

Optimal Cover Time for a Graph-Based Coupon Collector Process

Nedialko B. Dimitrov^{1,2} and C. Greg Plaxton^{1,3}

January 2005

Abstract

In this paper we study the following covering process defined over an arbitrary directed graph. Each node is initially uncovered and is assigned a random integer rank drawn from a suitable range. The process then proceeds in rounds. In each round, a uniformly random node is selected and its lowest-ranked uncovered outgoing neighbor, if any, is covered. We prove that if each node has in-degree $\Theta(d)$ and out-degree $O(d)$, then with high probability, every node is covered within $O(n \cdot \max(1, (\log n)/d))$ rounds, matching a lower bound due to Alon. Alon has also shown that, for a certain class of d -regular expander graphs, the upper bound holds no matter what method is used to choose the uncovered neighbor. In contrast, we show that for arbitrary d -regular graphs, the method used to choose the uncovered neighbor can affect the cover time by more than a constant factor.

Adler *et al.* have previously shown that a similar covering process is useful in the analysis of an optimal load balancing scheme for a hypercubic distributed hash table. We show that our covering process leads to an optimal load balancing scheme for any distributed hash table in which each machine has a logarithmic number of pointers to other machines.

¹ Department of Computer Science, University of Texas at Austin, 1 University Station C0500, Austin, Texas 78712–0233.

² Email: ned@cs.utexas.edu. Supported by an MCD Fellowship from the University of Texas at Austin.

³ Email: plaxton@cs.utexas.edu. Supported by NSF Grants CCR–0310970 and ANI–0326001.

1 Introduction

One of the most commonly discussed stochastic processes in computer science is the so-called coupon collector process [6]. In that process, there are n distinct coupons and we proceed in rounds, collecting one uniformly random coupon (with replacement) in each round. It is natural to ask how many rounds are needed to collect all n coupons. It is a well known fact, often given as an exercise, that with high probability this number of rounds is $O(n \log n)$.

This result is in some sense unsatisfying, since one might hope to collect the n coupons in $O(n)$ rounds. This shortcoming has motivated Adler *et al.* [1] and Alon [2] to study a similar graph-based covering process. The nodes of the graph represent the coupons and covering a node represents collecting a coupon. In each round, a uniformly random node w is selected. If an uncovered neighbor of w exists, choose one such uncovered neighbor and cover it. We refer to this process as process CC.

Process CC can use a variety of different *covering methods* to decide which uncovered neighbor to cover. If our ultimate goal is to minimize cover time, certainly the most powerful covering method available is an offline method with knowledge of the entire sequence of node selections and with infinite computing power. We refer to this powerful cover time minimizing version of process CC as process MIN. To achieve our $O(n)$ goal, it is natural to consider $\log n$ -regular graphs since the work of Alon implies process MIN has an expected cover time of $\Omega(n \max(1, (\log n)/d))$ rounds on d -regular graphs [2].

1.1 Logarithmic-Degree Graphs

Another natural version of process CC — in which the covering method chooses a uniformly random uncovered neighbor, if any — was studied by Adler *et al.* [1] and by Alon [2]. We refer to this version of process CC as process UNI. Alon shows that for logarithmic-degree Ramanujan expander graphs, process UNI completes in $O(n)$ time, matching the lower bound for process MIN.

Adler *et al.* show that for the hypercube, which has a weak expansion property but is not an expander, process UNI takes $O(n)$ time, also matching the lower bound for process MIN [1]. They also show that for arbitrary logarithmic-degree graphs, process UNI completes in $O(n \log \log n)$ time. Furthermore, Adler *et al.* present an application of process UNI to load balancing in a hypercubic distributed hash table (DHT).

A process that is intuitively similar to process UNI is one where we initially assign a rank to each node using a uniformly random permutation of the nodes, and the covering method covers the minimum-rank uncovered neighbor, if any. We refer to this permutation-based version of process CC as process P-RANK. In this paper, we show process P-RANK completes in $O(n)$ time on arbitrary logarithmic-degree graphs.

In fact, we analyze a more general and local version of process CC in which each node initially chooses a uniformly random rank in a suitable range, and the covering method covers the minimum-rank uncovered neighbor of the selected node. (We assume that the nodes are numbered from 1 to n , and that ties in rank are broken in favor of the lower-numbered node.) We refer to this random rank version of process CC as process R-RANK.

Adler *et al.* show that in a hypercubic DHT, if new nodes use process UNI to choose which existing node's name space to split, then there exist constants c_1 and c_2 such that after n joins, no node is responsible for more than a c_1/n or less than a $1/(c_2n)$ fraction of the name space with high probability. By applying our results within the framework of Corollary 2.3 of Adler *et al.*, we can show that if process R-RANK is used by joining nodes to choose which existing node to split, then in *any* DHT where each node holds $\log n$ pointers to other nodes in the table, there exists a constant c_3 such that after n joins no node is responsible for more than a c_3/n fraction of the name space with high probability. For example, if process R-RANK is used by joining nodes, our results establishes a load balance guarantee for the well-known DHT Chord, which has logarithmic degree but is not a hypercube.

As discussed in the preceding paragraph, our results may be used to upper bound the maximum load of any machine in the DHT. For completeness, it may be interesting to establish a lower bound on the minimum load, as Adler *et al.* show for the hypercubic DHT. We conjecture that with high probability the first $\lfloor \epsilon n \rfloor$ rounds of process R-RANK are successful on arbitrary logarithmic-degree graphs. If this conjecture holds, the result can be applied within the framework of Corollary 2.4 of Adler *et al.* to establish the desired lower bound on the minimum load for any hash table with logarithmic degree. In practice, however, such a lower bound on the minimum load is unimportant in load balancing applications where the objective is to avoid overload.

1.2 Results for General Graphs

Alon shows that process MIN on any d -regular graph has expected time at least $n - \frac{n}{d} + \frac{n}{d} \log_e(\frac{n}{d})$ [2]. Alon also shows that process UNI completes in time $n + (1 + o(1))n \frac{\log n}{d}$ for random nearly d -regular graphs. Alon further shows that on any (n, d, λ) -expander graph the expected time of process UNI is at most $n + n(\frac{\lambda}{d})^2(\log_e n + 1)$. In particular, this implies that on Ramanujan graphs process UNI completes in $(1 + o(1))n$ time, matching the lower bound for process MIN.

If our ultimate goal is to maximize cover time, certainly the most powerful neighbor selection method available is an offline adversary with knowledge of the entire sequence of node selections and with infinite computing power. We refer to this powerful cover time maximizing version of process CC as process MAX. Alon notes that the upper bounds for expanders hold even if after every round an adversary “is allowed to shift the uncovered nodes to any place he wishes, keeping their number.” In particular, this shows that on Ramanujan graphs, the cover time for process MAX matches the cover time for process MIN, up to constant factors. In effect, the covering method does not matter for this class of graphs.

Another previously studied variant of process CC favors covering the selected node. In this variant, we check — immediately after selecting a uniformly random node — if the selected node is uncovered. If it is, we cover it and move to the next selection. Only otherwise do we consider the neighbors of the selected node. We refer to the selection-biased variants of process UNI, process P-RANK, and process R-RANK as process UNI', process P-RANK', and process R-RANK', respectively.

Adler *et al.* show that for every d -regular graph, processes UNI and UNI' complete in $O(n + n \frac{\log n \log d}{d})$ time[1]. They also show that for random d -regular graphs only $O(n + n \frac{\log n}{d})$ steps are needed. Furthermore, they exhibit an application of process UNI' to load balancing in DHTs.

All of the results matching Alon's lower bound for process MIN presented prior to this work have used some expansion properties of the underlying graph. In contrast, our proof techniques do not require the underlying graph to have any particular structure. Thus we show the more general result that for directed graphs (with self-loops but no parallel edges) where each node has in-degree at least δ_{in} and at most Δ_{in} , while the out-degree is at most Δ_{out} , both process R-RANK and process R-RANK' cover all nodes in $O(n \max(\Delta_{in} \Delta_{out} / \delta_{in}^2, (\log n) / \delta_{in}))$ rounds with high probability. This result matches Alon's lower bound for $\delta_{in} = \Delta_{in} = \Delta_{out} = \Theta(d)$, and is thus optimal under these conditions.

Furthermore, Alon's results for Ramanujan graphs raise the question whether there is any separation between the cover times for process MAX and process MIN. In other words, are there any graphs for which the choice of covering method matters? We define a weakly adversarial process, process A-RANK, that is similar to process P-RANK. In process A-RANK, instead of picking a uniformly random permutation, an adversary is initially allowed to fix the permutation used to assign ranks to the nodes. We then proceed as in process P-RANK. Define the even weaker adversarial process A-RANK' similarly to process P-RANK'. We show that there exists a logarithmic-degree graph on which process A-RANK and process A-RANK' each take $\omega(n)$ rounds to complete. This implies that in general there is separation between the cover times of process MIN and process MAX. Or, in other words, the covering method does matter.

1.3 Proof Outline

The proof of our theorem is inspired by the delay sequence argument used by Ranade for the analysis of a certain packet routing problem on the butterfly [7] (see also [5]). In a delay sequence argument, we identify certain combinatorial structures that exist whenever the random process lasts for a long time. Then, we show that the probability any of these structures exist is small. This in turn implies an upper bound on the running time of the random process.

There are significant differences between our proof and that of Ranade. For example, in our problem, the connection between the running time and the length of a delay sequence is not clear-cut, while in the butterfly routing problem analyzed by Ranade, the length of the delay sequence is equal to the running time. But let us begin by giving the notion of a delay sequence in our problem.

Consider the node that was covered last, w_1 . Why wasn't w_1 covered earlier? It was not covered earlier because at the last opportunity to cover w_1 — that is, the last selection in w_1 's neighborhood — we covered some other node, w_2 instead. Thus, w_1 was delayed by w_2 . Similarly, w_2 was delayed by some node w_3 , et cetera, until finally we reach a node w_k that was not delayed, i.e., w_k was covered at the first opportunity. The sequence of nodes w_1, \dots, w_k corresponds to our notion of a delay sequence.

In analyzing process R-RANK, we find it useful to first analyze a much simpler process, process SELECT, in which we repeatedly select a uniformly random node, never covering anything. After establishing several lemmas for the simpler process, we proceed to analyzing process R-RANK. This is the bulk of the proof, and includes a technical lemma to work around the difficulties in linking cover time to delay sequence length. Finally, we reduce process R-RANK' to process R-RANK to show that the same bounds hold.

The rest of the paper is structured as follows. In Section 2, we establish helpful definitions and lemmas about random variables. In Section 3, we analyze the simple process. In Section 4, we analyze process R-RANK. In Appendix D, we analyze process R-RANK' via a reduction from process R-RANK. In Section 5, we show the existence of a logarithmic-degree graph on which process A-RANK and process A-RANK' each take $\omega(n)$ rounds to complete, establishing that the covering method does matter. Section 6 provides some concluding remarks.

2 Preliminaries

For the sake of brevity, we use the term ℓ -sequence to refer to a sequence of length ℓ . For any ℓ -sequence σ of elements of a given type, and any element x of the same type, we let $\sigma : x$ denote the $(\ell + 1)$ -sequence obtained by appending element x to σ .

For any nonnegative integer n and probability p , we let $X \sim \text{Bin}(n, p)$ denote that the random variable X has a binomial distribution with n trials and success probability p . We let $X \sim \text{Geo}(p)$ denote that the random variable X has a geometric distribution with success probability p . We let $X \sim \text{NegBin}(r, p)$ denote that the random variable X has a negative binomial distribution with r successes and success probability p . See Appendix A for proofs of the basic probabilistic lemmas stated below.

Lemma 2.1. *Let p denote an arbitrary probability, let ℓ denote an arbitrary nonnegative integer, and let $X \sim \text{NegBin}(\ell, p)$. For any integer j such that $1 \leq j \leq \ell$, let p_j denote an arbitrary probability such that $p_j \geq p$, let $Y_j \sim \text{Geo}(p_j)$, and let $Y = \sum_{1 \leq j \leq \ell} Y_j$. Then for any nonnegative integer i , $\Pr(X \geq i) \geq \Pr(Y \geq i)$.*

Lemma 2.2. *For any nonnegative integers r and n , and any probability p $\Pr(X < r) = \Pr(Y > n)$, where $X \sim \text{Bin}(n, p)$ and $Y \sim \text{NegBin}(r, p)$.*

Lemma 2.3. *For any integer $r \geq 2$, $\Pr(X \geq 2E[X]) = \Pr(X \geq 2r/p) \leq \exp(-r/8)$, where $X \sim \text{NegBin}(r, p)$.*

Lemma 2.4. *Let p be an arbitrary probability and let X be the sum of n independent Bernoulli variables X_1, \dots, X_n , where X_j has success probability $p_j \geq p$. Then $\Pr(X \leq np/2) \leq \exp(-np/12)$.*

Lemma 2.5. *Suppose we repeatedly throw balls independently and uniformly at random into n bins, and let the random variable X denote the number of throws required for every bin to receive at least n balls. Then X is $O(n^2)$ with high probability.*

Lemma 2.6. *Let j balls be thrown independently and uniformly at random into n bins. Let X denote the number of bins with at least one ball at the end of the experiment. Then, $\Pr(X \leq \min(n/4, j/4)) \leq \exp(-j/2)$.*

3 Process SELECT

Throughout all sections of this paper related to establishing upper bounds — that is, Sections 3 and 4 as well as Appendix C — we fix an arbitrary directed graph $G = (V, E)$ where $|V| = n > 0$. We let δ_{in} denote the minimum in-degree of any node, Δ_{in} denote the maximum in-degree of any node, and we let Δ_{out} denote the maximum out-degree of any node. For ease of exposition, we assume throughout the paper that $\delta_{in} > 0$. The edge set E is allowed to contain loops but not parallel edges. For any node v , we define $\Gamma_{in}(v)$ as $\{w \mid (w, v) \in E\}$. For any sequence of edges $\sigma = (u_1, v_1), \dots, (u_\ell, v_\ell)$, we define the two sequences of nodes $src(\sigma) = u_1, \dots, u_\ell$ and $dst(\sigma) = v_1, \dots, v_\ell$.

In this section, we analyze a simple stochastic process, process SELECT, defined as follows. Initially, we fix a positive integer r and independently assign each node in V a uniformly random integer rank from $\{1, \dots, r\}$. Process SELECT then proceeds in an infinite number of rounds, indexed from 1. In each round, one node is selected uniformly at random, with replacement. The following definitions are central to our analysis of this process.

We inductively define the notion of a *rank-sorted* node sequence as follows. For ℓ equal to 0 or 1, any ℓ -sequence of nodes is rank-sorted. For $\ell > 1$, an ℓ -sequence of nodes of the form $\sigma : v : v'$ is rank-sorted if $\sigma : v$ is rank-sorted and $rank(v) \leq rank(v')$.

For any node sequence σ , we inductively define a nonnegative integer $duration(\sigma)$ and a node sequence $select(\sigma)$ as follows. If σ is empty, then $duration(\sigma)$ is 0 and $select(\sigma)$ is empty. Otherwise, σ is of the form $\tau : v$ for some shorter node sequence τ and node v . Let i denote the least i such that $i > duration(\tau)$ and the node selected in round i belongs to $\Gamma_{in}(v)$. Let u denote the node selected in round i . Then we define $duration(\sigma)$ as i , and $select(\sigma)$ as $select(\tau) : u$.

See Appendix B for proofs of the following lemmas related to process SELECT.

Lemma 3.1. *For any ℓ -sequence of distinct nodes σ , $\Pr(\sigma \text{ is rank-sorted}) = \binom{\ell+r-1}{\ell} r^{-\ell}$.*

Lemma 3.2. *For any ℓ -sequence of nodes $\sigma = v_1, \dots, v_\ell$ and any nonnegative integer i , we have $\Pr(duration(\sigma) = i) \leq \Pr(X \geq i)$, where $X \sim \text{NegBin}(\ell, \frac{\delta_{in}}{n})$.*

Lemma 3.3. *For any ℓ -sequence of edges σ , $\Pr(select(dst(\sigma)) = src(\sigma)) \leq \delta_{in}^{-\ell}$.*

Lemma 3.4. *For any ℓ -sequence of edges σ and nonnegative integer i , the events $A = \text{“}dst(\sigma) \text{ is rank-sorted”}$, $B = \text{“}duration(dst(\sigma)) = i\text{”}$, and $C = \text{“}select(dst(\sigma)) = src(\sigma)\text{”}$ are mutually independent.*

Lemma 3.5. *Let σ be an ℓ -sequence of edges such that the nodes of $dst(\sigma)$ are distinct, let $X \sim \text{NegBin}(\ell, \frac{\delta_{in}}{n})$, let i be a nonnegative integer, and let events A , B and C be defined as in the statement of Lemma 3.4. Then $\Pr(A \cap B \cap C) \leq \binom{\ell+r-1}{\ell} \Pr(X \geq i) (r\delta_{in})^{-\ell}$.*

4 Process R-RANK

In the section we analyze an augmented version of process SELECT, referred to as Process R-RANK, in which we maintain a notion of a “covered subset” of the nodes. Initially, all of the nodes are uncovered. Process R-RANK then proceeds in rounds in exactly the same manner as process SELECT, with the additional step that in any given round, if one or more outgoing neighbors of the selected node are uncovered, we cover the uncovered outgoing neighbor with minimum rank. (As indicated in Section 1, ties are broken according to some arbitrary numbering of the nodes.)

Note that process R-RANK simply augments process SELECT with the additional notion of covered nodes; rank assignment and selections are performed in exactly the same manner in the two processes. Thus all of the definitions and lemmas presented in Section 3 are applicable to process R-RANK. The following additional definitions are useful for our analysis of process R-RANK.

The *cover time* of process R-RANK is defined as the number of rounds required to cover all of the nodes.

We inductively define the notion of a *linked* sequence of edges. For ℓ equal to 0 or 1, any ℓ -sequence of edges is linked. For $\ell > 1$, an ℓ -sequence of edges of the form $\sigma : (u, v) : (u', v')$ is linked if the $(\ell - 1)$ -sequence $\sigma : (u, v)$ is linked and (u, v') belongs to E .

For any node v , we define $\text{parent}(v)$ as follows. Let i denote the round in which node v is covered. If i is the first round in which some node in $\Gamma_{in}(v)$ is selected, then $\text{parent}(v)$ is defined to be *nil*. Otherwise, $\text{parent}(v)$ is the node covered in the first round prior to round i in which the selected node belongs to $\Gamma_{in}(v)$.

We inductively define the notion of a *chronological* sequence of nodes as follows. Any ℓ -sequence of nodes with $\ell \leq 1$ is chronological. An ℓ -sequence of nodes of the form $\sigma : v : v'$ is chronological if $\sigma : v$ is chronological and node v is covered before node v' .

We inductively define the notion of an *active* node sequence as follows. The empty node sequence is active. A singleton node sequence consisting of the node v is active if $\text{parent}(v) = \text{nil}$. An ℓ -sequence of nodes of the form $\sigma : v : v'$ is active if $\sigma : v$ is active and $\text{parent}(v') = v$.

We call an ℓ -sequence of edges σ *active* if $\text{dst}(\sigma)$ is active and $\text{select}(\text{dst}(\sigma)) = \text{src}(\sigma)$.

We call an ℓ -sequence of edges σ *i-active* if it is active and either $\ell = i = 0$ or $\ell > 0$, σ is of the form $\sigma : (u, v)$, and v is covered in round i .

Lemma 4.1. *For any nonnegative integer ℓ , there are at most $n\Delta_{out}^\ell\Delta_{in}^{\ell-1}$ linked ℓ -sequences of edges.*

Proof. We proceed by induction on ℓ , treating $\ell = 0$ and $\ell = 1$ as base cases. For $\ell = 0$, the empty sequence is the only linked 0-sequence, and the claim holds since $n/\Delta_{in} \geq 1$. (Note that Δ_{in} is at most n since we do not allow parallel edges.) For $\ell = 1$, the number of linked 1-sequences is at most $|E| \leq n\Delta_{out}$.

Now let ℓ be greater than 1 and inductively assume that the number of linked $(\ell - 1)$ -sequences of edges is at most $n\Delta_{out}^{\ell-1}\Delta_{in}^{\ell-2}$. Recall that any linked ℓ -sequence of edges is of the form $\sigma : (u, v) : (u', v')$ where the $(\ell - 1)$ -sequence of edges $\sigma : (u, v)$ is linked and (u, v') belongs to E . Observe that for any linked $(\ell - 1)$ -sequence of edges $\sigma : (u, v)$, there are at most Δ_{out} nodes v' such that (u, v') belongs to E , and for each such choice of v' , there are at most Δ_{in} nodes u' such that (u', v') belongs to E . Thus the number of linked ℓ -sequences is at most $\Delta_{out}\Delta_{in}$ times the number of linked $(\ell - 1)$ -sequences, and the desired bound follows from the induction hypothesis. \square

Lemma 4.2. *Suppose we run two instances of process R-RANK in parallel using the same random ranks and the same sequence of random selections, but in the second instance, we allow an arbitrary subset of the covered nodes to be uncovered after each round. Then the cover time of the first instance is at most the cover time of the second instance.*

Proof. By a straightforward induction on the number of rounds, at all times, the set of covered nodes in the first instance contains the set of covered nodes in the second instance. The claim of the lemma follows. \square

Lemma 4.3. *For any rank assignment, the expected cover time of process R-RANK is $O(n^2)$.*

Proof. It follows from Lemma 2.5 that cover time is $O(n^2)$ with high probability since in that time each vertex would have been selected at least n times implying that all its neighbors are covered.

We can then consider a modified version of process R-RANK in which the infinite sequence of rounds is partitioned into epochs of $O(n^2)$ rounds, and where at the end of each epoch, if the nodes are not all covered, all nodes are uncovered before proceeding to the next epoch. Since each epoch covers all the nodes with high probability, the expected cover time of this modified version of process R-RANK is $O(n^2)$. By Lemma 4.2, for any rank assignment, the expected cover time of process R-RANK is $O(n^2)$. \square

Lemma 4.4. *Assume that node v is covered in round i and let u be the node selected in round i . Then there is an i -active edge sequence σ terminating in edge (u, v) and such that $\text{duration}(\text{dst}(\sigma)) = i$.*

Proof. Observe that u belongs to $\Gamma_{in}(v)$. Also observe that if $\text{parent}(v) = \text{nil}$, the singleton node sequence v is active with $\text{duration}(v) = i$, and hence the singleton edge sequence $\sigma = (u, v)$ is i -active with $\text{duration}(\text{dst}(\sigma)) = i$.

We prove the claim by induction on i . For $i = 1$, it must be true that $\text{parent}(v) = \text{nil}$ and the observations of the previous paragraph prove the claim.

For $i > 1$, if $\text{parent}(v) = \text{nil}$, the observations once again prove the claim. Otherwise, $\text{parent}(v) = v'$ where v' is some node covered in round $j < i$. Call the node selected in round j , u' . Since $j < i$, we can inductively assume that there is a j -active edge sequence, call it τ , terminating in edge (u', v') and such that $\text{duration}(\text{dst}(\tau)) = j$. Since τ is active, the node sequence $\text{dst}(\tau)$ is active and $\text{select}(\text{dst}(\tau)) = \text{src}(\tau)$. Let $\sigma = \tau : (u, v)$. Thus $\text{src}(\sigma) = \text{src}(\tau) : u$ and $\text{dst}(\sigma) = \text{dst}(\tau) : v$. Since $\text{parent}(v) = v'$ and $\text{dst}(\tau)$ is an active node sequence terminating in node v' , $\text{dst}(\sigma)$ is active. Since $\text{duration}(\text{dst}(\tau)) = j$, $\text{select}(\text{dst}(\tau)) = \text{src}(\tau)$, u was selected in round i , and i is the least integer greater than j such that the node selected in round i belongs to $\Gamma_{in}(v)$, we have $\text{duration}(\text{dst}(\sigma)) = i$ and $\text{select}(\text{dst}(\sigma)) = \text{src}(\sigma)$. Since $\text{dst}(\sigma)$ is active and $\text{select}(\text{dst}(\sigma)) = \text{src}(\sigma)$, σ is active. Since σ is active and node v is covered in round i , σ is i -active. Thus the edge sequence σ satisfies all of the requirements of the lemma. \square

Lemma 4.5. *Any active node sequence is rank-sorted, chronological, and consists of distinct nodes.*

Proof. Note that any chronological node sequence consists of distinct nodes. Thus, in what follows, it is sufficient to prove that any active node sequence is rank-sorted and chronological.

We proceed by induction on the length of the sequence. For the base case, note that any node sequence of length 0 or 1 is rank-sorted and chronological. For the induction step, consider an active node sequence σ of the form $\tau : v : v'$. Since σ is active, $\tau : v$ is active and $\text{parent}(v') = v$. Since $\tau : v$ is active, the induction hypothesis implies that it is also rank-sorted and chronological. Since $\text{parent}(v') = v$, $\text{rank}(v) \leq \text{rank}(v')$ and v is covered before v' . Hence σ is rank-sorted and chronological. \square

Lemma 4.6. *For any nonempty active edge sequence σ , if the last edge in σ is (u, v) , then node v is covered in round $\text{duration}(\text{dst}(\sigma))$ and node u is selected in the same round.*

Proof. We prove the claim by induction on the length of the active edge sequence σ .

If σ consists of a single edge (u, v) , then by the definition of an active edge sequence, the singleton node sequence $\text{dst}(\sigma)$ is active and $\text{select}(\text{dst}(\sigma)) = \text{src}(\sigma)$. Since $\text{dst}(\sigma)$ is active, $\text{parent}(v) = \text{nil}$, that is, node v is covered in the first round in which a node in $\Gamma_{in}(v)$ is selected, which is round $\text{duration}(\text{dst}(\sigma))$. Since $\text{select}(\text{dst}(\sigma)) = \text{src}(\sigma)$, node u is selected in the same round.

Now assume that σ is an active edge sequence of the form $\tau : (u, v)$, where τ is of the form $\tau' : (u', v')$. Since σ is active, the node sequence $\text{dst}(\sigma)$ is active and $\text{select}(\text{dst}(\sigma)) = \text{src}(\sigma)$. It follows that $\text{dst}(\tau)$ is active and $\text{select}(\text{dst}(\tau)) = \text{src}(\tau)$, that is, τ is also active. Since τ is active and shorter than σ , we can

inductively assume that v' is covered in round $\text{duration}(\text{dst}(\tau))$ and node u' is selected in the same round. Since $\text{dst}(\sigma)$ is active, $\text{parent}(v) = v'$, that is, node v is covered in the first round after round $\text{duration}(\text{dst}(\tau))$ in which a node in $\Gamma_{in}(v)$ is selected. Applying the definition of $\text{duration}(\text{dst}(\sigma))$, we conclude that v is covered in round $\text{duration}(\text{dst}(\sigma))$. Since $\text{select}(\text{dst}(\sigma)) = \text{src}(\sigma)$, node u is selected in the same round. \square

Lemma 4.7. *If σ is an active sequence of edges, then σ is linked.*

Proof. We proceed by induction on the length of σ . If the length of σ is 0 or 1, then σ is linked by definition.

Now assume that σ is an edge sequence of the form $\tau : (u, v)$, where τ is of the form $\tau' : (u', v')$ and σ is active. Since σ is active, $\text{dst}(\sigma)$ is active. Since $\text{dst}(\sigma)$ is active, $\text{dst}(\tau)$ is also active. Since $\text{dst}(\tau)$ is active and τ is shorter than σ , we can inductively assume that τ is linked. Therefore, in order to establish that σ is linked, it is sufficient to prove that (u', v) is an edge. Since $\text{dst}(\sigma)$ is active, $\text{parent}(v) = v'$. Hence, letting i denote the round in which node v is covered, we find that node v' is covered in the first round prior to round i in which the selected node belongs to $\Gamma_{in}(v)$. By Lemma 4.6, v' is covered in a round in which node u' is selected. Thus u' belongs to $\Gamma_{in}(v)$, that is, (u', v) is an edge, as required. \square

Lemma 4.8. *If an edge sequence σ is i -active then $\text{duration}(\text{dst}(\sigma)) = i$.*

Proof. If σ is empty, then the claim holds since $i = 0$ and $\text{duration}(\text{dst}(\sigma)) = 0$. Otherwise, σ is of the form $\tau : (u, v)$, and by the definition of an i -active edge sequence, v is covered in round i . By Lemma 4.6, v is covered in round $\text{duration}(\text{dst}(\sigma))$, so $\text{duration}(\text{dst}(\sigma)) = i$. \square

Lemma 4.9. *For any ℓ -sequence of edges σ , and any nonnegative integer i , the probability that σ is i -active is at most $\binom{\ell+r-1}{\ell} \Pr(X \geq i) (r\delta_{in})^{-\ell}$, where $X \sim \text{NegBin}(\ell, \frac{\delta_{in}}{n})$.*

Proof. If the nodes in $\text{dst}(\sigma)$ are not all distinct, then $\Pr(\sigma \text{ is } i\text{-active}) = 0$ by Lemma 4.5 and the claimed inequality holds since the right-hand side is nonnegative.

Now assume that $\text{dst}(\sigma)$ consists of distinct nodes, and let events A , B , and C be as defined in the statement of Lemma 3.4. Below we prove that if σ is i -active, then events A , B , and C all occur. The claimed inequality then follows by Lemma 3.5.

Assume that σ is i -active. Thus event B occurs by Lemma 4.8. Furthermore, σ is active, so $\text{dst}(\sigma)$ is active and event C occurs by the definition of active edge sequence. Since $\text{dst}(\sigma)$ is active, event A occurs by Lemma 4.5. \square

Lemma 4.10. *For any nonnegative integers i and ℓ , the probability that some ℓ -sequence of edges is i -active is at most*

$$n\Delta_{out}^{\ell}\Delta_{in}^{\ell-1}\binom{\ell+r-1}{\ell}\frac{\Pr(X \geq i)}{(r\delta_{in})^{\ell}}$$

where $X \sim \text{NegBin}(\ell, \frac{\delta_{in}}{n})$.

Proof. By Lemma 4.7, if an edge sequence σ is not linked, then $\Pr(\sigma \text{ is } i\text{-active}) = 0$. The union bound then implies that the probability some ℓ -sequence of edges is i -active is at most the number of linked ℓ -sequences of edges multiplied by the maximum probability that any particular ℓ -sequence is i -active. The lemma follows by Lemmas 4.1 and 4.9. \square

Lemma 4.11. *For nonnegative integers i , ℓ , and r such that $i \geq 64n \max(\Delta_{out}\Delta_{in}/\delta_{in}^2, (\ln n)/\delta_{in})$ and $r \geq \min(\lceil 2e^2\Delta_{out}\Delta_{in}/\delta_{in} \rceil, \ell)$, we have*

$$\Delta_{out}^{\ell}\Delta_{in}^{\ell-1}\binom{\ell+r-1}{\ell}\frac{\Pr(X \geq i)}{(r\delta_{in})^{\ell}} \leq \exp(-i\delta_{in}/(32n))$$

where $X \sim \text{NegBin}(\ell, \frac{\delta_{in}}{n})$.

Proof. See Appendix C. □

Lemma 4.12. *For $r = \min(\lceil 2e^2 \Delta_{out} \Delta_{in} / \delta_{in} \rceil, n)$, every active edge sequence is $O(n \max(\Delta_{out} \Delta_{in} / \delta_{in}^2, (\log n) / \delta_{in}))$ -active with high probability.*

Proof. Let c denote an arbitrary positive real greater than or equal to 1, and let i denote the positive integer $\lceil 64cn \max(\Delta_{out} \Delta_{in} / \delta_{in}^2, (\log n) / \delta_{in}) \rceil$.

For any nonnegative integer j , let p_j denote the probability that there is a j -active edge sequence. Any j -active edge sequence σ is active, so the associated node sequence $dst(\sigma)$ is active. It follows from Lemma 4.5 that any j -active sequence has length at most n . In other words, $\ell \leq n$ for any j -active ℓ -sequence of edges. Furthermore, if $j > 0$ then the length of a j -active sequence is nonzero. Since any j -active ℓ -sequence of edges must have $\ell \leq n$, the condition $r = \min(\lceil 2e^2 \Delta_{out} \Delta_{in} / \delta_{in} \rceil, n)$ is enough for us to apply Lemmas 4.10 and 4.11. Thus, the union bound and Lemmas 4.10 and 4.11 imply $p_j \leq n^2 \exp(-j\delta_{in}/(64n))$ for $j > i$.

Let p denote the probability that there is a j -active edge sequence for some $j \geq i$. By the union bound, $p \leq \sum_{j \geq i} p_j$. Using the upper bound on p_j derived in the preceding paragraph, we find that p is upper bounded by an infinite geometric sum with initial term $n^2 \exp(-i\delta_{in}/(64n))$ and ratio $\exp(-\delta_{in}/(64n))$. Thus

$$\begin{aligned} p &= O((n^3 / \delta_{in}) \exp(-i\delta_{in}/(64n))) \\ &= O(n^3 \exp(-c \max(\Delta_{out} \Delta_{in} / \delta_{in}, \log n))) \\ &= O(n^{3-c}). \end{aligned}$$

By setting c to a sufficiently large positive constant, we can drive p below any desired inverse polynomial threshold. The claim of the lemma follows. □

Lemma 4.13. *For $r = \min(\lceil 2e^2 \Delta_{out} \Delta_{in} / \delta_{in} \rceil, n)$, the cover time of process R-RANK is $O(n \max(\Delta_{out} \Delta_{in} / \delta_{in}^2, (\log n) / \delta_{in}))$ with high probability, and the same asymptotic bound holds for the expected cover time.*

Proof. The high probability claim is immediate from Lemmas 4.4 and 4.12.

Thus, for $c > 3$ with probability at least $1 - 1/n^c$ the cover time is $O(n \max(\Delta_{out} \Delta_{in} / \delta_{in}^2, (\log n) / \delta_{in}))$ and with probability at most $1/n^c$ we enter a bad case where we cannot apply our bound. In the bad case, we will use the $O(n^2)$ expected cover time bound provided by Lemma 4.3. Since $1/n^c \cdot O(n^2) = O(1)$, we have shown the desired result. □

Theorem 1. *If both Δ_{in} and Δ_{out} are $O(\delta_{in})$, there is an $r = O(\delta_{in})$ such that the cover time of process R-RANK is $O(n \max(1, (\log n) / \delta_{in}))$ with high probability, and the same asymptotic bound holds for the expected cover time.*

Proof. Immediate from Lemma 4.13. □

The result of Theorem 1 matches the lower bound proved by Alon for process MIN and is thus optimal [2].

Note that as r tends to infinity, the behavior of process R-RANK converges to that of process P-RANK. Thus, the bounds of Theorem 1 also hold for process P-RANK.

5 Lower Bounds

While the full proofs of the two theorems stated in this appendix are rather lengthy, the main ideas are straightforward. We summarize these main ideas in the two proof sketches that follow. The main technical tools employed in the full proofs are Chernoff bounds and Azuma's inequality (see, e.g., [6, 3]). Note that our lower bounds hold even if we restrict attention to the special class of directed graphs where edge (u, v) is present if and only if edge (v, u) is present; below we refer to such graphs as undirected.

Theorem 2. For all n , there is an n -node undirected graph G in which each node has degree $\Theta(\log n)$, and an assignment of ranks 1 through n to the nodes of G , such that process A-RANK has cover time $\Omega(n \sqrt{(\log n)/\log \log n}) = \omega(n)$.

Proof sketch: Fix n and construct G as follows. First, partition the n nodes into ℓ levels, numbered from 0 to $\ell - 1$, so that the following conditions hold: level 0 contains $\Theta(n)$ nodes; the ratio of the number of nodes at level $i - 1$ to the number of nodes at level i is approximately $\xi = \sqrt{(\lg n)/\lg \lg n}$, $1 \leq i < \ell$; level $\ell - 1$ is the only level with fewer than \sqrt{n} nodes. Thus $\ell = \Theta((\log n)/\log \log n)$. Assign ranks 1 through n to the nodes in such a way that nodes on lower-numbered levels have lower ranks. For each node u at level i , select $\lg n$ nodes at random from each of levels i and $i - 1$ (with replacement), and add an edge from u to each selected node. (If node u is at level 0, then select $2 \lg n$ nodes from level 0.)

We call a level *crowded* if more than half of the nodes on that level are covered. In what follows, let i be an arbitrary level, but for the sake of simplifying the exposition, assume that $0 < i < \ell - 1$. In any round in which a node in level i is covered, the selected node belongs to level $i - 1$, i , or $i + 1$. If the selected node belongs to level $i - 1$ (resp., i , $i + 1$) we refer to such a node covering as *ascending* (resp., *lateral*, *descending*). The main observation underlying our proof is that until level $i - 1$ is crowded, only a negligible fraction of the node coverings at level i are ascending or lateral. (This is because a node selected at level i or $i - 1$ is likely to have an uncovered neighbor at a level less than i .) Furthermore, a trivial upper bound on the number of descending node coverings at level i is given by the number of node selections at level $i + 1$, and we can upper bound the latter quantity using a standard Chernoff bound argument.

Using the preceding ideas, we obtain an upper bound the rate of coverage of nodes at level i until level $i - 1$ is crowded. Once level $i - 1$ is crowded, we upper bound the rate of node coverings at level i by pessimistically assuming that every selection in level $i - 1$, i , or $i + 1$ results in a node covering at level i . By applying these upper bounds on the rate of node coverage, we are able to prove by induction on i that, with high probability, level i is not crowded before round $\frac{n}{4} + \frac{ni}{4\xi}$. The theorem follows by setting $i = \ell - 1 = \Theta((\log n)/\log \log n)$. \square

Theorem 3. For all n , there is an n -node undirected graph G in which each node has degree $\Theta(\log n)$, and an assignment of ranks 1 through n to the nodes of G , such that process A-RANK' has cover time $\Omega(n \log \log n) = \omega(n)$.

Proof sketch: The proof of this theorem is similar to that of Theorem 2. The graph G is defined in the same way except that the ratio ξ of the number of nodes between successive levels is taken to be approximately $(\lg n)^{1/4}$, and we restrict the number of levels ℓ to $\Theta((\log n)^{3/8} \log \log n)$.

For rounds in which a node is covered that is different from the selected node, we refer to the node covering as ascending, lateral, or descending as in the proof of Theorem 2. If the covered node is equal to the selected node, we refer to the node covering as stationary.

We now call a level crowded if more than a $1 - (\lg n)^{-1/8}$ fraction of the nodes on that level are covered. The motivation for this change is that stationary node coverings quickly cover a significant fraction of the nodes in each level.

The proof now proceeds in much the same manner as the proof of Theorem 2, with the following two major differences. First, for any given level i , we now need to upper bound the number of stationary node coverings observed at level i within a given number of rounds. Such an upper bound is given by the number of distinct nodes selected at level i , which is not difficult to characterize. Second, the threshold on the number of rounds after which we claim that, with high probability, level i is not crowded, reduces to

$$\frac{n}{4} + \frac{ni}{4\xi(\lg n)^{1/8}},$$

$0 \leq i < \ell$. The theorem follows by setting $i = \ell - 1 = \Theta((\log n)^{3/8} \log \log n)$. \square

6 Concluding Remarks

As indicated in the introduction, we conjecture that there is a positive constant ϵ such that for any logarithmic-degree graph, each of the first $\lfloor \epsilon n \rfloor$ rounds of process R-RANK covers a node with high probability. A proof of this conjecture would provide load balance guarantees for a wide class of DHTs. It would also be interesting to see if the proof ideas used in this paper can be used to provide similarly optimal bounds for process UNI.

References

- [1] M. Adler, E. Halperin, R. Karp, and V. Vazirani. A stochastic process on the hypercube with applications to peer-to-peer networks. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC)*, pages 575–584, 2003.
- [2] N. Alon. Problems and results in extremal combinatorics, II. Manuscript, 2004.
- [3] N. Alon and J. H. Spencer. *The Probabilistic Method*. Wiley, New York, NY, 1991.
- [4] Stasys Jukna. *Extremal Combinatorics*, pages 224–225. Springer, 2001.
- [5] F. T. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, and Hypercubes*, pages 547–556. Morgan-Kaufmann, San Mateo, CA, 1991.
- [6] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, Cambridge, UK, 1995.
- [7] A. G. Ranade. How to emulate shared memory. *Journal of Computer and System Sciences*, 42:307–326, 1991.

A Proofs of Basic Probabilistic Lemmas

Proof of Lemma 2.1. Note that if $p_j = p$ for all j , then the random variables X and Y have the same distribution. Furthermore, increasing any of the p_j 's can only decrease Y . \square

Proof of Lemma 2.2. The random variables X and Y can be seen as different views of the same experiment where we successively flip coins with probability of success p . With Y , we ask “How many flips are required for r successes?” With X , we ask “How many successes are in the first n flips?” In this experiment, the event of seeing less than r successes in the first n flips ($X < r$) corresponds to the event that we have to wait more than n flips for the first r successes ($Y > n$). This gives the result. \square

Proof of Lemma 2.3. Let $j = \lfloor \frac{2r}{p} \rfloor - 1$ and let $Y \sim \text{Bin}(j, p)$. By Lemma 2.2, we know that $\Pr(X \geq \frac{2r}{p}) \leq \Pr(X \geq \lfloor \frac{2r}{p} \rfloor) = \Pr(X > \lfloor \frac{2r}{p} \rfloor - 1) = \Pr(Y < r) = \Pr(Y \leq r - 1)$.

$$\begin{aligned} \Pr\left(Y \leq \frac{j p}{2}\right) &= \Pr\left(Y \leq r - (\eta + 1)\frac{p}{2}\right) \\ &= \Pr(Y \leq r - 1) \end{aligned}$$

where $\frac{2r}{p} = \lfloor \frac{2r}{p} \rfloor + \eta$ and the last equality holds because $0 < (\eta + 1)\frac{p}{2} < 1$.

Recall the Chernoff bounds in the form

$$\Pr(Y \leq (1 - \lambda)jp) \leq \exp\left(\frac{-\lambda^2 jp}{2}\right)$$

for $0 < \lambda < 1$ (see [4, 3]).

We apply this bound with $\lambda = \frac{1}{2}$ to get

$$\begin{aligned} \Pr(Y \leq r - 1) &= \Pr\left(Y \leq \frac{jp}{2}\right) \\ &\leq \exp\left(\frac{-jp}{8}\right) \\ &= \exp\left(\frac{-2r + (\eta + 1)p}{8}\right) \\ &\leq \exp\left(\frac{-r}{8}\right) \end{aligned}$$

where η is as previously defined and the last inequality holds because $r \geq 2$. □

Proof of Lemma 2.4. The result follows from Chernoff bounds (see, e.g., [4, 3]). □

Proof of Lemma 2.5. The result follows from Lemma 2.4. □

Proof of Lemma 2.6. Let $[n] = \{1, 2, \dots, n\}$. Suppose $\min\left(\frac{n}{4}, \frac{j}{4}\right) = k$. Let $S \subseteq [n]$ be a particular subset of size k . Then,

$$\Pr(\text{all balls land in } S) \leq \left(\frac{k}{n}\right)^j$$

Thus,

$$\begin{aligned} \Pr(X \leq k) &= \Pr\left(\bigcup_{S \text{ s.t. } |S|=k} \text{all balls land in } S\right) \\ &\leq \binom{n}{k} \left(\frac{k}{n}\right)^j \\ &\leq \left(\frac{en}{k}\right)^k \left(\frac{k}{n}\right)^j \\ &= \left(\frac{en}{k}\right)^k \left(\frac{k}{n}\right)^{\frac{j}{2}} \left(\frac{k}{n}\right)^{\frac{j}{2}} \end{aligned}$$

Now, since $\frac{j}{2} \geq 2k$ and since $k \geq \frac{n}{4}$ implies $\frac{k}{n} < \frac{1}{4}$ implying $\frac{ek}{n} < 1$

$$\begin{aligned} \Pr(X \leq k) &\leq \left(\frac{ek}{n}\right)^k \left(\frac{k}{n}\right)^{\frac{j}{2}} \\ &\leq \left(\frac{ek}{n}\right)^k \left(\frac{1}{4}\right)^{\frac{j}{2}} \\ &\leq \exp\left(-\frac{j}{2}\right) \end{aligned}$$

□

B Proofs Related to Process SELECT

Proof of Lemma 3.1. There are $\binom{\ell+r-1}{\ell}$ ways that ranks can be assigned to the ℓ distinct nodes so that the resulting ℓ -sequence is rank-sorted. The result follows since each such assignment occurs with probability $r^{-\ell}$. \square

Proof of Lemma 3.2. We proceed by proving that

$$\Pr(\text{duration}(\sigma) = i) = \Pr\left(\sum_{k=1}^{\ell} Y_k = i\right)$$

where $Y_k \sim \text{Geo}\left(\frac{d_k}{n}\right)$ and d_k denotes the in-degree of v_k . The desired bound then follows by Lemma 2.1.

We prove the foregoing claim by induction on ℓ . If $\ell = 0$, the claim holds since $\text{duration}(\sigma) = \sum_{k=1}^{\ell} Y_k = 0$.

For $\ell > 0$, we let τ denote the node sequence $v_1, \dots, v_{\ell-1}$ and assume inductively that

$$\Pr(\text{duration}(\tau) = i) = \Pr\left(\sum_{k=1}^{\ell-1} Y_k = i\right)$$

Thus,

$$\begin{aligned} \Pr(\text{duration}(\sigma) = i) &= \sum_{j=0}^{i-1} \Pr(\text{duration}(\tau) = j) \cdot \Pr(\text{duration}(\sigma) - \text{duration}(\tau) = i - j \mid \text{duration}(\tau) = j) \\ &= \sum_{j=0}^{i-1} \Pr(\text{duration}(\tau) = j) \cdot \Pr(\text{duration}(\sigma) - \text{duration}(\tau) = i - j) \\ &= \sum_{j=0}^{i-1} \Pr(\text{duration}(\tau) = j) \cdot \Pr(Y_{\ell} = i - j) \\ &= \sum_{j=0}^{i-1} \Pr\left(\sum_{k=1}^{\ell-1} Y_k = j\right) \cdot \Pr(Y_{\ell} = i - j) \\ &= \Pr\left(\sum_{k=1}^{\ell-1} Y_k = i\right). \end{aligned}$$

The second equality comes from the fact that future selections are independent of past selections. The third equality comes from the fact that the number of rounds elapsed from any given time to the next selection in $\Gamma_{in}(v_{\ell})$ is distributed as Y_{ℓ} . \square

Proof of Lemma 3.3. We proceed by induction on ℓ . For $\ell = 0$, $\Pr(\text{select}(\text{dst}(\sigma)) = \text{src}(\sigma)) = 1 = \delta_{in}^0$ since we have assumed that $\delta_{in} > 0$.

For $\ell > 0$, σ can be written in the form $\tau : (u, v)$, where we inductively assume that the claim of the lemma holds for τ . Let A denote the event that the first node selected in $\Gamma_{in}(v)$ after round $\text{duration}(\text{dst}(\tau))$ is u . We have

$$\begin{aligned} \Pr(\text{select}(\text{dst}(\sigma)) = \text{src}(\sigma)) &= \Pr(\text{select}(\text{dst}(\tau)) = \text{src}(\tau)) \cdot \Pr(A \mid \text{select}(\text{dst}(\tau)) = \text{src}(\tau)) \\ &= \Pr(\text{select}(\text{dst}(\tau)) = \text{src}(\tau)) \cdot \Pr(A) \\ &\leq \delta_{in}^{-\ell}. \end{aligned}$$

The second step follows from the independence of the events A and $\text{select}(\text{dst}(\sigma')) = \text{src}(\sigma')$ (They are independent since future selections are independent of past selections). The third step follows from the induction hypothesis and the observation that $\Pr(A)$ is equal $1/\Gamma_{in}(v)$, which is at most $1/\delta_{in}$. \square

Proof of Lemma 3.4. Note that event A depends only on the rank assignments, while events B and C depend only on the selections. Thus A is independent of events B and C . Below we argue that events B and C are independent.

Let $\sigma = (u_1, v_1), \dots, (u_\ell, v_\ell)$ and let σ_j denote the length- j prefix of σ , $0 \leq j \leq \ell$. Define a selection to be j -special, $1 \leq j \leq \ell$, if it is the first selection after round $\text{duration}(\sigma_{j-1})$ in $\Gamma_{in}(v_j)$. A selection is special if it is j -special for some j . Note that event B depends only on the timing of the special events; in particular, B occurs if and only if the ℓ -special selection occurs in round i . Suppose we run process SELECT, but at each step, instead of revealing the selected node, we reveal only whether the selection is special. This information is sufficient to determine the unique i for which B occurs, but does not bias the distribution of $\text{select}(\text{dst}(\sigma))$. Since event C only depends on $\text{select}(\text{dst}(\sigma))$, it is independent of B . \square

Proof of Lemma 3.5. By Lemma 3.1, $\Pr(A) \leq \binom{\ell+r-1}{\ell} r^{-\ell}$. By Lemma 3.2, $\Pr(B) \leq \Pr(X \geq i)$. By Lemma 3.3, $\Pr(C) \leq \delta_{in}^{-\ell}$. The claim then follows by Lemma 3.4. \square

C Proof of Lemma 4.11

First, we show that the LHS of the claimed inequality is a nonincreasing function of r .

It is sufficient to prove that the expression $\binom{\ell+r-1}{\ell} r^{-\ell}$ is a nonincreasing function of r . Fix ℓ and let $f(r)$ denote the preceding expression. Note that

$$\begin{aligned} \frac{f(r+1)}{f(r)} &= \frac{r+\ell}{r} \left(\frac{r}{r+1} \right)^\ell \\ &= \left(1 + \frac{\ell}{r} \right) \left(1 + \frac{1}{r} \right)^{-\ell} \\ &\leq 1, \end{aligned}$$

where the last inequality holds since the binomial theorem implies $(1 + \frac{1}{r})^\ell \geq 1 + \frac{\ell}{r}$.

Since we have established that the LHS of the claimed inequality is a nonincreasing function of r , we can assume in what follows that $r = \min(\lceil 2e^2 \Delta_{out} \Delta_{in} / \delta_{in} \rceil, \ell)$.

Let us rewrite the LHS of the claimed inequality as $\lambda \cdot \Pr(X \geq i)$, where

$$\begin{aligned} \lambda &= \Delta_{out}^\ell \Delta_{in}^{\ell-1} \binom{\ell+r-1}{\ell} (r\delta_{in})^{-\ell} \\ &\leq \Delta_{out}^\ell \Delta_{in}^\ell \left(\frac{e(\ell+r-1)}{\ell r \delta_{in}} \right)^\ell \\ &\leq \left(\frac{e \Delta_{out} \Delta_{in} (\ell+r)}{\ell r \delta_{in}} \right)^\ell. \end{aligned} \tag{1}$$

We begin by establishing two useful upper bounds on λ , namely, Equations (2) and (4) below.

If $r = \lceil 2e^2 \Delta_{out} \Delta_{in} / \delta_{in} \rceil$, then since $r = \min(\lceil 2e^2 \Delta_{out} \Delta_{in} / \delta_{in} \rceil, \ell)$, we have $r \leq \ell$. Substituting the value of r in Equation (1), we find that

$$\begin{aligned} \lambda &\leq \left(\frac{e(\ell + r)}{2e^2 \ell} \right)^\ell \\ &\leq \left(\frac{2e\ell}{2e^2 \ell} \right)^\ell \\ &\leq e^{-\ell}. \end{aligned} \quad (2)$$

If $r = \ell$, then Equation (1) implies

$$\lambda \leq \left(\frac{2e\Delta_{out}\Delta_{in}}{\ell\delta_{in}} \right)^\ell. \quad (3)$$

Let $h(\ell)$ denote the natural logarithm of the RHS of Equation (3), that is, $h(\ell) = \ell \ln(2e\Delta_{out}\Delta_{in}/(\ell\delta_{in}))$. Using elementary calculus, it is straightforward to prove that the derivative of $h(\ell)$ with respect to ℓ is positive for $\ell < 2\Delta_{out}\Delta_{in}/\delta_{in}$, is 0 when $\ell = 2\Delta_{out}\Delta_{in}/\delta_{in}$, and is negative for $\ell > 2\Delta_{out}\Delta_{in}/\delta_{in}$. It follows that $h(\ell) \leq h(2\Delta_{out}\Delta_{in}/\delta_{in}) = 2\Delta_{out}\Delta_{in}/\delta_{in}$. Since \ln is monotonic, the RHS of Equation (3) is also maximized when $\ell = 2\Delta_{out}\Delta_{in}/\delta_{in}$. Combining this result with Equation (2), we have that regardless of the value of r

$$\lambda \leq \exp(2\Delta_{out}\Delta_{in}/\delta_{in}). \quad (4)$$

(Note that $\exp(2\Delta_{out}\Delta_{in}/\delta_{in}) \geq 1$ and Equation (2) implies $\lambda \leq 1$ when $r = \lceil 2e^2 \Delta_{out} \Delta_{in} / \delta_{in} \rceil$.)

We are now ready to proceed with the proof of the lemma. We consider the two cases $\ell > \lceil i\delta_{in}/(2n) \rceil$ and $\ell \leq \lceil i\delta_{in}/(2n) \rceil$ separately.

If $\ell > \lceil i\delta_{in}/(2n) \rceil$, then $\ell > 2ec \max(\Delta_{out}\Delta_{in}/\delta_{in}, \ln n)$ where $c = 16/e > e$. In specific, we have $\ell > \lceil 2e^2 \Delta_{out} \Delta_{in} / \delta_{in} \rceil$, and thus $r = \lceil 2e^2 \Delta_{out} \Delta_{in} / \delta_{in} \rceil$. It follows from Equation (2) that $\lambda \leq e^{-\ell} \leq \exp(-i\delta_{in}/(2n)) \leq \exp(-i\delta_{in}/(64n))$, and hence the claim holds since $\Pr(X \geq i) \leq 1$.

Now assume that $\ell \leq \lceil i\delta_{in}/(2n) \rceil$. Let $Y \sim \text{NegBin}\left(\lfloor \frac{i\delta_{in}}{2n} \rfloor, \frac{\delta_{in}}{n}\right)$ and $Z \sim \text{NegBin}\left(\lfloor \frac{i\delta_{in}}{2n} \rfloor - \ell, \frac{\delta_{in}}{n}\right)$. By the definition of the negative binomial distribution, $\Pr(Y \geq i) = \Pr(X + Z \geq i)$. And, since Z is nonnegative, $\Pr(X + Z \geq i) \geq \Pr(X \geq i)$. Thus

$$\Pr(X \geq i) \leq \Pr(Y \geq i). \quad (5)$$

Since $E[Y] \leq \frac{i}{2}$ and $\lfloor i\delta_{in}/(2n) \rfloor \geq \lfloor 32 \max(\Delta_{out}\Delta_{in}/\delta_{in}, \ln n) \rfloor > 2$, Lemma 2.3 implies $\Pr(Y \geq i) \leq \Pr(Y \geq 2E[Y]) \leq \exp\left(\frac{-i\delta_{in}}{16n} + \frac{1}{8}\right)$. The claim follows since

$$\begin{aligned} \lambda \cdot \Pr(X \geq i) &\leq \exp\left(\frac{2\Delta_{out}\Delta_{in}}{\delta_{in}}\right) \cdot \Pr(Y \geq i) \\ &\leq \exp\left(\frac{-i\delta_{in}}{16n} + \frac{1}{8} + \frac{2\Delta_{out}\Delta_{in}}{\delta_{in}}\right) \\ &\leq \exp\left(\frac{-i\delta_{in}}{32n} + \frac{1}{8}\right) \\ &\leq \exp\left(\frac{-i\delta_{in}}{64n}\right). \end{aligned}$$

(The first step follows from Equations (4) and (5). For the third step and fourth steps, note that the assumption $i \geq 64n \max(\Delta_{out}\Delta_{in}/\delta_{in}^2, (\log n)/\delta_{in})$ implies $i\delta_{in}/(32n) \geq 2\Delta_{out}\Delta_{in}/\delta_{in}$ and $i\delta_{in}/(64n) \geq 1/8$, respectively.)

D Process R-RANK'

In this section we analyze a biased version of process R-RANK, which we call process R-RANK'. Process R-RANK' is the same as process R-RANK, but now, immediately after a selection, if the selected node is uncovered we cover it and move to the next selection. Otherwise, we proceed as in process R-RANK.

In our analysis, we find it helpful to consider another process, which we call process H. Process H runs in two phases. For the first phase, consisting of the first $cn \max(1, (\log n)/\delta_{in})$ rounds, we run process SELECT. At the end of phase 1, we remove from the underlying graph all edges which did not have at least one end-point selected during phase 1. After the edge removal, we proceed to phase 2 where we begin to cover vertices as in process R-RANK.

Lemma D.1. *If process H and process R-RANK' use the same node numbering, random rank assignment, and infinite series of selections, the cover time of process R-RANK' is at most the cover time of process H.*

Proof. We prove the stronger claim that if process H and process R-RANK' use the same node numbering, random rank assignment, and infinite series of selections, in round i , every node covered in process H is also covered in process R-RANK'.

Call a round i low if $i \leq cn \max(1, (\log n)/\delta_{in})$, and high otherwise. We call a node marked if it was selected in some low round.

We proceed by induction on i . For the base case, we consider any low round i . In these rounds, process H covers no nodes, so there is nothing to prove.

Now, assume i is high. Let u be the node selected in round i (in both process R-RANK' and process H). If no node is covered in process H, claim follows from the induction hypothesis. Now assume node v covered in process H in round i , and assume that v is not covered in process R-RANK' prior to round i . (If v is covered in process R-RANK' prior to round i , there is nothing to prove.) We now complete the induction step by arguing that v is also covered in round i in process R-RANK'.

If v is marked, then v is already covered in process R-RANK' since it was selected in a low round. So we can assume that v is unmarked. Since H selects u and covers v in round i , (u, v) must not have been removed by process H at the end of phase 1. Thus, u and v cannot both be unmarked, so u is marked.

It follows that u is not equal to v and u is already covered in process R-RANK' as it was selected in a low round. Since u is marked, it has the same set of outgoing neighbors in both processes, i.e., no edge (u, w) was thrown away in process H at the end of the first phase.

Let S (resp., T) be the uncovered outgoing neighbors of u in process R-RANK' (resp., process H) at the beginning of round i . By the induction hypothesis, S is contained in T . Since both processes use the same node numbering and random ranks, the neighbor selection procedure gives well defined order of the nodes. Since $S \subseteq T$ and v is the minimum order node in T and belongs to S , v is the minimum order node in S . Thus v also is covered in round i in process R-RANK'. \square

Lemma D.2. *The cover time of process R-RANK' is $O(n \max(\Delta_{out} \Delta_{in} / \delta_{in}^2, (\log n) / \delta_{in}))$ with high probability. The expected cover time has the same bound.*

Proof. We run a copy of process R-RANK' in parallel with a copy of process H, using the same node numbering, random ranks, and selections.

We call phase 1 of process H successful if at least $\delta_{in}/4$ of every node's in-neighbors are selected. If phase 1 is unsuccessful, we over estimate the cover time of process R-RANK' by the $O(n \log n)$ cover time of coupon collector. If phase 1 is successful, by Lemma D.1 we may overestimate the cover time of process R-RANK' with the cover time bound of process H. To find the cover time bound of process H, we add the number of rounds during phase 1, to the cover time bound of process R-RANK during phase 2. We apply Lemma 4.13 to phase 2 of process H where the graph has in-degree at least $\delta_{in}/4$, to get a cover time bound

of $O(\max(\Delta_{out}\Delta_{in}/\delta_{in}^2, (\log n)/\delta_{in}))$ for process H. Since the bound on the cover time of process H is both with high probability and in expectation, if phase 1 is successful with high probability, the same bound holds for process R-RANK'.

All that remains to be shown to prove the required result is that phase 1 is successful with high probability.

Consider a specific node w . The probability of selecting a node in $\Gamma_{in}(w)$ on any selection is a Bernoulli random variable with success probability at least δ_{in}/n . The number of selections in $\Gamma_{in}(w)$ during phase 1 is the sum of $cn \max(1, (\log n)\delta_{in})$ such independent Bernoulli random variables. Thus, by Lemma 2.4, the probability of getting less than $(c/2) \max(\delta_{in}, \log n)$ selections in $\Gamma_{in}(w)$ during phase 1 is at most $\exp((c/12) \max(\delta_{in}, \log n))$, which is an arbitrary inverse polynomial by choosing a large enough constant c .

Given that $(c/2) \max(\delta_{in}, (\log n))$ selections during phase 1 select a vertex in $\Gamma_{in}(w)$, we apply Lemma 2.6. To do so, let the variables in the lemma be $n = |\Gamma_{in}(w)| \geq \delta_{in}$, and $j = (c/2) \max(\delta_{in}, (\log n))$ which is also at least δ_{in} if we set $c \geq 2$. Thus, Lemma 2.6 tell us that the probability less than $\frac{\delta_{in}}{4}$ distinct nodes of $\Gamma_{in}(w)$ are selected during phase 1 of process H is at most $\exp(\frac{c}{2} \max(\delta_{in}, \log n))$, which is an arbitrary inverse polynomial by selecting a large enough constant c . Taking the union bound over all nodes in the graph shows that phase 1 is successful with high probability. \square