Balanced allocation on graphs: A random walk approach*

Ali Pourmiri1,2

1 Department of Computing, Macquarie University, Sydney, Australia
2 Department of Software Engineering, University of Isfahan, Isfahan, Iran

Correspondence
Ali Pourmiri, Department of Computing, Macquarie University, Sydney, Australia.
Email: alipourmiri@gmail.com

Abstract
We propose algorithms for allocating $n$ sequential balls into $n$ bins that are interconnected as a $d$-regular $n$-vertex graph $G$, where $d \geq 3$ can be any integer. In general, the algorithms proceeds in $n$ succeeding rounds. Let $\ell > 0$ be an integer, which is given as an input to the algorithms. In each round, ball $1 \leq t \leq n$ picks a node of $G$ uniformly at random and performs a nonbacktracking random walk of length $\ell$ from the chosen node and simultaneously collects the load information of a subset of the visited nodes. It then allocates itself to one of them with the minimum load (ties are broken uniformly at random). For graphs with sufficiently large girths, we obtain upper and lower bounds for the maximum number of balls at any bin after allocating all $n$ balls in terms of $\ell$, with high probability.

KEYWORDS
balls-into-bins models, balanced allocation, nonbacktracking random walks

1 INTRODUCTION

A standard balls-into-bins model is a process which randomly allocates $m$ balls into $n$ bins, where each ball picks $d$ bins independently and uniformly at random and the ball is then placed on a least-loaded bin in the set of $d$ choices. When $m = n$ and $d = 1$, it is well known that at the end of the 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. Azar, Broder, Karlin, and Upfal [4] showed that for the $d$-choice process, $d \geq 2$, provided ties are broken.

*A preliminary version of this paper appeared in the proceedings of the 22nd International Computing and Combinatorics Conference (COCOON’16).
randomly, the maximum load is $\frac{\log \log n}{\log d} + O(1)$. For a complete survey on the standard balls-into-bins process, we refer the reader to [13]. Many subsequent works consider the settings where the choice of bins are not necessarily independent and uniform. For instance, Vöcking [16] proposed an algorithm called *always-go-left* that uses an exponentially smaller number of choices and achieves a maximum load of $\frac{\log \log n}{d \phi_d} + O(1)$ whp, where $1 \leq \phi_d \leq 2$ is an specified constant. 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 placed on a least-loaded bin among the chosen bins and ties are broken asymmetrically.

In many applications, selecting any random set of choices is costly. For example, in peer-to-peer or cloud-based systems balls (jobs, items,...) and bins (servers, processors,...) are randomly placed in a metric space (e.g., $\mathbb{R}^2$) and the balls have to be allocated to bins that are close to them as it minimizes the access latencies. With regard to such applications, Byer, Considine, and Mitzenmacher [7] 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 $d$ chosen locations. The ball is then allocated to one of the $d$ bins with the 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 no a uniform distribution over the potential choices. Here, the authors showed the maximum load is $\frac{\log \log n}{\log d} + O(1)$ whp. Later on, Kenthapadi and Panigrahy [11] proposed a model in which bins are interconnected as a $\Delta$-regular graph, and each ball picks a random edge of the graph. It is then placed on the one of its endpoints with the smaller load. This allocation algorithm results in a maximum load of $\log \log n + O\left(\frac{\log n}{\log(\Delta/\log^8 n)}\right) + O(1)$. Peres, Talwar, and Wieder [14] also considered a similar model where the number of balls can be much larger than bins (i.e., $m \gg n$) and the graph is not necessarily regular. Then, they established upper bound $O(\log n/\sigma)$ for the gap between the maximum and the minimum loaded bin after allocating $m$ balls, where $\sigma$ is the edge expansion of the graph. Following the study of balls-into-bins with correlated choices, Godfrey [10] generalized the model, introduced by Kenthapadi and Panigrahy, such that each ball picks a random edge of a hypergraph in which every hyperedge is of size $\Omega(\log n)$ and satisfies some mild conditions. Then, he showed that the maximum load is a constant whp.

Recently, Bogdan, Sauerwald, Stauffer, and Sun [6] studied a model where each ball picks a random node and performs a local search from the node to find a node with the local minimum load, where it is finally placed on. They showed that when the graph is a constant degree expander, the local search guarantees a maximum load of $O(\log \log n)$ whp.

### 1.1 Our results

In this paper, we study balls-into-bins models, where each ball chooses a set of related bins. We propose allocation algorithms for allocating $n$ sequential balls into $n$ bins that are organized as a $d$-regular $n$-vertex graph $G$, where $d \geq 3$. A nonbacktracking random walk (NBRW) $W$ of length $l > 0$ starting from a node is a random walk in $l$ steps so that in each step the walker picks a neighbor uniformly at random and moves to that neighbor with an additional property that the walker never traverses an edge twice in a row. Further information about NBRWs can be found in [1] and [2]. Here we consider two families of $d$-regular graphs, namely *high-degree* graphs, where $d = \omega(\log n)$ and *low-degree* graphs where, $d = O(\log n)$. Define parameter

$$r_d := \begin{cases} 1 & \text{if } d = \omega(\log n), \\ \lceil 2 \log_{d-1} \log n \rceil & \text{otherwise}. \end{cases}$$
Also, let $\ell > 0$ be a given integer. Our allocation algorithm, denoted by $A^\ell_r$, is based on a random sampling of bins from the neighborhood of a given node in $G$ by a NBRW from the node. The algorithm proceeds as follows: for each ball $t$, $1 \leq t \leq n$, the ball picks a node of $G$ uniformly at random and performs a NBRW of length $\ell r_d$, say $(u_0, u_1, \ldots, u_{\ell r_d})$, from the chosen node. Then, a subset of visited nodes namely $\{u_{j r_d} \mid 0 \leq j \leq \ell\}$, called potential choices, is selected and finally the ball is allocated to a least-loaded node of the potential choices (ties are broken randomly). Our results concern bounding the maximum load attained by $A^\ell_r$, denoted by $m^*$, in terms of $\ell$. Note that for the rest of the paper we focus on NBRWs of sublogarithmic length $\ell \in [o(1), o(\log n)]$. Because, if the balls are allowed to take NBRWs of length $\ell = \Omega(\log n)$ on a graph with girth at least $\ell$, then the visited nodes by each ball generates a random hyperedge of size $\ell + 1$. Then applying the Godfrey’s result [10] implies a constant maximum load whp. To show the results, we assume that $G$ is a $d$-regular $n$-vertex graph with girth at least $\omega(\ell \log \log n)$ and $d = \omega(\log n)$. It is worth mentioning that there exist several explicit families of $d$-regular $n$-vertex graphs with arbitrary degree $d \geq 3$ and girth $\Omega(\log_d n)$ (e.g., see [9]). Let us first consider high-degree graphs (i.e., $d = \omega(\log n)$) on $n$ nodes. As $d = \omega(\log n)$, by the definition of $r_d$ we have $r_d = 1$. Now, in order to present the upper bound attained by $A^1_r$, we consider two cases in terms of $\ell$:

1. Let $\gamma_d = \sqrt{\log_d n}$ and $\ell \geq 4\gamma_d$ be a given integer. Also assume that $G$ is a $d$-regular $n$-vertex graph with girth $\omega(\ell \log \log n)$. Then we show that whp,

$$m^* = \Theta\left(\frac{\log \log n}{\log(\ell / \gamma_d)}\right).$$

Note that if we set $\ell = \lfloor (\log n)^{\frac{1}{1+\epsilon}} \rfloor$, for any constant $\epsilon \in (0, 1)$, $G$ with girth at least $\omega(\ell)$ suffices as well. Applying the above upper bound, we have $m^* = \Theta(1/\epsilon)$ whp.

2. Let $\omega(1) \leq \ell \leq 4 \cdot \gamma_d$, be a given integer. Also assume that $G$ is a $d$-regular $n$-vertex graph with girth $\omega(\ell \log \log n)$. Then we show that whp,

$$m^* = \Theta\left(\frac{\log_d n \cdot \log \log n}{\ell^2}\right).$$

In addition to the upper bound, we prove that whp,

$$m^* = \Omega\left(\frac{\log_d n}{\ell^2}\right)$$

(for a proof see Section 7). Besides these results for high-degree graphs (i.e., $d = \omega(\log n)$), provided $G$ has girth at least $\omega(\ell (\log \log n)^2)$, we show the upper and lower bounds for the maximum load attained by $A^\ell_r$ on low-degree graphs (i.e., $d = \Theta(\log n)$) that are similar to ones we presented for $A^1_r$ on high-degree graphs (see Section 6).

Notice that for the sake of analysis, we require that each potential choice is randomly chosen from a set of nodes of size $\omega(\log n)$ (see the proof of Lemma 2.1). When the underlying graph has degree $\omega(\log n)$, then each visited node by a NBRW satisfies the condition. However, when the graph has degree $\Theta(\log n)$ and sufficiently large girth, then every $r_d$th visited node is randomly chosen from a set of size $d(d - 1)^{r_d} = \omega(\log n)$ and satisfies the requirement. Also, note that one might be able to analyze a random-walk based algorithm on sparse graphs that consider all visited nodes as potential choices, however our proof technique cannot be applied for such an algorithm.
1.2 Comparison with related works

The setting of our work is closely related to [6]. In this paper, authors consider a model in which there are \( n \) sequential balls and \( n \) bins that are organized as a \( d \)-regular \( n \)-vertex graph. In each step a ball arrives and picks a node of the graph uniformly at random, called the *birthplace* of the ball. Then starting from its birthplace, the ball examines the load of its adjacent nodes, if there is a node with load strictly smaller than the load of its current location, then the ball moves from its current location to that node. The moves continue until there is no adjacent node to the current location of the ball with a smaller load. The algorithm is called the *local search*. They show that if the graph is a constant degree expander and the local search allocates \( n \) balls, then the maximum load is \( \Theta(\log \log n) \) whp.

To prove the upper bound \( O(\log \log n) \), they exploit the local expansion property of the graph, and by applying majorization and probabilistic techniques show that the maximum load cannot exceed the bound. In comparison to this result, our new algorithms are based on random sampling of the bins using a NBRW of length \( \ell \). Furthermore, our result suggests a trade-off between \( \ell \) and the maximum load.

We also show a constant upper bound for sufficient long walks (i.e., \( \ell = (\log n)^{1+\varepsilon} \), for any constant \( \varepsilon \in (0, 1) \)), which is an improvement over the maximum load attained by the local search. To analyze the algorithms, we restrict our underlying graphs to regular graphs with sufficiently large girth.

From a technical point of view, our work is related to the one by Kenthapadi and Panigrahy [11] where each ball picks a random edge from a \( n^{\Omega(1/\log \log n)} \)-regular graph and places itself on the one of endpoints of the edge with a smaller load. This model results in a maximum load of \( \Theta(\log \log n) \) and the authors apply the witness tree technique to prove their result.

Godfrey [10] considered balanced allocation on hypergraphs, where balls choose a random edge \( e \) of a hypergraph satisfying some conditions, that is, first the size of each edge, say \( s \), is \( \Omega(\log n) \) and \( \Pr[u \in e] = \Theta(\frac{1}{n}) \) for every bin \( u \). The latter one is called *balanced condition*. Berenbrink, Brinkmann, Friedetzky, and Nagel [5] simplified Godfrey’s proof and slightly weakened the balanced condition, but since both analysis apply a Chernoff bound, it seems unlikely that one can extend the analysis for hyperedges of size \( o(\log n) \). Our model can also be viewed as a balanced allocation on hypergraphs, because every \( \ell \)-walk is a random hyperedge of size \( \ell + 1 \) that also satisfies the balanced condition (see Lemma 8.2). By setting the right parameter for \( \ell \in [o(1), o(\log_d n)] \), we show that the algorithm achieves a constant maximum load with sublogarithmic number of choices. Vöcking [16] proposed an algorithm that for each ball it probes the load information of \( \log \log n \) random bins and allocates \( n \) balls into \( n \) bins with a constant maximum load. Here, the algorithm requires \( \log \log n(\log n) \) random bits for each ball. In general, our results offer a variation of balls-into-bins model with the following combination of properties:

1. The construction of the scheme is explicit (a graph \( G \) with the desired degree and girth properties can be constructed as in [9]).
2. Each ball uses only \((1 + o(1))(\log n)\) random bits.
3. Each ball chooses a set containing \( o(\log n) \) bins.
4. The maximum load is constant.

It seems that the previous works could only handle any three of the four properties.

In a different context, Alon and Lubetzky [2] showed that if a particle starts a NBRW of length \( n \) on an \( n \)-vertex regular expander graph with high-girth then the number of visits to nodes has a Poisson distribution. In particular, they showed that the maximum number of visits to a node is at most \((1 + o(1)) \cdot \frac{\log n}{\log \log n} \). Our result can also be seen as an application of the mathematical concept of NBRWs to the task allocation in distributed networks, where the goal is to minimize the maximum load imposed on each server (for a similar model, e.g., see [15]). In such a model, a set of processors are connected
over a network and the processors randomly receive requests in sequential rounds. Then, the processor performs a NBRW of length $\ell$ and allocates the request to a visited processor with the minimum load.

### 1.3 | Techniques

To derive a lower bound for the maximum load, we first show that whp there is a path of length $\ell$ which is picked by at least $\Omega\left(\log_d n / \ell^\delta\right)$ balls. Also, each path contains $\ell + 1$ choices and hence, by pigeonhole principle there is a node with load at least $\Omega\left(\log_d n / \ell^2\right)$, which is a lower bound for the maximum load $m^*$. We establish an upper bound for the maximum load, which is based on witness graph technique. In our model, the potential choices for each ball are highly correlated, so building the witness graph is somewhat different from the one for the standard balls-into-bins. Our technique applies a key property of the algorithm, called $(a, n_1)$-uniformity (see Definition 1). Let us assume that $n_1$ balls are allocated. Similar to the standard witness graph technique, we start from a node with load $m_1^*$, and hence it implies that there is an $\ell$-walk, say $R$, whose nodes have a load at least $m_1^* - 1$. Since every node in $R$ has load at least $m_1^* - 1$, we are able to select a set of $\ell$-walks, say $\{W_1, W_2, \ldots, W_k\}$, that intersect $R$ and each $W_i$ contains nodes with load at least $m_i^* - \rho$, where $\rho$ is a specified value. Similarly, we expand each $W_i$ to a set of new $\ell$-walks with desired properties. After a number of iterations, we get a collection of intersecting $\ell$-walks that contain $\Omega(\log n)$ distinct nodes in which every node has load at least $c$. The resulting witness structure is called the $c$-loaded $(\mu, \lambda)$-tree, where $\mu = \Omega(\log n)$ is the number of distinct nodes and $\lambda$ is the number of intersecting $\ell$-walks. (for a definition see Section 2). Now, by using the $(a, n_1)$-uniformity, we guarantee that whp such a $c$-loaded $(\mu, \lambda)$-tree for some $c = \Theta(1)$ does not exist. Putting these together implies that after allocating $n_1$ balls the maximum load is bounded by $m_1^*$. To show an upper bound for the maximum load, after allocating $n$ balls, we divide the allocation process into $n / n_1$ phases and prove that the maximum load at the end of each phase increases by at most $m_1^*$, and hence $m^* \leq (n / n_1)m_1^*$ whp.

### 1.4 | Discussion and open problems

In this paper, we proposed balls-into-bins model, where each ball picks a set of nodes that are visited by a NBRW of length $\ell$ and places itself on a visited node with the minimum load. One may ask whether it is possible to replace a NBRW of length $\ell$ by several parallel random walks of shorter length (started from the same node) and get the similar results?

In our result, we constantly use the assumption that the graph locally looks like a $d$-ary tree. It is also known that cycles in random regular graph are relatively far from each other (see, e.g., [8]), so we believe that our approach can be extended for balanced allocation on random regular graphs.

Many works in this area (see, e.g., [6, 11]) assumed that the underlying networks is regular, it would be interesting to investigate random walk-based algorithms for irregular graphs.

### 1.5 | Outline

In Section 2, we formally define $(a, n_1)$-uniformity and the witness graph structure, called $c$-loaded $(\lambda, \mu)$-tree, and show some preliminary results. In Section 3 we show how to construct a $c$-loaded $(\lambda, \mu)$-tree, which is our witness graph. In Section 4, we calculate the existing probability of a $c$-loaded $(\mu, \lambda)$-tree. In Section 5, by applying the results from Sections 3 and 4, we show an upper bound for the maximum load. In Section 6 we present allocation algorithm $A_{\ell}^\alpha$ for $d$-regular graphs with $d = \Theta(\log n)$ and give a proof sketch for the results in this section. In Section 7 we prove a lower bound for the maximum load attained by the algorithms. Finally, in Section 8, we provide the missing proofs.
2 | NOTATION, DEFINITIONS, AND PRELIMINARIES

To analyze allocation algorithm $A_{r_d}^\ell$ on $d$-regular graphs, in this section we provide notation, definitions, and some preliminary results. In particular, we formally define the notation of $c$-loaded $(\mu, \lambda)$-trees.

A NBRW $W$ of length $\ell' > 0$ starting from a node is a random walk in $\ell'$ steps so that in each step the walker picks a neighbor uniformly at random and moves to that neighbor with an additional property that the walker never traverses an edge twice in a row. Recall that for every $d$-regular $n$-vertex graph $G$, where $d \geq 3$. We have,

$$r_d := \begin{cases} 1 & \text{if } d = \omega(\log n), \\ \lceil 2 \log_{d-1} \log n \rceil & \text{otherwise}. \end{cases}$$

Also assume that $\ell > 0$ is a given integer. Our allocation algorithm on $G$, denoted by $A_{r_d}^\ell$, proceeds as follows: for each ball $t$, the ball picks a node of $G$ uniformly at random and performs a NBRW of length $\ell \cdot r_d$, say $(u_0, u_1, \ldots, u_{\ell \cdot r_d})$, from the chosen node. Then a subset of visited nodes namely $\{u_j \cdot r_d | 0 \leq j \leq \ell\}$, called potential choices, is selected and finally the ball is allocated to a least-loaded node of potential choices (ties are broken randomly). Note that for the sake of simplicity we abbreviate $A_{1}^\ell$ as $A_1^\ell$.

2.1 | $(\alpha, n_1)$-Uniformity of the algorithms

In this subsection, we define the notion of $(\alpha, n_1)$-uniformity and present our key lemma whose proof can be found in Section 8.

**Definition 1.** Suppose that $B$ is an algorithm that sequentially allocates $n$ balls into $n$ bins. Let for some $1 \leq t \leq n$, $X_1, \ldots, X_t$ denote the random variables such that $X_i, 1 \leq i \leq t$, is the bin to which the $i$th ball is allocated. Then we say $B$ is an $(\alpha, n_1)$-uniform, if for every $1 \leq t \leq n_1 - 1$ and every bin $u$ we have

$$\Pr[\text{ball } t + 1 \text{ is allocated to } u \text{ by } B \mid X_1 = v_1, \ldots, X_t = v_t] \leq \frac{\alpha}{n},$$

where $v_i$'s are the outcome of allocating $t$ balls into $t$ bins by $B$ and $\alpha$ is some constant.

**Lemma 2.1 (Key Lemma).** Suppose that $\ell$ is a given integer and $G$ is a $d$-regular $n$-vertex graph with girth at least $\omega(\ell')$. Also define $r_d = 1$, for every $d = \omega(\log n)$, and $r_d = \lceil 2 \log_{d-1} \log n \rceil$ otherwise. Then, there exists some constant $\alpha$ for which allocation algorithm $A_{r_d}^\ell$ on nodes of $G$ is $(\alpha, n_1)$-uniform, where

$$n_1 = \lfloor n/(6e\alpha) \rfloor.$$  \hspace{1cm} (1)

2.2 | $c$-Loaded $(\mu, \lambda)$-tree

To analyze the allocation algorithm $A_\ell$ (i.e., $A_1^\ell$) on high-degree graphs, we need to define the notion of $c$-loaded $(\mu, \lambda)$-tree.

**Remark 1.** Note that during the execution of algorithm $A_\ell$, for each ball $t$, $1 \leq t \leq n$, $W_t$ denotes the $t$th $\ell'$-walk chosen by the algorithm, which is a path of length $\ell$. We abuse notation $W_t$ and sometimes
refer to it as an \((\ell + 1)\)-element subset of \(V(G)\), visited by the \(r\)th ball. Hence, it is well-defined to have union and intersection of two or more \(W_i\)‘s.

**Definition 2** (Interference graph). For every given pair \((G, \ell)\), the interference graph \(I(G, \ell)\) is defined as follows: The vertex set of \(I(G, \ell)\) is the set of all \(\ell\)-walks in \(G\). Two vertices, say \(W\) and \(W'\), are connected if and only if \(W \cap W' \neq \emptyset\). Note that if pair \((G, \ell)\) is clear from the context, then the interference graph is denoted by \(I\).

**Definition 3.** Let \(1 \leq n_1 \leq n\) be a given integer. Also, assume that \(A_\ell\) has allocated balls until the \((n_1 + 1)\)th ball. We then define \(H_{n_1}(G, \ell)\) to be a subgraph of \(I(G, \ell)\) induced by \(\{W_i : 1 \leq i \leq n_1\} \subset V(I)\). If pair \((G, \ell)\) is clear from the context, we use \(H_n\) to denote the graph.

**Definition 4** \((c\text{-loaded}) (\lambda, \mu)\)-tree). Let \(\lambda\) and \(\mu\) be given positive integers. We say tree \(T \subset I(G, \ell)\) is a \((\lambda, \mu)\)-tree if \(T\) is a rooted tree satisfying:

1. \(|V(T)| = \lambda\),
2. \(|\cup_{W \in V(T)} W| \geq \mu\).

Note that \(V(T)\) is the vertex set of \(T\) and the latter condition is well-defined because \(T \subset I(G, \ell)\) and every vertex \(W \in V(T)\) is an \((\ell + 1)\)-element subset of \(V(G)\). A \((\lambda, \mu)\)-tree \(T\) is called \(c\text{-loaded}\), if, for some given \(1 \leq n_1 \leq n\), \(T\) is contained in \(H_{n_1}\) and every node in \(\cup_{W \in V(T)} W\) has load at least \(c\).

We now show some graphical properties of interference graph \(I(G, \ell)\).

**Lemma 2.2.** Suppose that \(G\) is a \(d\)-regular \(n\)-vertex graph with girth at least \(\omega(\ell)\). Then,

1. \(|V(I(G, \ell))| = nd(d - 1)^{\ell - 1}\),
2. \(\Delta(I(G, \ell)) \leq (\ell + 1)^2d(d - 1)^{\ell - 1}\),

where \(V(I)\) and \(\Delta(I)\) denote the vertex set and the maximum degree of \(I(G, \ell)\), respectively. Furthermore, the number of rooted \(\lambda\)-vertex trees contained in \(I\) is bounded by \(4^\lambda \cdot |V(I)| \cdot \Delta(I)^{\lambda - 1}\).

**Proof.** \(G\) has girth at least \(\omega(\ell)\), so the number of \(\ell\)-walks is exactly \(nd(d - 1)^{\ell - 1}\), and hence \(|V(I)| = nd(d - 1)^{\ell - 1}\). Since the graph locally looks like a \(d\)-ary tree, the total number of \(\ell\)-walks including an arbitrary node \(v\) as the \(j\)th visited node, for every \(0 \leq j \leq \ell\), is

\[d(d - 1)^{\ell - 1}(d - 1)^{\ell - j} = d(d - 1)^{\ell - 1}.
\]

Index \(j\) varies from 0 to \(\ell\), so \(v\) can be a node of at most \((\ell + 1)d(d - 1)^{\ell - 1}\) \(\ell\)-walks. Also, every \(\ell\)-walk contains \(\ell + 1\) nodes, and hence every \(\ell\)-walk intersects at most \((\ell + 1)^2d(d - 1)^{\ell - 1}\) other \(\ell\)-walks. Thus we get

\[\Delta(I) \leq (\ell + 1)^2 \cdot d(d - 1)^{\ell - 1}.
\]

Let us now bound the total number of rooted \(\lambda\)-vertex trees contained in \(I\). Note that two rooted trees have different shapes if they are not isomorphic. The total number of different shape rooted trees on \(\lambda\) vertices is exactly the number of ordered trees on \(\lambda\) vertices, which is known to be the \((\lambda - 1)\)th Catalan number, which is

\[\frac{1}{\lambda} \left(\frac{2\lambda - 2}{\lambda - 1}\right) < \frac{2^{2\lambda - 2}}{\lambda} < 4^\lambda.
\]
(For example see [12]). For any given shape, there are $|V(I)|$ ways to choose the root. As soon as the root is fixed, each vertex in the first level can be chosen in at most $\Delta(I)$ many ways. By selecting the vertices of the tree level by level, we have that each vertex except the root can be chosen in at most $\Delta(I)$ ways. So the total number of rooted $\lambda$-vertex trees in $I$ is bounded by

$$4^\lambda \cdot |V(I)| \cdot \Delta(I)^{\lambda - 1}.$$ 

\[\blacksquare\]

**Corollary 2.3.** The number of $(\lambda, \mu)$-trees contained in $I(G, \ell)$ is bounded by $4^\lambda |V(I)| \Delta(I)^{\lambda - 1}$.

**Proof.** We know that every $(\lambda, \mu)$-tree $T$ is a rooted $\lambda$-vertex tree contained in $I$ with the additional property that

$$|\bigcup_{W \in V(T)} W| \geq \mu.$$ 

This implies that the number of rooted $\lambda$-vertex trees contained in $I$ is an upper bound for the number of $(\lambda, \mu)$-trees contained in $I$. Thus, by applying Lemma 2.2, we obtain the upper bound $4^\lambda |V(I)| \Delta(I)^{\lambda - 1}$.

\[\blacksquare\]

### 3 | WITNESS GRAPH

In this section, we show that if $A_\ell$ has allocated at most $n_1 = \Theta(n)$ balls and there is a node whose load is larger than a critical value, then we can construct a $c$-loaded $(\lambda, \mu)$-tree contained in $H_n \ (G, \ell)$, which is our witness graph. This section is organized as follows:

In Section 3.1, we present a 2-step procedure, called Partition-Branch, which proceeds as follows. For given $\ell$-walk $W$, the procedure takes $W$ as an input. It first partitions $W$ into a set of edge-disjoint subpaths (Partition step). Now, for each subpath, it possibly finds another $\ell$-walk chosen by the algorithm which intersects the subpath with some desired properties (Branch step). We say Partition-Branch is valid, if the Branch step can successfully find a desired $\ell$-walk for each subpath.

Afterward, we define event $\mathcal{N}_\delta$ (see Definition 5) and claim that if $A_\ell$ has allocated at most $n_1$ balls and this event holds, then for a given $\ell$-walk chosen by the algorithm, Partition-Branch is valid on the $\ell$-walk. We state and show the claim in Lemma 3.1. In Section 3.2, assuming there is a node whose load is larger than a specific value, we iteratively apply the procedure to build a $c$-loaded $(\lambda, \mu)$-tree. In Lemma 3.2, we show how the iterative application of the procedure will result in a $c$-loaded $(\lambda, \mu)$-tree with specified parameters. Before, we present the results of this section, let us state some notation and facts.

Note that $W$ denotes both an $\ell$-walk (a path of length $\ell$) and the set of nodes contained in the $\ell$-walk with specified loads. Let us define

$$f(W) = \min_{u \in W} \text{load}(u),$$

where load($u$) indicates the number of balls allocated to node $u$. Also, note that since our graph has girth at least $\omega(\ell)$, every path of length at most $\ell$ in the graph can be uniquely determined by its endpoints. Thus, we use closed interval $[u, v]$ to represent a path whose endpoints are $u$ and $v$. By the girth condition, one can easily see that the intersection of two paths of length at most $\ell$ can be only one of these three cases: (1) a single node, (2) a path of length at most $\ell$, (3) an empty graph. The height of a ball allocated to a node is the number of balls that are allocated to the node before that ball.
3.1 | Partition-Branch procedure

In this subsection we formally define the Partition-Branch procedure, event $\mathcal{N}_\delta$, and state a lemma regarding to them.

3.1.1 | Partition-Branch

Let $k \geq 1$ and $\rho \geq 1$ be given integers and $W$ be an $\ell$-walk with $f(W) \geq \rho + 1$. The Partition-Branch procedure on $W$ with parameters $\rho$ and $k$, denoted by $PB(\rho, k)$, proceeds as follows:

Partition: It partitions $W$ into $k$ edge-disjoint subpaths:

$$P_k(W) = \{ [u_i, u_{i+1}] \subseteq W, 0 \leq i \leq k - 1 \},$$

where $[u_i, u_{i+1}]$ is a subpath of length either $\lfloor \ell/k \rfloor$ or $\lceil \ell/k \rceil$.

Branch: In this step, for given subpath $P_i = [u_i, u_{i+1}] \in P_k(W)$, the Branch step finds (if exists) an $\ell$-walk chosen by $A_\ell$ to allocate a ball, say $W_{P_i}$, that satisfies the following conditions:

(C1) $W_{P_i}$ intersects $P_i = [u_i, u_{i+1}] \in P_k(W)$ but $W_{P_i}$ does not contain $u_i$ and $u_{i+1}$.

(C2) Every node in $W_{P_i}$ contains at least $f(W) - \rho$ balls (i.e., $f(W_{P_i}) \geq f(W) - \rho$).

We say procedure $PB(\rho, k)$ on a given $\ell$-walk, say $W$, is valid, if for every $P \in P_k(W)$, $W_P$ exists. We refer to $W$ as the father of $W_P$. For a graphical view of the Partition-Branch procedure see Figure 1.

Definition 5 (Event $\mathcal{N}_\delta$). Suppose that $W = \{W_1, W_2, \ldots, W_n\}$ be the set of $\ell$-walks that are chosen by $A_\ell$ for allocating $n_1$ balls. Then, for any given $1 \leq \delta \leq \ell$, we say that event $\mathcal{N}_\delta$ holds, if every path of length $\delta$ in $G$ is contained in less than $(6 \log_d n) / \delta$ $\ell$-walks in $W$.

To state the following lemma, let us define a set of parameters depending on $d$, $n$, and $\ell$.

$$k := \max\{4, \lfloor \ell / \sqrt{\log_d n} \rfloor \},$$

$$\delta := \lceil \lfloor \ell/k \rfloor / 4 \rceil,$$

$$\rho := \lceil 6 \log_d n / \delta^2 \rceil.$$

Lemma 3.1. Assume that $G$ is a $d$-regular $n$-vertex graph with girth $o(\ell)$, where $\ell \in [o(1), o(\log_d n)]$. And, $A_\ell$ has allocated at most $n_1$ balls on $G$. Suppose that $W$ be an $\ell$-walk chosen by the algorithm for some ball with $f(W) \geq \rho + 1$ and $\mathcal{N}_\delta$ holds. Then $PB(\rho, k)$ on $W$ is valid.

Proof. For given $W$, let us apply the Partition step with parameter $k$. Then we have a set of $k$ subpaths

$$P_k(W) = \{ [u_i, u_{i+1}] \subseteq W, 0 \leq i \leq k - 1 \},$$
where \( d(u_i, u_{i+1}) \in \{[\ell/k], [\ell/k] \} \) and \( d(u_i, u_{i+1}) \) denotes the length of a shortest path between \( u_i \) and \( u_{i+1} \). Let us fix an arbitrary subpath \( P_i = [u_i, u_{i+1}] \in P_k(W) \). Again, we partition \( P_i = [u_i, u_{i+1}] \) into 3 edge-disjoint subpaths, say \([u_i, u], [u, v] \) and \([v, u_{i+1}] \) such that

\[
d(u, u) = d(v, u_{i+1}) = \delta,
\]

where \( \delta = \lfloor \ell/k \rfloor \), defined in (2). Since \( d(u_i, u_{i+1}) \in \{[\ell/k], [\ell/k] \} \), we have \( d(u, u) \geq 4\delta \) and hence

\[
d(u, v) = d(u_i, u_{i+1}) - 2\delta \geq 4\delta - 2\delta = 2\delta.
\] (3)

So \([u, v]\) is a subpath of \( W \) whose length is at least \( 2\delta \). Define \( S \) to be the set of all nodes contained in path \([u, v]\). Also let \( B(S) \) denotes the set of all balls allocated to nodes of \( S \) at height at least \( f(W) - \rho \). Consider arbitrary ball \( t \in B(S) \), which was allocated to a node of \( S \), say \( w \). This implies that \( W_t \), chosen \( \ell' \)-walk by this ball, shares \( w \) with \( S \subset W \) and hence \( W_t \) intersects \( W \). On the other hand, by the definition of \( B(S) \), ball \( t \in B(S) \) has height at least \( f(W) - \rho \), which implies that \( f(W_t) \geq f(W) - \rho \geq 1 \), otherwise ball \( t \) was allocated to a node in \( W_t \), say \( w' \), with load \( \omega(w') < f(W) - \rho \). Thus, for every \( t \in B(S), W_t \) satisfies (C2) (i.e., \( f(W_t) \geq f(W) - \rho \)) and also \( W_t \) intersects \( P_i \subset W \). Now, it only remains to show there exists an \( \ell' \)-walk in \([W_t | t \in B(S)]\) satisfying (C1). Since \( S \subset W \) and \( f(W) \geq \rho + 1 \), each node of \( S \) has \( \rho \) balls at height at least \( f(W) - \rho \geq 1 \). Therefore, by Inequality (3) we have

\[
|B(S)| \geq |S|\rho \geq (2\delta)\rho \geq (2\delta)(6\log_{d-1} n/\delta^2)
\]

\[
= 12\log_{d-1} n/\delta.
\]

By using the above inequality we have,

\[
|\{W_t | t \in B(S)\}| = |B(S)| \geq 12\log_{d-1} n/\delta.
\]

Recall that \( P_i = [u_i, u] \cup [u, v] \cup [v, u_{i+1}] \) and, for every \( t \in B(S) \), \( W_t \) intersects \([u, v]\). If for some \( t \in B(S), W_t \) contains \( u_i \) (or \( u_{i+1} \)), then it must contain subpath \([u_i, u] \) (or \([v, u_{i+1}] \)), because \( W_t \) intersects \([u, v] \) and \( G \) has girth \( \omega(\ell') \). Conditioning on \( \mathcal{N}_\delta \) implies that \([u_i, u] \) and \([v, u_{i+1}] \) are contained in less than \( (12\log_{d-1} n)/\delta \) \( \ell' \)-walks. So the above inequality shows that there is at least one ball, say \( t_0 \in B(S) \), whose corresponding \( \ell' \)-walk \( W_{t_0} \) contains neither \( u_i \) nor \( u_{i+1} \) and thus it satisfies (C1). Therefore, we conclude that, for each \( P_i \in P_k(W), W_{P_i} \) satisfying (C1) and (C2) exists and hence \( PB(\rho, k) \) on \( W \) is valid.

3.2 Construction of witness graph

In this subsection, we first define event \( U_{n, \ell, h} \). Then we present a recursive method that utilizes the Partition-Branch procedure and build a \( c \)-loaded \((\lambda, \mu)\)-tree contained in \( H_{n_1}(G, \ell) \). Finally in Lemma 3.2 we prove that the method outputs a witness graph with specified values \( c, \lambda \) and \( \mu \).

**Definition 6 (Event \( U_{n, \ell, h} \)).** Suppose that \( A_\ell \) has allocated \( n_1 \leq n \) balls, where \( \ell \in [\omega(1), o(\log_d n)] \). Then, define \( U_{n, \ell, h} \) to be the event that a node with load at least \( hp + c + 1 \) exists, where \( c = O(1) \) and \( h = \Omega(\log \log n) \) are positive integers that will be fixed later.

In the following, we show how to recursively construct the \( c \)-loaded \((\lambda, \mu)\)-tree. The construction is based on an iterative application of \( PB(\rho, k) \) and Lemma 3.1.
3.2.1 | Recursive construction

Let $\delta$ be the parameter defined in (2) and $\ell \in [\omega(1), o(\log d n)]$ be a given integer. Suppose that event $U_{n, \ell, \delta} \cap \mathcal{N}_\delta$ holds and $G$ is a regular graph with girth at least $\omega(\ell)$. This implies that there is at least one ball at height $h_p + c$ and thus there is an $\ell$-walk $R$, called root, so that $f(R) \geq h_p + c$. Applying Lemma 3.1 shows that $PB(\rho, k)$ on $R$ is valid, which means for every $P \in \mathcal{P}_k(R)$ there is an $\ell$-walk, say $W_p$, chosen by the algorithm satisfying (C1) and (C2). So, let us define

$$\mathcal{L}_1 := \{W_p, P \in \mathcal{P}_k(R)\},$$

which we call it the first level of our $c$-loaded $\langle \lambda, \mu \rangle$-tree. One may note that $R$ is the father of all $\ell$-walks in $\mathcal{L}_1$. Condition (C2) in the Partition-Branch procedure ensures that for every $W \in \mathcal{L}_1$,

$$f(W) \geq (h - 1)\rho + c.$$

Once we have the first level, we recursively build the $i$th level from the $(i - 1)$th level, for every $2 \leq i \leq h$. We know that each $W$ in level $\mathcal{L}_{i-1}$, $2 \leq i \leq h$, is created by the Branch step on its father. Let us fix arbitrary $\ell$-walk $W \in \mathcal{L}_{i-1}$ and its father $W'$. Let us apply the Partition step on $W$ and get $\mathcal{P}_k(W)$. We say $P \in \mathcal{P}_k(W)$ is a free subpath of $W$ if it does not share any node with its father (i.e., $P \cap W' = \emptyset$). We now show that $\mathcal{P}_k(W)$ contains at least $k - 2$ free subpaths. For some $0 \leq i \leq k - 1$, $W$ is created by the Branch step on $[u_i, u_{i+1}] \in \mathcal{P}_k(W')$ and also $W$ does not contain $u_i$ and $u_{i+1}$ (by (C1)). By the girth condition on $G$, the intersection of $W$ and $W'$ is only a subpath of $[u_i, u_{i+1}]$, say $[u, v]$. Thus,

$$d(u, v) < d(u_i, u_{i+1}) \leq \lceil \ell / k \rceil.$$

On the other hand, the length of each subpath in $\mathcal{P}_k(W)$ is at most $\lceil \ell / k \rceil$. So we conclude that $[u, v] \in W$ shares node(s) with at most $2$ subpaths in $\mathcal{P}_k(W)$. Therefore, $\mathcal{P}_k(W)$ contains at least $k - 2$ free subpaths. Let $\mathcal{P}^0_k(W) \subset \mathcal{P}_k(W)$ denote an arbitrary set of $k - 2$ free subpaths. By (C2) and the recursive construction, we have that for each $W \in \mathcal{L}_{i-1}$, $f(W) \geq (h - i + 1)\rho + c$. Therefore, by Lemma 3.1, $PB(\rho, k)$ on $W$ is valid. Therefore, we can define the $i$th level as follows,

$$\mathcal{L}_i = \bigcup_{W_p \in \mathcal{L}_{i-1}} \{W_p, P \in \mathcal{P}^0_k(W)\},$$

where each $W_p$ is the result of the Branch step on $P$. For a graphical view see Figure 2. The following lemma guarantees that our construction gives a $c$-loaded $\langle \lambda, \mu \rangle$-tree with desired parameters whose proof can be found in Section 8.2.
Lemma 3.2. Suppose that $\ell' \in [\omega(1), o(\log_d n)]$ be a given integer and, for some $h = O(\log \log n)$, $G$ be a $d$-regular $n$-vertex graph with girth at least $10h\ell'$. Also, let $k$ and $\delta$ be the parameters defined in (2). Assume that after allocating at most $n_1 \leq n$ balls on $G$ by $A_{\ell'}$, $U_{n_1,\ell',\delta} \cap N_{\delta}$ holds. Then, there exists a $c$-loaded $(\lambda, \mu)$-tree $T \subset H_{n_1}(G, \ell')$, where $c > 2$ is any constant, $\lambda = 1 + k \sum_{j=0}^{h-1} (k - 2)^j$, and $\mu = (\ell' + 1) \cdot k(k - 2)^{h-1}$.

4 | APPEARANCE PROBABILITY OF A C-LOADED $(\lambda, \mu)$-TREE

In this section, we derive an upper bound for the probability that for some $n_1 = \Theta(n)$, a $c$-loaded $(\lambda, \mu)$-tree appears in $H_{n_1}$. Recall that $H_{n_1}$ is a subgraph of $I(G, \ell')$ induced by $\{W_1, W_2, \ldots, W_{n_1}\}$, where $W_t, 1 \leq t \leq n_1$, is the $\ell'$-walk chosen by $A_{\ell'}$ for the $t$th ball. Our proof technique is inspired by [11, Lemma 2.1].

Lemma 4.1. Assume that $\ell' \in [\omega(1), o(\log_d n)]$, $\mu$, $\lambda$ and $c$ are positive integers, where $\mu \geq \lambda$. Also, let $G$ be a $d$-regular $n$-vertex graph with girth at least $o(\ell')$. If $A_{\ell'}$ is an $(\alpha, n_1)$-uniform allocation algorithm, then, after allocating $n_1$ balls, the probability that a $(c+1)$-loaded $(\lambda, \mu)$-tree appears in $H_{n_1}(G, \ell')$ is at most

$$n \cdot \exp(4\lambda \log(\ell' + 1) - c\mu).$$

Proof. For each ball $1 \leq t \leq n_1$, $A_{\ell'}$ performs a NBRW of length $\ell'$ starting from a randomly chosen node. Hence, for each ball, $A_{\ell'}$ in fact samples an $\ell'$-walk uniformly at random. So let us interpret allocation algorithm $A_{\ell'}$ on $G$ as follows: For every ball $1 \leq t \leq n_1$, the ball picks a vertex of $I(G, \ell')$, say $W_t$, uniformly at random and then allocates itself to a least-loaded node contained in $W_t$, in which ties are broken randomly (see Definition 2). To have a $(c+1)$-loaded $(\lambda, \mu)$-tree, we have to allocate at least $c + 1$ balls to each node contained in $\bigcup_{W \in V(T)} W$. We may assume that for some integer $q \geq 0$, $|\bigcup_{W \in V(T)} W| = \mu + q \geq \mu$ and thus, in total we have to allocate at least $(c + 1)(q + \mu)$ balls to $\mu + q$ nodes. Let us divide the process of allocating balls in two phases: We first use $\lambda$ balls to build a $(\lambda, \mu)$-tree and then in the second phase we use remaining $(c + 1)(\mu + q) - \lambda$ balls to increase the loads to have a $(c + 1)$-loaded $(\lambda, \mu)$-tree.

4.1 | First phase

Let $p_1$ be the probability that using $\lambda$ balls, a $(\lambda, \mu)$-tree is built. Let us fix an arbitrary $(\lambda, \mu)$-tree $T \subset I(G, \ell')$. There are at most $n_1 \leq n$ ways to choose one ball per vertex of $T$ and hence at most $n_1^{\lambda} \leq n^{\lambda}$ ways to choose $\lambda$ balls that are going to pick the vertices of $T$. On the other hand, every ball picks a given vertex of $T$ with probability $1/|V(I)|$. Thus we get,

$$p_1 \leq n^{\lambda} \cdot (1/|V(I)|)^{\lambda},$$

4.2 | Second phase

To have a $(c+1)$-loaded $(\lambda, \mu)$-tree, we must allocate at least $(c + 1)(\mu + q) - \lambda$ balls to bins in $\bigcup_{W \in V(T)} W$. By the lemma assumption we have $\mu \geq \lambda$ and thus

$$(c + 1)(\mu + q) - \lambda \geq (c + 1)(\mu + q) - \mu = c(\mu + q) + q \geq c(\mu + q).$$
This means we have to allocate at least \( c(\mu + q) \) balls to bins in \( \cup_{W \in V(T)} W \). There are at most \( \binom{n_1}{c(\mu + q)} \) ways to choose \( c(\mu + q) \) balls and then the algorithm must allocate them to \( \mu + q \) bins. By Lemma 2.1, \( A_{c} \) is \((\alpha, n_1)\)-uniform and hence, using the union bound over all \( \mu + q \) nodes yields that a node in \( \cup_{W \in V(T)} W \) gets a ball with probability at most \( \frac{\alpha(\mu + q)}{n} \). Since balls are mutually independent, \( c \cdot (\mu + q) \) balls are allocated to \( \mu + q \) nodes with probability at most

\[
\left( \frac{\alpha \cdot (\mu + q)}{n} \right)^{c(\mu + q)}
\]

Let \( p_2 \) denote the probability that \( c(\mu + q) \) balls are allocated to nodes in \( \cup_{W \in V(T)} W \). Thus, we get

\[
p_2 \leq \sum_{q=0}^{\infty} \left( \frac{n_1}{c \cdot (\mu + q)} \right) \left( \frac{\alpha \cdot (\mu + q)}{n} \right)^{c(\mu + q)}
\]

\[
\leq \sum_{q=0}^{\infty} \left( \frac{e \cdot n_1}{c \cdot (\mu + q)} \right)^{c(\mu + q)} \cdot \left( \frac{\alpha \cdot (\mu + q)}{n} \right)^{c(\mu + q)}
\]

\[
\leq \sum_{q=0}^{\infty} \left( \frac{n_1 \cdot e}{n \cdot c} \right)^{c(\mu + q)} = (1/6c)^{\mu} \sum_{q=0}^{\infty} (1/6c)^{cq}
\]

\[
\leq 2 \cdot (1/6c)^{\mu},
\]

where we use the fact that for integers \( 1 \leq a \leq b, \left( \frac{b}{a} \right) \leq \left( \frac{b}{a} \right)^{a} \) and the last inequality follows from \( \sum_{q=0}^{\infty} (1/6c)^{cq} \leq 2 \). The balls that are responsible for the loads of nodes in \( \cup_{W \in V(T)} W \) are different from the balls that build the \((\lambda, \mu)\)-tree. Moreover, balls are mutually independent. So \( p_1 \cdot p_2 \) is an upper bound for the probability that \((c + 1)\)-loaded \((\lambda, \mu)\)-tree \( T \) appears in \( H_{n_1} \). By Corollary 2.3, we have an upper bound for the number of \((\lambda, \mu)\)-trees contained in \( I(G, \ell) \). Hence, taking the union bound over all \((\lambda, \mu)\)-trees gives an upper bound for the appearance probability of a \((c + 1)\)-loaded \((\lambda, \mu)\)-tree in \( H_{n_1} \subset I(G, \ell) \). Thus we get,

\[
4^{\ell} |V(I)| \cdot \Delta(I)^{\ell-1} \cdot p_1 \cdot p_2 \leq 2 \cdot 4^{\ell} |V(I)| \cdot \Delta(I)^{\ell-1} \left( \frac{n}{V(I)} \right)^{\lambda} \cdot \left( \frac{1}{6c} \right)^{\mu}
\]

\[
\leq 2n^4 \cdot 4^\lambda \cdot \left( \frac{\Delta(I)}{V(I)} \right)^{\ell-1} \cdot \left( \frac{1}{6} \right)^{\mu}.
\]

By Lemma 2.2, we have \( |V(I)| = nd(d-1)^{\ell-1} \) and \( \Delta(I) \leq (\ell + 1)^2d(d-1)^{\ell-1} \). So the above bound is simplified as follows,

\[
2n \cdot 4^\lambda \left( (\ell + 1)^2 \right)^{\ell-1} \left( \frac{1}{6c} \right)^{\mu} \leq n(\ell + 1)^{4\lambda} 6^{-\mu} \leq n \exp(4\lambda \log(\ell + 1) - c\mu),
\]

where the first inequality follows from

\[
2 \cdot 4^\lambda \cdot (\ell + 1)^{2(\ell-1)} = 8^\lambda (\ell + 1)^{2(\ell-1)} \leq (\ell + 1)^{4\lambda},
\]

which is true for every \( \ell \geq 2. \)
5 \hspace{1cm} \textbf{BALANCED ALLOCATION ON HIGH-DEGREE GRAPHS}

In this section we show the upper bound for the maximum load attained by $A_\ell$ for $d$-regular graph with $d = \omega(\log n)$ (Theorem 5.2). For a given $G$ and $\ell' \in [\omega(1), \omega(\log_d n)]$, let us recall the set of parameters as follows,

\begin{align*}
k &:= \max\{4, \lfloor \ell' / \sqrt{\log_d n} \rfloor \}, \\
\delta &:= \lfloor \ell' / k \rfloor / 4, \\
\rho &:= [8 \log_d n / \delta^2], \\
h &:= \left\lfloor \log \log n \right\rfloor \left(\log (k - 2) \right) / (\log k - 2).
\end{align*}

(4)

Recall that $U_{n,\ell,h}$ is the event that at the end of allocating $n_1$ balls, there is a nodes with load at least $h\rho + c + 1$, where $c$ is a constant. Also, $\mathcal{N}_\delta$ denotes the event that after allocating at most $n_1$ balls no path of length $\delta$ appears in $(6 \log_{d-1} n) / \delta$ $\ell'$-walks chosen by $A_\ell$ (see Definition 5). Note that when $\ell' = (\log n)^{1/2}$ with constant $\epsilon \in (0, 1)$, then

$$k = \lfloor \ell' / \sqrt{\log_d n} \rfloor \geq \ell' / \sqrt{\log_3 n} \geq (\log n)^{\ell' / 3}.$$ Thus, $h = \left\lceil \frac{\log \log n}{\log (k - 2)} \right\rceil$ is a constant. Therefore, in order to apply Lemma 3.2 for this case, it is sufficient that $G$ has girth at least $10h\ell'$ or $\omega(\ell')$. Before showing Theorem 5.2, let us prove the following useful lemma.

\textbf{Lemma 5.1.} Suppose that $G$ be a $d$-regular $n$-vertex graph with girth $\omega(\ell')$, where $\ell' \in [\omega(1), \omega(\log_d n)]$. Also assume that $A_\ell$ has allocated at most $1 \leq n_1 \leq n$ balls to nodes of $G$. Then, with probability $1 - o(1/n)$, $\mathcal{N}_\delta$ holds.

\textbf{Proof.} Define $n_\delta = \lfloor 6 \log_{d-1} n / \delta \rfloor$. For every path of length $\delta$, say $[u,v]$, and a sequence of rounds of size $n_\delta$, say $t_1, t_2, \ldots, t_{n_\delta}$, we define indicator random variable $X_{t_1,t_2,\ldots,t_{n_\delta}} ([u,v])$ as follows

$$X_{t_1,t_2,\ldots,t_{n_\delta}} ([u,v]) := \begin{cases} 
1 & \text{if $[u,v]$ is contained in all $W_{t_1}, W_{t_2}, \ldots, W_{t_{n_\delta}}$,} \\
0 & \text{otherwise,}
\end{cases}$$

Let us first estimate $\Pr \left[ X_{t_1,t_2,\ldots,t_{n_\delta}} ([u,v]) = 1 \right]$. Fix an arbitrary path $[u,v]$ of length $\delta$, where

$$\delta = \lfloor \ell' / k \rfloor / 4 = \min\{ \lfloor \ell' / 16 \rfloor, \lfloor \sqrt{\log_3 n} / 4 \rfloor \}.\quad (5)$$

The second equality is true, because we have set $k = \max\{4, \lfloor \ell' / \sqrt{\log_d n} \rfloor \}$. Clearly, if $W$ be an $\ell'$-walk and $[u,v] \subseteq W = [u_0, u_{\ell'}]$, then

$$d(u_0, u) + d(v, u_{\ell'}) = \ell' - \delta.$$ Moreover, $G$ is a $d$-regular graph with girth at least $\omega(\ell')$, so the total number of $\ell'$-walks containing $[u,v]$ is

$$\sum_{a+b=\ell' - \delta} (d - 1)^a (d - 1)^b = (\ell' - \delta + 1) \cdot (d - 1)^{\ell' - \delta}.$$
On the other hand, the total number of \(\ell\)-walks is \(nd(d - 1)^{\ell - 1}\). So the probability that in some round \(t, 1 \leq t \leq n_1\), we get \([u, v] \subseteq W_t\) is at most

\[
\frac{(\ell - \delta + 1)(d - 1)^{\ell - \delta}}{n \cdot d \cdot (d - 1)^{\ell - 1}} = \frac{(\ell - \delta + 1)(d - 1)}{n \cdot d \cdot (d - 1)^{\delta}} \leq \frac{\ell}{n(d - 1)^{\delta}}.
\]

Thus, we get

\[
\Pr\left[X_{t_1, t_2, \ldots, t_{n_1}} ([u, v]) = 1\right] \leq \left(\frac{\ell}{n(d - 1)^{\delta}}\right)^{n_1} = n^{-n_1}(d - 1)^{(\log_{d - 1}(\ell) - \delta)n_1} \leq n^{-n_1}(d - 1)^{-(n_1^{1/2})} = n^{-n_1}n^{-3},
\]

where the last inequality follows from

\[
\log_{d - 1}(\ell) \leq \delta/2,
\]

as we have \(\ell \in [\omega(1), o(\log_d n)]\) and by (5), \(\delta = \min\{[\ell/16], [\sqrt{\log_d n}/4]\}\). There are at most \(n_1^{n_1} \leq n^{n_1}\) sequences of rounds of size \(n_1\) and \(n(d - 1)^{\delta - 1}\) paths of length \(\delta\), which is at most \(nd\delta\). Thus, by Inequality (6) and the union bound over all sequence of rounds and paths of length \(\delta\) we have

\[
\sum_{\delta-\text{path}} \sum_{t_1, t_2, \ldots, t_{n_1}} \Pr\left[X_{t_1, t_2, \ldots, t_{n_1}} ([u, v]) = 1\right] \leq nd(d - 1)^{\delta - 1}n^{n_1}\Pr\left[X_{t_1, t_2, \ldots, t_{n_1}} ([u, v]) = 1\right] \leq nd^\delta n^{n_1}\Pr\left[X_{t_1, t_2, \ldots, t_{n_1}} ([u, v]) = 1\right] = o(n^2)n^{n_1}\Pr\left[X_{t_1, t_2, \ldots, t_{n_1}} ([u, v]) = 1\right] = o(1/n),
\]

where the first equality follows from \(d^\delta = o(n)\) because \(\delta \leq \ell = o(\log_d n)\). This implies that, with probability \(1 - o(1/n)\), there is no path of length \(\delta\) contained in at least \(n\delta = [6 \log_{d - 1} n/\delta] \ell\)-walks or equivalently \(N\beta\) holds.

**Theorem 5.2.** Suppose that \(\ell \in [\omega(1), o(\log_d n)]\) and \(G\) is a \(d\)-regular \(n\)-vertex graph with girth at least \(10h\ell\) and \(d = \omega(\log n)\), where \(h\) was defined in (4). Then, with high probability, the maximum load attained by \(A_\ell\), denoted by \(m^*\), is bounded from above as follows:

1. If \(\omega(1) \leq \ell \leq 4\gamma_d\), where \(\gamma_d = \sqrt{\log_d n}\). Then,

\[
m^* = \Theta\left(\frac{\log d \cdot n \cdot \log \log n}{\ell^2}\right).
\]

2. If \(\ell \geq 4\gamma_d\), then we have

\[
m^* = \Theta\left(\frac{\log \log n}{\log(\ell / \gamma_d)}\right).
\]

Note that when \(\ell = \Theta(\gamma_d)\), we get the maximum load \(\Theta(\log \log n)\).
Proof. By Lemma 2.1 we have that $A_r$ is an $(\alpha, n_1)$-uniform, where $n_1 = \left\lfloor n/(6\alpha) \right\rfloor$ and $\alpha$ is a constant. Let us divide the allocation process into $s$ phases, where $s$ is the smallest integer satisfying $sn_1 \geq n$. We now focus on the maximum load attained by $A_r$ after allocating $n_1$ balls in the first phase, which is denoted by $m^*_1$. Let us assume that $U_{n_1, \varepsilon, h}$ holds. By Lemma 3.2, if $G$ has girth at least $10\ell h$ and $U_{n_1, \varepsilon, h} \cap \mathcal{N}_d$ happens, then there is a $c$-loaded $(\lambda, \mu)$-tree $T$ contained in $\mathcal{H}_{n_1}$, where $c > 2$ is any constant, $\lambda = 1 + k \sum^{h-1}_{j=0}(k-2)^j$ and $\mu \geq (\ell+1) \cdot k(k-2)^{h-1}$. Let $\mathcal{T}$ denotes the event that such $T$ exists. Thus, we get

$$\Pr[U_{n_1, \varepsilon, h} \cap \mathcal{N}_d] \leq \Pr[\mathcal{T}].$$

Therefore, using the law of total probability and the above inequality we have

$$\Pr[U_{n_1, \varepsilon, h}] = \Pr[U_{n_1, \varepsilon, h} \cap \mathcal{N}_d] + \Pr[U_{n_1, \varepsilon, h} \cap \neg \mathcal{N}_d] \leq \Pr[\mathcal{T}] + \Pr[\neg \mathcal{N}_d] = \Pr[\mathcal{T}] + o(1/n). \quad (7)$$

where the last inequality follows from $\Pr[\neg \mathcal{N}_d] = o(1/n)$ by Lemma 5.1. By the definition of $h$ in (4), we get

$$\lambda \leq 1 + k(1 + (k-2)^h) \leq 2k \log n$$

and

$$\mu = (\ell+1)k(k-2)^{h-1} \geq (\ell+1)(k-2)^h \geq (\ell+1) \log n.$$ 

It only remains to bound $\Pr[\mathcal{T}]$. The definition of $k$ in (4) implies that $k < \ell/2$ and hence it is easy to see that $\mu \geq \lambda$. Now, by applying Lemma 4.1 we get

$$\Pr[\mathcal{T}] \leq n \exp(4\lambda \log(\ell+1) - c\mu) \leq n \exp(-z \log n),$$

where $z = c(\ell+1) - 8k \log(\ell+1)$. Depending on $k$ (defined in (4)) we consider two cases: (1) $k = 4$. Then it is easy to see there exists a constant $c > 2$ such that $z \geq 2.$ (2) $k = \lfloor \ell/\gamma_d \rfloor$. Let us substitute $k$ by $\ell/\gamma_d$. Then,

$$z = c(\ell+1) - 8k \log(\ell+1) \geq \ell(c - (8 \log \ell)/\gamma_d)$$

Recall that $\ell = o(\log_d n) < \log_d n = \gamma_d^2$ and hence, $\log \ell \leq 2 \log \gamma_d$. So,

$$z \geq \ell(c - 16 \log \gamma_d/\gamma_d) = \ell(c - o(1)).$$

This yields that for every $c > 2$, $z = \ell(c - o(1)) > 2$ and hence in both cases we get $\Pr[\mathcal{T}] = o(1/n).$ So, by Inequality (7) we get

$$\Pr[U_{n_1, \varepsilon, h}] \leq \Pr[\mathcal{T}] + o(1/n) = o(1/n).$$

This implies that with probability $1-o(1/n)$, $m^*_1 \leq h\rho + c + 1$. In what follows we show the subadditivity of $A_r$ and conclude that in the second phase the maximum load increases by at most $m^*_1$, whp. Assume that we have a copy of $G$, say $G'$, whose nodes have load exactly $m^*_1$. Also consider allocation algorithm
that allocates balls to nodes of $G'$, which is the same allocation algorithm as $A_\ell$ but it has its own random source and independent from $A_\ell$. Let us couple both algorithms $A_\ell$ and $A_\ell'$ on $G$ and $G'$, respectively. To do so, the coupled process allocates a pair of balls to nodes of $(G,G')$, which proceeds as follows: for every $1 \leq t \leq n_1$, it first picks a one-to-one labeling function $\sigma_t : V(G) \rightarrow \{1,2,\ldots,n\}$ uniformly at random. (Note that $\sigma_t$ is also defined for $G'$ as $V(G) = V(G')$.) Then, the process applies $A_\ell$ on $G$ and picks $W_{n_1+t}$ in $G$ and its copy in $G'$, say $W'_{t}$. After that, the pair of balls, namely $(n_1+t, t)$, is allocated to least-loaded nodes of $W_{n_1+t}$ and $W'_{t}$, respectively, and ties are broken in favor of nodes with the minimum label. One may notice that the processes faithfully implement both allocation algorithms $A_\ell$ on $G$ and $A_\ell'$ on $G'$, simultaneously. Let $X_{u}^{n_1+t}$ and $Y_{u}^{t}$, $t \geq 0$ denote the load of $u \in V(G) = V(G')$ after allocating the $(n_1+t)$th ball and the $t$th ball by the coupled process on $G$ on $G'$, respectively. Now, we show that for every integer $0 \leq t \leq n_1$ and $u \in V(G)$ we have that

$$X_{u}^{n_1+t} \leq Y_{u}^{t}. \quad (8)$$

When $t = 0$, clearly the inequality holds because $Y_{u}^{0} = m^*_1$, for every $u \in V(G')$. Let us assume that for every $t' \leq t$ Inequality (8) holds, then we show it for $t + 1$. Let $v \in W_{n_1+t+1}$ and $v' \in W'_{t+1}$ denote the nodes that receive pair $(n_1 + t + 1, t + 1)$. We now consider two cases:

**Case 1:** $X_{v}^{n_1+t} < Y_{v'}^{t}$. Then allocating ball $n_1 + t + 1$ on $v$ implies that $X_{v}^{n_1+t} + 1 = X_{v}^{n_1+t+1} \leq Y_{v}^{t+1}$, and $Y_{v'}^{t} \leq Y_{v}^{t+1}$.

So, Inequality (8) holds for $t + 1$ and every $u \in V(G)$.

**Case 2:** $X_{v}^{n_1+t} = Y_{v'}^{t}$. Since $W_{n_1+t+1}^{t}$ is a copy of $W_{n_1+t+1}$, $v \in W_{t+1}'$ and $v' \in W_{n_1+t+1}'$. Also, we know that $v$ and $v'$ are nodes with the minimum load contained in $W_{n_1+t+1}$ and $W'_{t+1}$, respectively. So we have $X_{v}^{n_1+t} = X_{v'}^{n_1+t}$.

Since $Y_{v}^{t} = X_{v}^{n_1+t}$, we have $Y_{v'}^{t} = Y_{v}^{t} = X_{v}^{n_1+t}$.

If $v \neq v'$ and $\sigma_{t+1}(v') < \sigma_{t+1}(v)$, then it contradicts the fact that ball $n_1 + t + 1$ is allocated on $v$. Similarly, if $\sigma_{t+1}(v') > \sigma_{t+1}(v)$, it contradicts that ball $t$ is allocated to $v'$. So, we have $v = v'$ and

$$X_{v}^{n_1+t} + 1 = X_{v}^{n_1+t+1} = Y_{v}^{t} + 1 = Y_{v'}^{t+1}.$$

Thus, in both cases, Inequality (8) holds for every $t \geq 0$. If we set $t = n_1$, then the maximum load attained by $A_\ell'$ is at most $2m^*_1$ whp. Therefore, by Inequality (8), $2m^*_1$ is an upper bound for the maximum load attained by $A_\ell$ in the second phase as well. Similarly, we apply the union bound and conclude that after allocating the balls in $s$ phases, the maximum load $m^*$ is at most $sm^*_1$, with probability $1 - o(s/n) = 1 - o(1/n)$.

### 6 | Balanced Allocation on Low-Degree Graphs

In this section we present allocation algorithm $A_\ell'$ on $d$-regular graphs, with $d \geq 3$ and $d = O(\log n)$. The algorithm proceeds as follows: In each round, every ball picks a node, uniformly at random, and it takes a NBRW of length $\ell \cdot r_d$ from the chosen node, where $r_d = \lceil 2 \log_{d-1} \log n \rceil$. After that, the ball
collects the load information of every \( r_d \)th visited node, called the potential choice, it then places itself on a least-loaded potential choice (ties are broken randomly). We now present the following theorem for the maximum load attained by \( A'_\ell \).

**Theorem 6.1.** Suppose that \( \ell \in [o(1), o(\log_d n)] \) be a given integer and \( G \) is a \( d \)-regular graph with \( d = \Theta(\log n) \) and girth at least \( 10\ell(\log \log n)^2 \). Then, with high probability, the maximum load attained by \( A'_\ell \), denoted by \( m^* \), is bounded from above as follows:

1. If \( o(1) \leq \ell < 4\gamma'_d \), where \( \gamma'_d = \sqrt{\log_d n/r_d} \), then we have
   
   \[
   m^* = \Theta\left( \frac{\log_d n \cdot \log \log n}{\ell^2 r_d} \right).
   \]

2. If \( \ell \geq 4\gamma'_d \), then we have
   
   \[
   m^* = \Theta\left( \frac{\log \log n}{\log(\ell/r'_d)} \right).
   \]

There is a one-to-one correspondence between a set of potential choices chosen by a ball and an \( \ell \cdot r_d \)-walk, so the analysis of allocation algorithm \( A'_\ell \) is almost the same as \( A \) and hence we only outline parts of the proof and notations that are slightly different. Let us start by defining an interference graph and show some of its properties (similar to Section 2)

**Definition 7** (Interference graph). For any given pair \((G, \ell)\), let us define interference graph \( I'(G, \ell) \), where each vertex is a set of potential choices contained in an \( \ell \cdot r_d \)-walk on \( G \). Also two vertices are connected if and only if their intersection is nonempty.

Suppose that \( A'_\ell \) has allocated \( n_1 \leq n \) balls. We then define subgraph \( H'_{n_1}(G, \ell) \subset I' \) induced by \( \{S_1, S_2, \ldots, S_{n_1}\} \), where each \( S_i \) is the set of potential choices contained in the \( \ell \cdot r_d \)-walk performed by the \( i \)th ball. For some \( 1 \leq n_1 \leq n \), a \( c \)-loaded \((\lambda, \mu)\)-tree \( T \) in \( H'_{n_1} \) is a \( \lambda \)-vertex tree \( T \subset H_{n_1} \), where \( |\cup_{S \in V(T)} S| \geq \mu \) and each node contained in \( \bigcup_{S \in V(T)} S \) has load at least \( c \).

**Lemma 6.2.** Suppose that \( G \) is a \( d \)-regular \( n \)-vertex graph with girth at least \( o(\ell r_d) \). Then

\[
|V(I'(G, \ell))| = nd(d - 1)^{\ell r_d - 1},
\]

\[
\Delta(I'(G, \ell)) \leq (\ell + 1)^2 d(d - 1)^{\ell r_d - 1}.
\]

Furthermore, the number of rooted \( \lambda \)-vertex trees contained in \( I' \) is at most

\[
4^\lambda V(I') \Delta(I')^{\lambda - 1}.
\]

**Proof.** Since \( G \) has girth at least \( \ell \cdot r_d \), each \( \ell \cdot r_d \)-walk uniquely determines a set of potential choices and hence \( |V(I')| \) is the number of all \( \ell r_d \)-walks, that is,

\[
|V(I')| = nd(d - 1)^{\ell r_d - 1}.
\]

Similar to the proof of Lemma 2.2, we have that for every \( 0 \leq i \leq \ell \), each node can be the \((ir_d)\)th visited node of \( d(d - 1)^{\ell r_d - 1} \) \( \ell \cdot r_d \)-walks. So every node is contained in at
most \((\ell + 1)d(d - 1)^{\ell r_d - 1}\) many walks. On the other hand, each set contains \(\ell + 1\) nodes and hence,

\[
\Delta(I') \leq (\ell + 1)^2d(d - 1)^{\ell r_d - 1}.
\]

Similar to the proof of Lemma 2.2, we show that the number of rooted \(\lambda\)-vertex tree in \(I'\) is at most

\[
4^\lambda V(I')\Delta(I')^{\lambda - 1}.
\]

In order to show an upper bound for the probability that a \(c\)-loaded \((\lambda, \mu)\)-tree in \(H'_{n_1}(G, \ell)\) exists, we have the following lemma.

**Lemma 6.3.** Suppose that for some \(1 \leq n_1 \leq n\), \(A'_{\chi}\) has allocated \(n_1\) balls and \(c\), \(\lambda\) and \(\mu\) are given integers. If \(A'_{\chi}\) is an \((a, n_1)\)-uniform allocation algorithm, then the probability that a \(c\)-loaded \((\lambda, \mu)\)-tree appears in \(H'_{n_1}(G, \ell)\) is at most

\[
n \cdot \exp(4\lambda \log(\ell + 1) - c\mu).
\]

**Proof.** By applying Lemma 2.1 and a counting argument similar to the proof of Lemma 4.1, we prove the lemma.

In order to construct the witness structure, let us define a set of parameters as follows,

\[
k' := \max\{4, \lceil \ell r_d / \sqrt{\ell r_d \log_d n} \rceil\},
\]

\[
\delta' := \lceil \lceil \ell r_d / k' \rceil / 4 \rceil,
\]

\[
\rho' := \lceil 8 \log_d nr_d / \delta' \rceil^2.
\]

Also, for every \(\ell r_d\)-walk \(W\), let

\[
f'(W) = \min_{u \in S} \text{load}(u),
\]

where \(S\) is the set of potential choices corresponding to \(W\) and \(\text{load}(u)\) indicates the load of \(u\). Similar to Section 3, for every \(\ell r_d\)-walk \(W\) we define the Partition-Branch procedure \(PB(k', \rho')\). In the Partition step, \(W\) is partitioned into \(k\) edge-disjoint paths, \(P_k(W) = \{[u_i, u_{i+1}] \mid 0 \leq i \leq k - 1\}\), where each of them has length either \(\lceil \ell r_d / k' \rceil\) or \(\lceil \ell r_d / k' \rceil\). We will also have the Branch step in which for every subpath \(P_i = [u_i, u_{i+1}] \in P_k(W)\), we find another \(\ell r_d\)-walk \(W_{P_i}\), chosen by the algorithm, satisfying

(C1) \(\emptyset \neq W_{P_i} \cap W \subseteq P_i \setminus \{u_i, u_{i+1}\}\).

(C2) \(f'(W_{P_i}) \geq f'(W) - \rho'\).

The recursive construction for the witness graph exactly follows the one in Section 3. Similarly, we define events \(U'_{n, \ell r_d, \delta'}\) and \(N'_{\delta}\). \(U'_{n, \ell r_d, \delta'}\) is the event that at the end of allocating \(n_1\) balls by \(A'_{\chi}\), there is a nodes with load at least \(h'\rho' + c + 1\), where \(c\) is a constant and

\[
h' := \left\lceil \frac{\log n}{\log(k' - 2)} \right\rceil.
\]

Suppose that \(A'_{\chi}\) has allocated \(n\) balls and \(W = \{W_1, \ldots, W_n\}\) be the set of \(\ell r_d\)-walk chosen by \(n\) balls. Then \(N'_{\delta}\) denotes the event that no path of length \(\delta'\) appears in \((6 \log_{d-1} n) / \delta'\) \(\ell r_d\)-walks in \(W\).
7 | A LOWER BOUND

In this section we derive a lower bound for the maximum load attained by allocation algorithms $A'_{\ell}^{d}$ on $G$, where $G$ is an $n$-vertex $d$-regular graph with $d \geq 3$. Recall that

$$r_d := \begin{cases} 
1 & \text{if } d = o(\log n), \\
2 \log_{d-1} \log n & \text{otherwise.}
\end{cases}$$

**Theorem 7.1 (Lower bound).** Suppose that $\ell \in [o(1), o(\log_d n)]$ be a given integer. Also let $G$ be a $d$-regular $n$-vertex graph with girth at least $o(\ell r_d)$. Then, with probability $1 - n^{-O(1)}$, for every $r_d$, the maximum load attained by $A'_{\ell}^{d}$ on $G$ is at least

$$\Omega \left( \frac{\log_d n}{r_d \cdot \ell^2} \right).$$

**Proof.** For every $\ell \cdot r_d$-walk $W$ in $G$, let us define indicator random variable $X_W$ as follows,

$$X_W := \begin{cases} 
1 & \text{if } W \text{ is chosen by at least } \tau \text{ balls in } A'_{\ell}^{d}, \\
0 & \text{otherwise,}
\end{cases}$$

where $\tau$ will be specified later. Define $s$ to be the total number of paths of length $\ell \cdot r_d$. Then,

$$n \leq s \leq na_{\ell r_d}.$$

(9)

Since for each ball $A'_{\ell}^{d}$ samples an $\ell r_d$-walk uniformly at random, we get

$$\Pr[X_W = 1] = \sum_{i=\tau}^{n} \binom{n}{i} \left( \frac{1}{s} \right)^i \left( 1 - \frac{1}{s} \right)^{n-i} \geq \left( \frac{n}{s \cdot \tau} \right)^\tau \left( 1 - \frac{1}{s} \right)^n$$

$$\geq \left( \frac{1}{d^{\ell r_d} \cdot \tau} \right)^\tau \left( 1 - \frac{1}{s} \right)^s \geq d^{-\ell r_d + \log_d \tau} \frac{n^{-\ell r_d + \log_d \tau}}{e}. \quad (10)$$

By setting $\tau = \frac{\log_d n}{\ell r_d}$, and using the fact that $\log_d \tau < \log_d \log_d n \leq r_d \leq \ell r_d$ we get

$$(\ell r_d + \log_d \tau) \tau \leq (\ell r_d + \ell r_d) \tau \leq \log_d n / 3.$$ 

By substituting the above upper bound in (10), we get

$$\Pr[X_W = 1] = \Omega(n^{-1/3}).$$

Let us define the random variable $Y = \sum_{\text{all walks}} X_W$. By linearity of expectation and Inequality (9), we have

$$E[Y] = s \cdot \Pr[X_W = 1] = s \cdot \Omega(n^{-1/3}) = \Omega(n^{2/3}). \quad (11)$$

Intuitively, if an $\ell r_d$-walk, say $W$, is chosen by at least $\tau$ balls, then it is less likely for another $\ell r_d$-walk $W'$ to be chosen by other $\tau$ balls. More precisely, we have

$$\Pr[X_{W'} = 1|X_W = 1] \leq \sum_{i=\tau}^{n-\tau} \binom{n-\tau}{i} \left( \frac{1}{s} \right)^i \left( 1 - \frac{1}{s} \right)^{n-i} \leq \Pr[X_{W'} = 1].$$
This implies that for every two walks $W$ and $W'$ of length $\ell r_d$, the random variables $X_W$ and $X_{W'}$ are negatively correlated and hence
\[
\mathbb{E}[X_W \cdot X_{W'}] \leq \mathbb{E}[X_W] \cdot \mathbb{E}[X_{W'}].
\]
This implies that
\[
\text{Var}[Y] = \sum_W (\mathbb{E}[X_W^2] - (\mathbb{E}[X_W])^2) + \sum_{W \neq W'} (\mathbb{E}[X_W X_{W'}] - \mathbb{E}[X_W] \mathbb{E}[X_{W'}]) \leq 0.
\]
Applying Chebychev’s inequality and above inequality yield that
\[
\text{Pr}[Y = 0] \leq \text{Pr}(|Y - \mathbb{E}[Y]| \geq \mathbb{E}[Y]) = \frac{\text{Var}[Y]}{\left(\mathbb{E}[Y]\right)^2} \leq \frac{1}{\mathbb{E}[Y]}.
\]
By equality (11), we have that $\mathbb{E}[Y] = \Omega(n^{2/3})$. Therefore, with probability at least $1 - \mathcal{O}(n^{-2/3})$, we have $Y \geq 1$ which means there exists a $\ell r_d$-walk $W$, chosen by at least $\tau$ balls. Since each $W$ contains $\ell + 1$ choices, by the pigeonhole principle there is a node with load at least
\[
\Omega\left(\frac{\tau}{\ell}\right) = \Omega\left(\frac{\log_d n}{r_d \ell^2}\right).
\]

8 | MISSING PROOFS

This section is devoted to the missing proofs.

8.1 | Proof of the key lemma

In what follows we prove Lemma 2.1, before that let us define notations and show some useful lemmas. Recall that for every $d$-regular $n$-vertex graph with $d \geq 3$ we define
\[
r_d := \begin{cases} 
1 & \text{if } d = \omega(\log n), \\
\lfloor 2 \log_{d-1} \log n \rfloor & \text{otherwise}.
\end{cases}
\]
In this subsection, we assume that $G$ is a $d$-regular $n$-vertex graph with girth at least $\omega(\ell r_d)$. For each ball $1 \leq t \leq n$, $A_t^{r_d}$ picks a node, uniformly at random, and then it performs a NBRW of length $\ell r_d$, say $W_t$, from the node. Then, for every $0 \leq i \leq \ell$, the $ir_d$th visited node is considered as a potential choice and the ball is allocated to a least-loaded node in the set of potential choices. For every $S \subseteq V(G)$, $\text{Empt}t_{Y_t}(S)$ denotes the number of empty nodes contained in $S$ after allocating $t$ balls by $A_t^{r_d}$. Let $N_{r_d}(u)$ denote the set of nodes at distance $r_d$ from $u$, which are exactly $d(d - 1)^{r_d - 1}$ nodes. Also, let $S_i$ denote the set of potential choices corresponding to $W_i$ performed by the $t$th ball which is the set of every $r_d$th visited node by the $t$th ball. Note that when $r_d = 1$, then $S_i$ is the set of all visited nodes by the $t$th ball. Before we present proof of the key Lemma, let us state some useful lemmas.
Lemma 8.1 (Deviation bounds for moderate independency). Let $X_1, \ldots, X_n$ be arbitrary binary random variables. Let $X^*_1, X^*_2, \ldots, X^*_n$ be binary random variables that are mutually independent and such that for all $i$, $X^*_i$ is independent of $X_1, \ldots, X_{i-1}$. Assume that for all $i$ and all $x_1, \ldots, x_{i-1} \in \{0, 1\}$,

$$
\Pr[X_i = 1 | X_1 = x_1, \ldots, X_{i-1} = x_{i-1}] \geq \Pr[X^*_i = 1].
$$

Then, for all $a \geq 0$, we have

$$
\Pr\left[ \sum_{i=1}^{n} X_i \leq a \right] \leq \Pr\left[ \sum_{i=1}^{n} X^*_i \leq a \right]
$$

and the latter term can be bounded by any deviation bound for independent random variables.

The proof of the above lemma can be found in [3, Lemma 1.18].

Lemma 8.2. Suppose that, for some $0 \leq t \leq n$, $A^t_r$ has allocated the balls until the $(t+1)$th ball on $G$. Also for every $0 \leq i \leq \ell$, let $C_{i,u}$ denotes the event that $u$ is the $i r_d$th visited node by the $(t+1)$th ball. Then for every $0 \leq i \leq \ell$ we have

$$
\Pr[C_{i,u}] = 1/n.
$$

Moreover,

$$
\Pr[u \in S_{t+1}] = \sum_{i=0}^{\ell} \Pr[C_{i,u}] = (\ell + 1)/n,
$$

where $S_{t+1}$ is the set of potential choices corresponding to walk $W_{t+1}$.

Proof. Let us fix an arbitrary $0 \leq i \leq \ell$ and some $u \in V(G)$. Since $G$ has girth at least $\omega(\ell r_d)$, it locally looks like a $d$-regular tree from any node and hence we can easily compute the number of $\ell r_d$-walks visiting $u$ in the $i r_d$th step, that is

$$
d(d-1)^{r_d-1} \times (d-1)^{\ell - i r_d} = d(d-1)^{\ell r_d-1},
$$

where the first multiplier is the number of paths of length $i r_d$ from $u$ and the second one is the number of the paths of length $\ell - i r_d$ from $u$. On the other hand in each round, $A^t_r$ picks an $\ell r_d$-walk randomly from $n d(d-1)^{\ell r_d-1}$ possible $\ell r_d$-walks. Thus, we get

$$
\Pr[C_{i,u}] = \frac{d(d-1)^{\ell r_d-1}}{n d(d-1)^{\ell r_d-1}} = \frac{1}{n}.
$$

Moreover, we know that $u \in S_{t+1}$ if and only if $u$ is the $i \cdot r_d$th visited node for some unique index $0 \leq i \leq \ell$. So we have

$$
\Pr[u \in S_{t+1}] = \Pr[\bigcup_{i=0}^{\ell} C_{i,u}] = \sum_{i=0}^{\ell} \Pr[C_{i,u}] = \sum_{i=0}^{\ell} \frac{1}{n} = \frac{\ell + 1}{n},
$$

where the second equality follows from the fact that all $C_{i,u}$’s are mutually disjoint. 

\hspace{1cm} ◼
Lemma 8.3. Let for some $0 \leq t \leq n - 1$, $X_1, \ldots, X_t$ denote the random variables such that $X_i$ is the vertex to which the $i$th ball is allocated. Also let for each $1 \leq i \leq t$, the $i$th ball is allocated to $v_i$ by $A_{r,i}$. Now suppose that for every $u \in V(G)$ we have,

$$\Pr \left[ \exists y_t \in \mathcal{N}_r(u) \mid |N_r(u)|/2 \mid X_1 = v_1, \ldots, X_t = v_t \right] = 1 - o(n^{-2}),$$

where $\exists y_t \in \mathcal{N}_r(u)$ is the number of empty bins contained in $N_r(u)$ after allocating $t$ balls. Then, for every $u \in V(G)$,

$$\Pr \left[ \text{ball } t + 1 \text{ is allocated to } u \mid X_1 = v_1, \ldots, X_t = v_t \right] \leq \frac{a}{n},$$

where $a$ is a constant.

Proof. Assume that for some $1 \leq t \leq n - 1$, $A_{r,i}^t$ has allocated $t$ balls and $S_{t+1}$ denote the set of potential choices chosen by the $(t + 1)$-ball. Conditional on the allocation of $r$ balls, define events $F_{t+1}$ and $E_{t+1,u}$. Let $F_{t+1}$ to be the event that at least $\lfloor \ell/10 \rfloor$ of nodes in $S_{t+1}$ are empty and $\neg F_{t+1}$ denotes its negation. Also $E_{t+1,u}$ denotes the event that ball $t + 1$ is placed on a given node $u \in V(G)$. Using the law of total probability, for every $u \in V(G)$ we have

$$\Pr \left[ E_{t+1,u} \right] = \Pr \left[ E_{t+1,u} \mid u \notin S_{t+1} \right] \cdot \Pr \left[ u \notin S_{t+1} \right]$$

$$\quad + \Pr \left[ E_{t+1,u} \mid u \in S_{t+1} \text{ and } F_{t+1} \right] \cdot \Pr \left[ u \in S_{t+1} \text{ and } F_{t+1} \right]$$

$$\quad + \Pr \left[ E_{t+1,u} \mid u \in S_{t+1} \text{ and } \neg F_{t+1} \right] \cdot \Pr \left[ u \in S_{t+1} \text{ and } \neg F_{t+1} \right]$$

$$\leq \frac{10}{\ell} \cdot \Pr \left[ u \in S_{t+1} \right] + \Pr \left[ u \in S_{t+1} \text{ and } \neg F_{t+1} \right]$$

where the first summand follows since if $u \notin S_{t+1}$, then ball $t + 1$ cannot be placed on $u$ and the second one follows because ties are broken uniformly at random. Now, by applying Lemma 8.2 and Bayes’ rule we have,

$$\Pr \left[ E_{t+1,u} \right] \leq \frac{10(\ell + 1)}{\ell n} + \sum_{i=0}^{\ell} \Pr \left[ \neg F_{t+1} \text{ and } C_{i,u} \right]$$

$$= \frac{10(\ell + 1)}{\ell n} + \sum_{i=0}^{\ell} \Pr \left[ \neg F_{t+1} \mid C_{i,u} \right] \Pr \left[ C_{i,u} \right]$$

$$= \frac{10(\ell + 1)}{\ell n} + \sum_{i=0}^{\ell} \Pr \left[ \neg F_{t+1} \mid C_{i,u} \right] (1/n), \quad (12)$$

where the last equality follows from $\Pr \left[ C_{i,u} \right] = 1/n$ for every $0 \leq i \leq \ell$, (Lemma 8.2). Now we claim that for every $0 \leq i \leq \ell$, $\Pr \left[ \neg F_{t+1} \mid C_{i,u} \right] \leq 20/\ell$. Assuming the claim is true, plugging this upper bound in Inequity (12) yields that for every $u \in V(G)$,

$$\Pr \left[ E_{t+1,u} \right] \leq \frac{10(\ell + 1)}{\ell n} + \frac{20(\ell + 1)}{\ell n} \leq \frac{30(\ell + 1)}{\ell n},$$
where \(30(\ell + 1)/\ell\) is indeed a constant and hence the statement of the lemma is proved. So it only remains to show the claim.

**Claim 8.4.** For every \(0 \leq i \leq \ell\),

\[
\Pr [\neg P_{i+1} | C_{i,u}] \leq 20/\ell.
\]

**Proof of the claim.** To allocate the \((t + 1)\)th ball, the ball uniformly picks an \(\ell r_d\)-walk, however, conditioning on event \(C_{i,u}\), we only know that node \(u\) is the \(ir_d\)th visited node in \(\ell r_d\)-walk \(W_{t+1}\). Thus, \(W_{t+1}\) can be viewed as the union of two edge-disjoint NBRWs of lengths \(ir_d\) and \(\ell r_d - ir_d\) started from \(u\), namely \(W_u^1\) and \(W_u^2\). Without loss of generality, assume that for some \(2 \leq s \leq \ell\),

\[
|V(W_u^1)| = sr_d \geq 2r_d.
\]

Now, we run a NBRW of length \(sr_d\) from \(u\). For every \(1 \leq j \leq s\), let \(u_j\) denote the \(jr_d\)th visited node and let \(u_0 = u\). It is easy to see that for every \(1 \leq j \leq s\), \(u_j \in N_{r_d}(u_{j-1})\) and \(\{u_1, u_2, \ldots, u_s\}\) is the set of potential choices contained in \(W_u^1\) except \(u_0 = u\). Let \(T_j\) denote a subset of \(N_{r_d}(u_{j-1})\) that can be reached from \(u_{j-1}\) by a NBRW. Note that if \(u_{j-1}\) is a starting node for the NBRW, then \(T_j = N_{r_d}(u_{j-1})\). Otherwise, \(T_j \subset N_{r_d}(u_{j-1})\) and we have \(|T_j| = (d-1)^{\ell_s}\). Thus,

\[
|T_j| \in \{d(d-1)^{\ell_s-1}, (d-1)^{\ell_s}\}.
\]

Conditioning on the event that \(t\) balls were allocated by \(A_{\ell}^{\ell_s}\), for every \(u_j \in \{u_1, u_2, \ldots, u_s\}\) (i.e., set of potential choices corresponding to \(W_u^1\)), define indicator random variable \(I_{u_j}\) as follows:

\[
I_{u_j} := \begin{cases} 
1 & \text{if } u_j \text{ is empty} \\
0 & \text{otherwise},
\end{cases}
\]

One can easily see that a NBRW of length \(r_d\) from \(u_{j-1}\) samples \(u_j\) from \(T_j\), uniformly at random. Thus,

\[
\Pr [I_{u_j} = 1 | u_{j-1} \text{ is chosen as the previous potential choice}] = \frac{\text{Empty}_{Y_j}(T_j)}{|T_j|},
\]

where \(\text{Empty}_{Y_j}(T_j)\) is the size of empty nodes contained in \(T_j\) after allocating \(t\) balls. In what follows we find a lower bound for (15). Let \(Z_{t+1}\) denote the event that the number of nonempty nodes in \(N_{r_d}(u_{j-1})\) after allocating \(t\) balls is at most \(d(d-1)^{\ell_s-1}/2\). By the lemma assumption for every \(u \in V(G)\) we have,

\[
\Pr [Z_{t+1}] = 1 - o(n^{-2}).
\]

Conditional on \(Z_{t+1}\), for every \(1 \leq j \leq s\), the number of nonempty nodes in \(T_j \subset N_{r_d}(u_{j-1})\) is at most

\[
|N_{r_d}(u)|/2 = d(d-1)^{\ell_s-1}/2.
\]

It follows that \(\text{Empty}_{Y_j}(T_j) \geq |T_j| - d(d-1)^{\ell_s-1}/2\). Hence,

\[
\Pr [I_{u_j} = 1 | u_{j-1} \text{ is the previous potential choice and } Z_{t+1}] = \frac{\text{Empty}_{Y_j}(T_j)}{|T_j|} \geq \frac{|T_j| - d(d-1)^{\ell_s-1}/2}{|T_j|} = 1 - \frac{d(d-1)^{\ell_s-1}}{2|T_j|}.
\]
By (14) we have $|T_j| \geq (d - 1)^r$. Thus,

$$\Pr \left[ I_{u_j} \mid u_{j-1} \text{ is the previous potential choice and } Z_{t+1} \right] \geq 1 - \frac{d(d-1)^{r-1}}{2(d-1)^r} = 1 - \frac{d}{2(d-1)} = \frac{d-2}{2(d-1)}.$$  \hspace{1cm}

Applying the above lower bound and the law of total probability, we get

$$\Pr \left[ I_{u_j} = 1 \mid u_{j-1} \text{ is chosen as the previous potential choice} \right] = \Pr \left[ I_{u_j} = 1 \mid u_{j-1} \text{ is chosen as the previous potential choice and } Z_{t+1} \right] \Pr \left[ Z_{t+1} \right] + \Pr \left[ I_{u_j} = 1 \mid u_{j-1} \text{ is chosen as the previous potential choice and } \neg Z_{t+1} \right] \Pr \left[ \neg Z_{t+1} \right] \geq \frac{d-2}{2(d-1)} \left( 1 - o(n^{-2}) \right) + o(n^{-2}) \geq \frac{1}{5},$$

where the last inequality holds for every $d \geq 3$. Therefore, we conclude that for every $j$ the lower bound is independent of the previous potential choice. So we have that for every $1 \leq j \leq s$, and all $x_1, \ldots, x_{j-1} \in \{0, 1\}$ we will have

$$\Pr \left[ I_{u_j} = 1 \mid I_{u_1} = x_1, \ldots, I_{u_{j-1}} = x_{j-1} \right] \geq 1/5. \quad (16)$$

A similar argument also works for the potential choices contained in $W^2_{t+1}$ and hence for every $v \in S_{t+1} \setminus \{u\}$, we get $\Pr \left[ I_{v} = 1 \right] \geq 1/5$. Conditional on $C_{i,u}$, define random variable $Y = \sum_{v \in S_{t+1} \setminus \{u\}} I_{v}$ which is the number of empty nodes in $S_{t+1} \setminus \{u\}$. Then, $E \left[ Y \right] \geq \ell / 5$. Also, recall that $\neg F_{t+1}$ denotes the event that the number of empty nodes in $S_{t+1}$ is at most $\lfloor \ell / 10 \rfloor - 1$. Recall that we have defined random variable $Y$ conditioning on $C_{i,u}$. Also note that $Y + 1$ is an upper bound for the number of empty nodes in $S_{t+1}$ so we have

$$\Pr \left[ \neg F_{t+1} \mid C_{i,u} \right] \leq \Pr \left[ Y \leq \ell / 10 \right]. \quad (17)$$

Let $Y^*$ be the summation of $\ell$ independent Bernoulli random variables with success probability $1/5$. Hence, $E \left[ Y^* \right] = \ell / 5$. Applying the lower bound obtained in (16) and Lemma 8.1 we get,

$$\Pr \left[ Y \leq \ell / 10 \right] \leq \Pr \left[ Y^* \leq E \left[ Y^* \right] / 2 \right] \quad (18)$$

Since $Y^*$ is the summation of $\ell$ independent Bernoulli random variables, we have $\text{Var} \left[ Y^* \right] \leq E \left[ Y^* \right]$. Also applying Chebychev’s inequality results in

$$\Pr \left[ Y^* \leq E \left[ Y^* \right] / 2 \right] \leq \Pr \left[ \left| Y^* - E \left[ Y^* \right] \right| \geq E \left[ Y^* \right] / 2 \right] \leq \frac{\text{Var} \left[ Y^* \right]}{\left( E \left[ Y^* \right] / 2 \right)^2} \leq \frac{4}{E \left[ Y^* \right]}. $$

Using the above upper bound and Inequalities (17) and (18), we get

$$\Pr \left[ \neg F_{t+1} \mid C_{i,u} \right] \leq \Pr \left[ Y \leq \ell / 10 \right] \leq \frac{4}{E \left[ Y^* \right]} \leq \frac{4}{\ell / 5} = \frac{20}{\ell},$$

which proves the claim.

In order to prove our key lemma, we apply a potential function argument which is similar to [6, Theorem 1.4 ].
Proof of Key Lemma (Lemma 2.1). In the following we show that there exists some \( n_1 = \Theta(n) \) such that if \( A_{\ell}^{r_d} \) allocates first \( n_1 \) balls, then with high probability for every node \( u \), \( N_{r_d}(u) \) does contain at least \( |N_{r_d}(u)|/2 = d(d-1)^{r_d-1}/2 \) empty nodes and hence by using the Lemma 8.3 we get the \((\alpha, n_1)\)-uniformity of \( A_{\ell}^{r_d} \). To show this, let us define potential function

\[
\Phi(t) = \sum_{u \in V(G)} e^{a_t(u)},
\]

where \( a_t(u) \) denotes the number of nonempty nodes of \( N_{r_d}(u) \) after allocating \( t \) balls. It is clear that \( \Phi(0) = n \). Define

\[
\Delta = \frac{d(d - 1)^{r_d}}{4}.
\]

Now, let us assume for some \( 1 \leq t \leq n \), after allocating \( t \) balls by \( A_{\ell}^{r_d} \) we have \( \Phi(t) \leq n \cdot e^\Delta \). Thus, for every \( u \in V(G) \),

\[
e^{a_t(u)} \leq \Phi(t) \leq e^{\log n + \Delta}.
\]

By the definition of \( r_d \), we have \( \Delta = o(\log n) \) and hence,

\[
a_t(u) \leq \log n + \Delta \leq 2\Delta = d(d - 1)^{r_d-1}/2.
\]

So for every \( u \in V(G) \),

\[
\text{Empty}_t(N_{r_d}(u)) \geq |N_{r_d}(u)|/2 = 2\Delta.
\]

Let us define indicator random variable \( I_{t+1}(u) \) for every \( u \in V(G) \) as follows:

\[
I_{t+1}(u) := \begin{cases} 1 & \text{if ball } t+1 \text{ is placed on an empty node in } N_{r_d}(u) \text{ by } A_{\ell}^{r_d}, \\ 0 & \text{otherwise}. \end{cases}
\]

Applying Lemma 8.3 shows that if, for some \( 1 \leq t \leq n \),

\[
\text{Empty}_t(N_{r_d}(u)) \geq |N_{r_d}(u)|/2 = 2\Delta,
\]

then for every \( u \in V(G) \)

\[
\Pr \left[ \text{ball } t+1 \text{ is allocated to } u \mid X_1 = v_1, \ldots, X_t = v_t \right] \leq \frac{a}{n},
\]

where \( a \) is a constant. Thus by the union bound over all empty nodes of \( N_{r_d}(u) \) we have

\[
\Pr \left[ I_{t+1}(u) = 1 \mid \text{Empty}_t(N_{r_d}(u)) \geq 2\Delta \right] \leq \sum_{u \in N_{r_d}(u)} \frac{a}{n} = \frac{4a\Delta}{n},
\]

where it follows from \( |N_{r_d}(u)| = 4\Delta \). Since \( \Phi(t) \leq n \cdot e^\Delta \) implies \( \text{Empty}_t(N_{r_d}(u)) \geq 2\Delta \), we have

\[
\Pr \left[ I_{t+1}(u) = 1 \mid \Phi(t) \leq n \cdot e^\Delta \right] \leq \frac{4a\Delta}{n}. \tag{19}
\]
Thus,

\[ E \left[ \Phi(t + 1) \mid \Phi(t) \leq n \cdot e^\Delta \right] \]

\[ \leq \sum_{u \in V(G)} \Pr \left[ I_{t+1}(u) = 1 \mid \Phi(t) \leq n \cdot e^\Delta \right] \cdot e^{\alpha(u)+1} \]

\[ + \sum_{u \in V(G)} \Pr \left[ I_{t+1}(u) = 0 \mid \Phi(t) \leq n \cdot e^\Delta \right] \cdot e^{\alpha(u)} \]

\[ \leq \sum_{u \in V(G)} \left( \frac{\alpha \cdot e \cdot 4\Delta}{n} + 1 \right) \cdot e^{\alpha(u)} = \left( 1 + \frac{4\alpha \cdot e \cdot \Delta}{n} \right) \Phi(t). \]

So we get,

\[ E \left[ \Phi(t + 1) \mid \Phi(t) \leq n \cdot e^\Delta \right] = \left( 1 + \frac{4\alpha \cdot e \cdot \Delta}{n} \right) \Phi(t). \] (20)

Let us define \( \Psi(t) := \min \{ \Phi(t), n \cdot e^\Delta \} \). Then we claim that,

\[ E \left[ \Psi(t + 1) \mid \Psi(t) \right] \leq \left( 1 + \frac{4\alpha \cdot e \cdot \Delta}{n} \right) \Psi(t). \]

To show the above inequality we consider two cases:

1. If \( \Psi(t) < n \cdot e^\Delta \), then we have \( \Psi(t) = \Phi(t) \) and hence by inequality (20) it holds.
2. If \( \Psi(t) = n \cdot e^\Delta \), then it does not depend on \( t \) and again we have the above inequality.

Applying the law of total expectation and the fact that \( \Phi(0) = n \) result that

\[ E \left[ \Psi(t) \right] \leq \left( 1 + \frac{4\alpha \cdot e \cdot \Delta}{n} \right)^t \Psi(0) = \left( 1 + \frac{4\alpha \cdot e \cdot \Delta}{n} \right)^t \Phi(0) = \left( 1 + \frac{4\alpha \cdot e \cdot \Delta}{n} \right)^t n. \]

Let us set \( n_1 = n/(6e\alpha) \). Then applying Markov’s inequality implies that

\[ \Pr \left[ \Psi(n_1) \geq n \cdot e^\Delta \right] \leq \frac{\left( 1 + \frac{4\alpha e\Delta}{n} \right)^{n_1}}{e^{\Delta}} \leq e^{-\Delta/3} = e^{-o(\log n)}, \]

where it follows from definition of \( r_d \). So with probability \( 1 - n^{-o(1)} \), we have \( \Phi(n_1) = \Psi(n_1) < n \cdot e^\Delta \).

Since \( \Phi(t) \) is an increasing function in \( t \), for every \( t + 1 \leq n_1 \) we have

\[ \Phi(t + 1) \leq n \cdot e^\Delta. \]

In particular, this implies that with probability \( 1 - o(n^{-2}) \), for every \( u \in V(G) \),

\[ \text{Empty}_Y(Y_{t+1}(N_{r_d}(u)) \geq 2\Delta. \]

So, applying Lemma 8.3 shows that for every \( 0 < t < n_1 \) and \( u \in V(G) \),

\[ \Pr \left[ \text{ball} \ t + 1 \text{ is placed on} \ u \mid X_1 = v_1, \ldots, X_t = v_t \right] \leq \frac{\alpha}{n}. \]

\[ \blacksquare \]
8.2 Proof of Lemma 3.2

Before we present the proof of Lemma 3.2, we need to show some lemmas about the properties of the recursive construction of the witness graph. Note that the intersection (union) of two arbitrary graphs is a graph whose vertex set and edge set are the intersection (union) of the vertex and edge sets of those graphs. Let $\cap$ and $\cup$ denote the graphical intersection and union. Sometimes we consider $\ell$-walks as the set of nodes contained in $W$. So for every two $\ell$-walks $W$ and $W'$, $W \cap W'$ denotes the intersection of their nodes. In this subsection, for every $0 \leq j \leq h$, $H_j \subset G$ denotes the graphical union of all $\ell$-walks up to the $(j + 1)$th level, called $\mathcal{L}_{j+1}$, (for the definition of $\mathcal{L}_j$’s see Section 3). So we have,

$$
H_0 := R
$$

$$
H_1 := (\cup_{W \in \mathcal{L}_1} W) \cup H_0,
$$

$$
\vdots
$$

$$
H_h := (\cup_{W \in \mathcal{L}_h} W) \cup H_{h-1}.
$$

It also follows that

$$
H_0 \subset H_1 \subset H_2 \subset \ldots \subset H_h \subset G.
$$

We also use $V(H_j)$ to denote the vertex set of $H_j$. Let us show the following useful lemma.

**Lemma 8.5.** If $G$ has girth at least $10h\ell$, then for every $0 \leq j \leq h$, $H_j$ is a tree.

*Proof.* We first inductively prove that for every $0 \leq j \leq h$, the diameter of $H_j$ is at most $(2j + 1)\ell$. If $j = 0$, then we have $H_0 = R$, where $R$ is an $\ell$-walk and hence $H_0$ is a tree with diameter $\ell$. Assume that for every $0 \leq j \leq j_0 < h$, the diameter of $H_j$ is at most $(2j_0 + 1)\ell$ (induction hypothesis). We know that every $\ell$-walk in the $(j_0 + 1)$th level intersects a path in $H_{j_0}$, so the distance between any two nodes of $H_{j_0+1}$ increases by at most $2\ell$ and thus the diameter of $H_{j_0+1}$ increases by at most $2\ell$. Thus, we have

$$(2j_0 + 1)\ell + 2\ell = (2(j_0 + 1) + 1)\ell.$$

So we conclude that, for every $0 \leq j \leq h$, $H_j$ is a connected graph with diameter at most $(2h + 1)\ell$. Now, we have to show all $H_j$’s are tree. Toward a contradiction, assume that for some $0 \leq j \leq h$, $H_j$ contains at least one cycle. Let $s$ be the length of a shortest cycle contained in $H_j$. Then $s$ cannot exceeds twice of the diameter of $H_j$ and we have

$$s \leq 2 \cdot \text{diam}(H_j) \leq 2(2h + 1)\ell = (4h + 2)\ell \leq 6h\ell.$$

This contradicts the fact that $H_j \subset G$ and $G$ has girth at least $10h\ell$.

**Lemma 8.6.** For every $1 \leq j \leq h$, the $j$th level, $\mathcal{L}_j$, contains $k(k - 2)^{j-1}$ disjoint $\ell$-walks. Moreover, every $\ell$-walk in $\mathcal{L}_j$ intersects only one $\ell$-walk in the previous level, which is its father.

*Proof.* We inductively show the lemma level by level. Recall that every $P \in \mathcal{P}^0_k(W)$ is a free subpath, as it does not share any node with $W$’s father. Let us assume that for some $1 \leq j \leq h$, $W_p \in \mathcal{L}_j$ be an arbitrary $\ell$-walls created by the branch step on $P = [u, v]$. Then, we say $H_j$ has $D_j$ property if for every $W_p \in \mathcal{L}_j$, every $x \in W_p$, and every $y \in V(H_j) \setminus W_p$, the unique path in $H_j$ ($H_j$ is tree by Lemma 8.5) connecting $x$ to $y$ contains $u$ or $v$, which are called the boundary of $P = [u, v]$. It is not hard to see that
if for some $j$ the lemma holds, then $D_j$ also holds for $H_j$. Because $\ell$-walks in the last level are mutually disjoint and they only intersect their father(s). So one must use the boundaries to connect nodes from different $\ell$-walks in the last level (see Figure 2).

Let us begin with $j = 1$. For the sake of contradiction, assume that $W_{P_1}, W_{P_2} \in \mathcal{L}_1$ intersect each other. Recall that $W_{P_1}$ and $W_{P_2}$ are created by the Branch step on edge-disjoint paths, $P_l = [u_l, u_{l+1}], P_r = [u_r, u_{r+1}] \in \mathcal{P}_1(R)$. Let $\{u_l, u_{l+1}\}$ and $\{u_r, u_{r+1}\}$ be the boundary of $P_l$ and $P_r$, respectively. Clearly, $W_{P_1} \cup_g W_{P_2}$ is a connected graph as they intersect each other. So, by Condition (C1), in the Partition-Branch procedure, we are able to choose two arbitrary nodes $z \in P_l \cap W_{P_1}$ and $z' \in P_r \cap W_{P_2}$. Also by (C1), $W_{P_1}$ and $W_{P_2}$ excludes the boundaries and thus we get a path from $z$ to $z'$ via $W_{P_1} \cup_g W_{P_2} \subset H_1$ that excludes the boundaries. On the other hand, we know $z$ and $z'$ are nodes of $H_0 = R$, which is a tree. Then there is a unique path in $H_0$ that connects $z$ to $z'$ via the boundaries of $P_l$ and $P_r$. This contradicts the fact that there is a unique path in $H_1 \supset H_0$, because $H_1$ is a tree by Lemma 8.5. So we infer that there are $k$ disjoint $\ell$-walks in $\mathcal{L}_1$. Therefore, the lemma is proved for $j = 1$.

Suppose that for some $j_0$, $1 \leq j_0 \leq h$, the statement of the lemma holds (induction hypothesis). This assumption implies that the $\ell$-walks in level $\mathcal{L}_{j_0}$ are mutually disjoint and they only intersect their father(s) and consequently $D_{j_0}$ holds for $H_{j_0}$. Now, let us prove the lemma for the next level (i.e., $j_0 + 1$). Similar to case $j = 1$, toward a contradiction assume that two $\ell$-walks $W_{P_1}, W_{P_2} \in \mathcal{L}_{i_{j_0}+1}$ intersect each other, which are created by the branch on $P$ and $P'$, respectively. Also, let $x \in P \cap W_{P_1}$ and $y \in P \cap W_{P_2}$ be two arbitrary nodes. Then, we get a path in $W_{P_1} \cup_g W_{P_2} \subset H_{i_{j_0}+1}$ that connects $x$ to $y$ and excludes the boundaries of $P$ and $P'$, because by (C1) $W_{P_1}$ and $W_{P_2}$ do not contain the boundaries. Notice that $P$ and $P'$ are free subpaths from $\ell$-walks in $\mathcal{L}_j$. So by $D_{j_0}$ property, the path in $H_{j_0}$ connecting $x$ to $y$ uses nodes from the boundaries, while we get a path in $H_{i_{j_0}+1}$ that exclude boundaries. This contradicts the fact that $H_{i_{j_0}+1} \supset H_{j_0}$ is a tree by Lemma 8.5. So the $\ell$-walks in $\mathcal{L}_{i_{j_0}+1}$ are disjoint and by the construction we have $|\mathcal{L}_{i_{j_0}+1}| = (k-2)|\mathcal{L}_{j_0}|$ and hence $|\mathcal{L}_{j_0+1}| = k(k-2)^h$.

It only remains to prove every $\ell$-walk only intersect its father in the previous level. Toward a contradiction, assume that $W_{P} \in \mathcal{L}_{i_{j_0}+1}$ intersects an $\ell$-walk, say $W$, in some previous level that is not its father. Recall that $W_{P}$ is created by the branch step on free subpath $P = [u, v] \in \mathcal{P}_{k}^0(W')$, where $W'$ denotes the father of $W_P$. Let us fix some arbitrary nodes $x \in W_{P_1} \cap P$ and $y \in W_{P_2} \cap P$. By (C1), $x$ is not the boundary of $P$ (i.e., $\{u, v\}$). We now get a new path from $x$ to $y$ in $H_{i_{j_0}+1}$ excluding and $v$ (via a path in $W_{P_1} \cup_g W'$). On the other hand, $D_{j_0}$ property holds for $H_{j_0}$ and there is only one path from $x \in P$ to $y \in V(H_{i_{j_0}})$ including a node from the boundary of $P$ as $P$ is a free subpath of $W' \in \mathcal{L}_{j_0}$. So we now get two different paths from $x$ to $y$ in $H_{i_{j_0}+1}$, which contradicts the fact that $H_{i_{j_0}+1}$ is a tree. ■

**Proof of Lemma 3.2.** Consider graph $T \subset \mathcal{H}_0$ whose nodes are the set of all $\ell$-walks in $\bigcup_{j=0}^{h-1} \mathcal{L}_j$, where $\mathcal{L}_0 = \{R\}$. And, two nodes are connected, if and only if the corresponding $\ell$-walks intersect each other. By Lemma 8.6 for every $1 \leq j \leq h$, the $j$th level contains $k(k-2)^{j-1}$ disjoint $\ell$-walks and they intersect either their fathers or their $k-2$ children. This implies that

$$|V(T)| = \lambda = 1 + k \sum_{j=0}^{h-1} (k-2)^j.$$ 

If we only consider the $h$th level, then we get

$$|\bigcup_{W \in V(T)} W| \geq \mu = (\ell + 1) \cdot k(k-2)^{h-1}.$$ 

By (C2) in the Partition-Branch procedure, we have that for every $W \in \mathcal{L}_j$, $1 \leq j \leq h$,

$$f(W) \geq (h-j)\rho + c.$$ 

Hence every node in $\bigcup_{W \in V(T)} W$ has load at least $c$. ■
ACKNOWLEDGMENT

The author wants to thank Thomas Sauerwald for introducing the problem and several helpful discussions.

REFERENCES

1. N. Alon, I. Benjamini, E. Lubetzky, and S. Sodin, Non-backtracking random walks mix faster, Commun. Contemp. Math. 9 (2007), 585–603.
2. N. Alon and E. Lubetzky, Poisson approximation for non-backtracking random walks, Israel J. Math. 174 (2009), 227–252.
3. A. Auger and B. Doerr, Theory of randomized search heuristics: Foundations and recent developments, World Scientific Publishing Co., River Edge, NJ, 2011.
4. Y. Azar, A. Z. Broder, A. R. Karlin, and E. Upfal, Balanced allocations, SIAM J. Comput. 29 (1999), 180–200.
5. P. Berenbrink, A. Brinkmann, T. Friedetzky, and L. Nagel, Balls into bins with related random choices, J. Parallel Distrib. Comput. 72 (2012), 246–253.
6. P. Bogdan, T. Sauerwald, A. Stauffer, and H. Sun, Balls into bins via local search. Proc. 24th Symp. Discrete Algorithms (SODA), 2013, pp. 16–34.
7. J. W. Byers, J. Considine, and M. Mitzenmacher, Geometric generalizations of the power of two choices. Proc. 16th Symp. Parallelism in Algorithms and Architectures (SPAA), 2004, pp. 54–63.
8. C. Cooper, A. M. Frieze, and T. Radzik, Multiple random walks in random regular graphs, SIAM J. Discrete Math. 23 (2009), 1738–1761.
9. X. Dahan, Regular graphs of large girth and arbitrary degree, Combinatorica. 34 (2014), 407–426.
10. B. Godfrey, Balls and bins with structure: Balanced allocations on hypergraphs. Proc. 19th Symp. Discrete Algorithms (SODA), 2008, pp. 511–517.
11. K. Kenthapadi and R. Panigrahy, Balanced allocation on graphs. Proc. 17th Symp. Discrete Algorithms (SODA), 2006, pp. 434–443.
12. Donald Knuth, The art of computer programming. Fundamental Algorithms, Addison-Wesley, Vol. 1, 3rd ed., 1997.
13. M. Mitzenmacher, A. W. Richa, and R. Sitaraman, The power of two random choices: A survey of technique and results. Handbook of Randomized Computation, Vol. 1, 2001, pp. 255-312.
14. Y. Peres, K. Talwar, and U. Wieder, Graphical balanced allocations and the (1 + \(\beta\))-choice process, Random Structures Algorithms (2014). https://doi.org/10.1002/rsa.20558.
15. M. J. Siavoshani, A. Pourmiri, and S. P. Shariatpanahi, Storage, communication, and load balancing trade-off in distributed cache networks, IEEE Trans. Parallel Distrib. Syst. 29 (2018), 943–957.
16. B. Vöcking, How asymmetry helps load balancing, J. ACM. 50 (2003), 568–589.

How to cite this article: Pourmiri A. Balanced allocation on graphs: A random walk approach*. Random Struct Alg. 2019;55:980–1009. https://doi.org/10.1002/rsa.20875