and $k = \log n$ in the above formula, we infer that the probability that $\mathcal{C}_n$ contains a c_1-loaded tree with $\log n$ vertices is at most

$$
n8^{k-1} \left(\frac{2\mathrm{e}}{c_1}\right)^{c_1 k} < n2^{3k} 2^{-12(c+1)k} \leqslant n2^{-ck-9k} \leqslant n^{-c},
$$

completing the proof. □

Lemma 21. *Suppose that the conflict graph $\mathcal{C}_n$ contains a c-loaded k-vertex tree T, where $c > 4\mathrm{e}$ is any constant and k is a positive integer. Let p denotes the number of cycle-producing edges (with respect to T) which have been added between vertices in this tree during the allocation process. Then $p < 2(c+1)/\varepsilon$ with probability at least $1 - n^{-c}$.*

Proof. For a given connected component of k vertices, there are at most $\binom{k}{2}$ edges whose addition may produce a cycle. This includes edges already present in the component, as an edge with multiplicity 2 (double edge) forms a 2-cycle. Thus, the p edges can be chosen in $\binom{k}{2}^p < k^{2p}$ ways. Let $\{e_1, e_2, \ldots, e_p\}$ denote a set of p cycle-producing edges (some of these may create 2-cycles). Also let $(t_1, \ldots, t_p)$ denote a sequence of rounds, where $t_i \in \{1, \ldots, n\}$ is the round in which the t_i-th ball picks edge e_i. For each round $t = 1, 2, \ldots, n$ and $i = 1, \ldots, p$, let us define $\mathbb{I}_t(e_i)$ as follows:

$$
\mathbb{I}_t(e_i) = \begin{cases} 1 & \text{if } e_i \in E_t, \\ 0 & \text{otherwise.} \end{cases}
$$

It is easy to see that

$$
\mathbf{Pr}\left[\text{ball } t \text{ picks edge } e_i \text{ of } G^{(t)}\right] = \frac{\mathbb{I}_t(e_i)}{|E_t|}.
$$

Now $\mathtt{vis}(e_i) = \sum_{t=1}^{n} \mathbb{I}_t(e_i)$ for $i = 1, \ldots, p$. Using this, and the fact that $|E_t| \geqslant n/2$ for each t (since $G^{(t)}$ is regular with degree at most 1), the probability that $e_1, e_2, \ldots, e_p$ are chosen is at most

$$
\sum_{(t_1,\ldots,t_p)} \left\{ \prod_{i=1}^{p} \frac{\mathbb{I}_{t_i}(e_i)}{E_{t_i}} \right\} \leqslant \prod_{i=1}^{p} \left\{ \sum_{t=1}^{n} \frac{\mathbb{I}_t(e_i)}{E_t} \right\} \leqslant \prod_{i=1}^{p} \frac{4\,\mathtt{vis}(e_i)}{n} \leqslant \left(\frac{4n^{1-\varepsilon}}{n}\right)^p = \left(\frac{4}{n^{\varepsilon}}\right)^p. \tag{17}
$$

Moreover, applying Lemma 20 shows that the probability that $\mathcal{C}_n$ contains a c-loaded k-vertex tree is at most

$$
n \cdot 8^k \cdot \left(\frac{2\mathrm{e}}{c}\right)^{ck} \leqslant n \cdot 2^{-k}, \tag{18}
$$

as $c \geqslant 4\mathrm{e}$. So, with high probability, $\mathcal{C}_n$ does not contain any c-loaded tree with at least $(\log n)^2$ vertices. Now assume that $k < (\log n)^2$. Combining (17) and (18), and taking the union bound over all choices for a set of p edges, we find that the probability that a c-loaded k-vertex tree contains p cycle-producing edges is at most

$$k^{2p} \cdot \left(\frac{4}{n^{\varepsilon}}\right)^{p} \cdot n \cdot 2^{-k} = \left(\frac{4 \cdot k^2}{n^{\varepsilon}}\right)^{p} \cdot n \cdot 2^{-k} \leqslant n^{-\varepsilon p/2} \cdot n \cdot 2^{-k}, \tag{19}$$

where the inequality holds as $k < (\log n)^2$. Therefore the probability that $p = \lceil 2(c+1)/\varepsilon \rceil$ cycle-producing edges are present is at most n^{-c}. We conclude that $p < 2(c+1)/\varepsilon$ with probability at least $1 - n^{-c}$. $\square$

Proof of Proposition 13. Combining the Lemmas 20 and 21 establishes the proposition. $\square$