Title: Balanced Allocation Through Random Walk

Abstract: We consider the allocation problem in which $m \leq (1-\varepsilon)dn$ items are to be allocated to $n$ bins with capacity $d$. The items $x_1, x_2, \ldots, x_m$ arrive sequentially and when item $x_i$ arrives it is given two possible bin locations $p_i = h_1(x_i), q_i = h_2(x_i)$ via hash functions $h_1, h_2$. We consider a random walk procedure for inserting items and show that the expected time insertion time is constant provided $\varepsilon = \Omega\left(\sqrt{\frac{\log d}{d}}\right)$. 
We show that a random walk algorithm has constant expected time to insert an item into a table where (i) each item has a random choice of two bins and (ii) each bin can hold up to d items.
Balanced Allocation Through Random Walk

Alan Frieze∗ Samantha Petti†

August 22, 2017

Abstract

We consider the allocation problem in which $m \leq (1 - \varepsilon)dn$ items are to be allocated to $n$ bins with capacity $d$. The items $x_1, x_2, \ldots, x_m$ arrive sequentially and when item $x_i$ arrives it is given two possible bin locations $p_i = h_1(x_i), q_i = h_2(x_i)$ via hash functions $h_1, h_2$. We consider a random walk procedure for inserting items and show that the expected time insertion time is constant provided $\varepsilon = \Omega\left(\sqrt{\frac{\log d}{d}}\right)$.

1 Introduction

We consider the following allocation problems. We have $m$ items that are to be allocated to $n$ bins, where each bin has space for $d$ items. The items $x_1, x_2, \ldots, x_m$ arrive sequentially and when item $x_i$ arrives it is given two possible bin locations $p_i = h_1(x_i), q_i = h_2(x_i)$ via hash functions $h_1, h_2$. We shall for the purpose of this paper assume that $p_i \neq q_i$ for $i \in [m]$ and that $(p_i, q_i)$ is otherwise chosen uniformly at random from $[n]^2$. This model is explicitly discussed in Dietzfelbinger and Weidling [1].

Suppose now that $m \leq d(1 - \varepsilon)n$ where $m, n$ grow arbitrarily large and that $\varepsilon > 0$ is small and independent of $n$. They prove the following:

1. If $d \geq 1 + \frac{\log(1/\varepsilon)}{1 - \log 2}$ then w.h.p.1 all the items can placed into bins.

2. If $d > 90\log(1/\varepsilon)$ then the expected time for a Breadth First Search (BFS) procedure to insert an item is at most $(1/\varepsilon)^{O(\log d)}$.

This model is related to a $d$-ary version of Cuckoo Hashing (Pagh and Rodler [6]) that was discussed in Fotakis, Pagh, Sanders and Spirakis [2]. Here there are $d$ hash functions and the bins are of size one. This latter paper also uses BFS to insert items.

Item insertion in both of these models can also be tackled via random walk. For $d$-ary Cuckoo Hashing, Frieze, Mitzenmacher and Melsted [5] and Fountoulakis, Panagiotou and Steger [3] gave $O((\log n)^{O(1)})$ time bounds for random walk insertion and more recently Frieze and Johansson [4] proved an $O(1)$ time bound on random walk insertion, for $d$ sufficiently large.

∗Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh PA15213. Research supported in part by NSF grant DMS0753472
†School of Mathematics, Georgia Tech., Atlanta, GA30313. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1650044.

1 A sequence of events $(\mathcal{E}_n, n \geq 0)$ is said to occur with high probability (w.h.p.) if $\lim_{n \to \infty} \Pr[\mathcal{E}_n] = 1.$
The authors of [1] ask for an analysis of a random walk procedure for inserting an item. They ask for bounds of $O(\log 1/\varepsilon)$ insertion time while maintaining $d = O(\log 1/\varepsilon)$. While we cannot satisfy these demanding criteria, in this note we are able to establish constant expected time bounds with a larger value of $d$. We first describe the insertion algorithm. We say a bin is saturated if it contains $d$ items.

Random Walk Insertion: RWI

**For** $i = 1$ to $m$ **do**

**begin**

Generate $p_i, q_i$ randomly from $[n]$ if either of bins $p_i, q_i$ are not saturated, then assign item $i$ arbitrarily to one of them.

if both bins $p_i, q_i$ are saturated then **do**

**begin**

Choose $b$ randomly from $\{p_i, q_i\}$.

**repeat**

Let $x$ be a randomly chosen item from bin $b$.

Remove $x$ from bin $b$ and replace it with item $i$.

Let $c$ be the other bin choice of item $x$.

$b \leftarrow c$.

**until** bin $b$ is unsaturated.

Place item $x$ in bin $b$.

**end**

**end**

Let $r_i$ denote the number of steps in loop A of algorithm RWI. Then,

**Theorem 1.** Let $m \leq (1 - \varepsilon)dn$. Then for some absolute constant $M > 0$,

$$E[r_i] \leq \frac{4M}{\varepsilon^2} \text{ w.h.p. for } i \in [m] \text{ provided } \varepsilon \geq \sqrt{\frac{M (\log(4d) + 1)}{d}}. \quad (1)$$

In the analysis below, we take $M = 96$. It goes without saying that we have not tried to optimize $M$ here.

There are two sources of randomness here. The random choice of the hash functions and the random choices by the algorithm. The w.h.p. concerns the assignment of items to bins by the hash functions and the $E[r_i]$ is then the conditional expectation given these choices.

2 Graphical Description

We use a digraph $D'$ to represent the assignment of items to bins. Each bin is represented as a vertex in $D'$ and item $i$ is represented as a directed edge $(p_i, q_i)$ or $(q_i, p_i)$ that is oriented toward its assigned bin. We say a vertex is saturated if its in-degree in $D'$ is $d$. As the algorithm is executed, we in fact build two digraphs $D$ and $D'$ simultaneously.

We now describe the insertion of an item in terms of $D, D'$. Let $x$ and $y$ denote the two randomly selected bins for an item. We place a randomly oriented edge between $x$ and $y$ in $D$. If $x$ and $y$ are unsaturated in $D'$, then we place the edge in the same orientation as in $D$. If $x$ or $y$ is saturated
Lemma 3. Let \( S = A \cup T \) as in Definition 1 and let \( G_S \) be the graph on \( S \) induced by \( G \). Each spanning tree of a component in \( G_S \) has the following form:
1. A set $K$ of $k$ vertices.

2. A set $L = K \cap A$ of size $\ell$. We suppose that $L$ induces a forest with $s$ components and $\ell - s$ edges.

3. A set $K \setminus L = K \cap T = \{v_1, v_2, \ldots, v_{k-\ell}\}$ where $v_i$ is saturated by the algorithm after $v_i$, $v_i$ has $s_i \geq 2$ neighbors in $L \cup \{v_1, v_2, \ldots, v_{i-1}\}$, and $s_1 + s_2 + \cdots + s_{k-\ell} = k - 1 - (\ell - s)$.

4. $\ell \geq \frac{k+2}{2}$.

Proof. As described in the proof of Lemma 2 each vertex in $T$ is adjacent to at least two vertices in $S$. Therefore (1), (2), (3) hold. For (4), note

$$2(k - \ell) \leq s_1 + s_2 + \cdots + s_{k-\ell} = k - 1 - (\ell - s)$$

which implies

$$k - \ell \leq s - 1 \leq \ell - 2,$$

and (4) follows directly. \hfill \square

Lemma 4. Let $S$ be as in Definition 1 and let $M = 96$. When

$$\sqrt{\frac{M (\log(4d) + 1)}{d}} \leq \varepsilon,$$

the expected number of components of $G_S$ of size $k$ is bounded above by

$$\frac{n}{k^2} \exp \left( - \frac{\varepsilon^2 dk}{M} \right).$$

Proof. We use an upper bound on the expected number of spanning trees of size $k$ in $G_S$ as an upper bound on the number of components of $G_S$ of size $k$.

Let $T$ be a tree on a specified labeled set of $k$ vertices, and let $L$ be a specified subset of vertices of $V(T)$ of size $\ell$. Let $A_{T,L}$ be the event that $T$ is present in $G$ and all vertices in $L$ are in $S$. Let $I(L)$ be the sum of the in-degrees of the vertices in $L$, and let $I^*(L)$ be $I(L)$ minus the number of edges in $T$ oriented into a vertex in $L$. We compute

$$\Pr[A_{T,L}] \leq \Pr[T \text{ appears in } G] \Pr[I(L) \geq \ell(d - 1) \mid T \text{ appears in } G] \leq \Pr[T \text{ appears in } G] \Pr[I^*(L) \geq \ell(d - 1) - (k - 1) \mid T \text{ appears in } G] \leq \left( \frac{2d}{n} \right)^{k-1} \exp \left( - \frac{\varepsilon^2 dk}{96} \right)$$

Explanation of (2). If the tree $T$ is present in $G$, then $I(L) - I^*(L)$ is at most $k - 1$. Therefore $I^*(L)$ must be at least $\ell(d - 1) - (k - 1)$.

Explanation of (3). The term $\left( \frac{2d}{n} \right)^{k-1}$ can be explained as follows: Let $m = (1 - \varepsilon)dn$ be the number of pairs offered to algorithm RWI. Each pair has probability $1/n$ of being a particular edge. The probability that $G$ contains $k - 1$ given edges is then at most

$$m^{k-1} \left( \frac{2}{n(n-1)} \right)^{k-1} = \left( \frac{2(1 - \varepsilon)d}{n} \right)^{k-1} \leq \left( \frac{2d}{n} \right)^{k-1}.$$
The term \( \exp\left(-\frac{\varepsilon^2 dk}{96}\right) \) can be explained as follows: \( I^*(L) \) is dominated by the sum of \( \ell(n - \ell) \) independent Bernoulli random variables, \( \xi_{u,v} \), each of which corresponds to an ordered pair \( (u,v) \) where \( u \not\in L, v \in L \) and \( \{u,v\} \not\in T \). There are at most \( \ell(n - 1) - (k - 1) \) such pairs. Here \( \Pr[\xi_{u,v} = 1] \leq \frac{m}{2(n)} = \frac{(1-\varepsilon)d}{n-1} \) bounds the probability that the edge \( \{u,v\} \) exists in \( G \) and oriented from \( u \) to \( v \) in \( D \). The existence of edges are negatively correlated since the number of edges is bounded above. Therefore, \( I^*(L) \) is dominated by \( X_{\varepsilon} \sim \text{Bin}\left(\ell(n - 1), \frac{(1-\varepsilon)d}{n-1}\right) \).

Thus if

\[
p^* = \Pr[I^*(L) \geq \ell(d - 1) - (k - 1) | T \text{ appears in } G]
\]

then

\[
p^* \leq \Pr[X_{\varepsilon} \geq \ell(d - 1) - (k - 1)] \\
\leq \Pr[X_{\varepsilon} \geq \ell(d - 3)] \\
\leq \Pr[X_{\varepsilon} \geq (1 + \theta) E[X_{\varepsilon}]],
\]

where \( \theta = \frac{\varepsilon d - 3}{(1-\varepsilon)d} \). Note that \( \theta \leq 1 \) if \( \varepsilon \leq \varepsilon_0 = \frac{d+3}{2d} \).

The Chernoff bounds then imply that if \( \varepsilon \leq 1/2 \leq \varepsilon_0 \) then

\[
p^* \leq \exp\left(-\frac{1}{3} \cdot \left(\frac{\varepsilon d - 3}{(1-\varepsilon)d}\right)^2 \ell d (1 - \varepsilon) \right) \leq e^{-\varepsilon^2 \ell d/12} \leq e^{-\varepsilon^2 kd/24}
\]

since \( \varepsilon d \geq 6 \). When \( \varepsilon > 1/2 \) we see that \( X_{\varepsilon} \) is dominated by \( X_{1/2} \) and then replacing \( \varepsilon^2 \) by \( (\varepsilon/2)^2 \), we have (3).

We now compute the expected number of trees of size \( k \). Let \( Z_k \) denote the number of spanning trees in \( G_S \) with \( k \) vertices. Then

\[
E[Z_k] \leq \left(\frac{n}{k}\right)^k k^{k-2} \sum_{\ell \geq k/2} \binom{k}{\ell} \Pr[A_{T,L}]
\leq \frac{(ne)^k}{k^2} 2^{k-1} \left(\frac{2d}{n}\right)^{k-1} \exp\left(-\frac{\varepsilon^2 dk}{96}\right)
\leq \frac{n}{k^2} \exp\left(k \left(\log(4d) + 1 - \frac{\varepsilon^2 d}{96}\right)\right)
\leq \frac{n}{k^2} \exp\left(-\frac{\varepsilon^2 dk}{M}\right).
\]

Finally, we give a high probability bound on the size of the maximum component in \( G_S \) and show that with high probability each component of \( G_S \) contains at most one cycle.

**Lemma 5.** Let \( \varepsilon \geq \sqrt{\frac{M(\log(4d) + 1)}{d}} \) and \( S \) be as in Definition 1. Then, w.h.p., the maximum size of a component of \( G_S \) is at most

\[
k_0 = \alpha \log n \text{ where } \alpha = \frac{M}{\varepsilon^2 d},
\]

and each component contains at most one cycle.
Proof. Let $Z_k$ be the expected number of components of $G_S$ of size $k$. We use Lemma 4 to compute
\[
\Pr[\exists \text{ a component of size } > k_0] \leq \sum_{k_0 < k < n} E[Z_k] \leq n \sum_{k_0 < k < n} k^{-2} \exp\left(-\frac{\varepsilon^2 dk}{M}\right) = o(1).
\]
It follows that w.h.p. the maximum size of a component of $G_S$ is at most $k_0$.

Next we show that w.h.p. each component of $G_S$ contains at most one cycle. Let $Y_k$ be the number of sets $Q$ of size $k$ with $e(Q) \geq |Q| + 1$. We bound the number structures on $k$ vertices consisting of a tree plus two edges by $k^{k+2}$ and compute
\[
\Pr[\exists Q: |Q| \leq k_0, e(Q) \geq |Q| + 1] \leq \sum_{k \leq k_0} E[Y_k]
\]
\[
\leq \sum_{k \leq k_0} \binom{n}{k} k^{k+2} \left(\frac{2d}{n}\right)^{k+1}
\]
\[
\leq \frac{2d}{n} \sum_{k \leq k_0} k^2 (2de)^k
\]
\[
= o(1).
\]

4 Expected insertion time

Recall from Section 2 the definition of a replacement walk. To bound the the expected length of the replacement walk, we first show that a random walk on $G'$ that starts in a size $k$ component of $G_S$ will remain in the component for at most $2k$ steps in expectation.

Lemma 6. Let $G$ satisfy the conditions of Lemma 5, and let $S$ be as in Definition 1. Consider a random walk on $G'$ beginning in a size $k$ component of $G_S$. Then expected number of steps this random walk takes before leaving $G_S$ is bounded above by $2k$.

Proof. Consider a random walk on $G'$ beginning in a size $k$ component $C$ of $G_S$. Note the edges of $G'$ are oriented, so the walk does not back track. Therefore, if $C$ is a tree, a random walk can take at most $k$ steps in $C$ before leaving $C$.

Next suppose $C$ has a cycle of length $\ell$. Since $C$ has only one cycle, in any walk in $C$ all edges of the cycle must be visited consecutively (i.e. it is not possible to take a walk on the cycle, leave the cycle, and then return to the cycle). Let $v$ be the first vertex visited by the walk that is on the cycle. The expected number of times the walk completes the entire cycle at this point is
\[
\sum_{i=1}^{\infty} \left(\frac{1}{d^\ell}\right)^i \left(1 - \frac{1}{d^\ell}\right)^i = \frac{1}{d^\ell - 1}.
\]
Since the walk cannot return to the cycle after the walk leaves the cycle, the expected length of the walk before it leaves $C$ is at most
\[
k + \frac{\ell}{d^\ell - 1} \leq 2k.
\]
Finally, we prove Theorem 1.

Proof. (of Theorem 1). The expected insertion time is the probability two randomly selected vertices $x$ and $y$ of $G'$ are saturated times the expected length of a replacement walk in the graph $G'$ starting a random saturated vertex. Since the replacement walk ends once an unsaturated vertex is reached, Lemma 6 implies that the expected length of a replacement walk starting at a vertex in a component of size $k$ is bounded above by $2k$. We compute

$$
E[\text{insertion time}] \leq 2 \sum_{i=1}^{k_0} \Pr[x \in S] \Pr[y \text{ in component of size } k \text{ in } G_S]k
$$

$$
\leq 2 \sum_{i=1}^{k_0} |S| \frac{n}{k} \exp\left(\frac{-\varepsilon^2 dk}{M}\right)k
$$

$$
\leq 2 \sum_{i=1}^{\infty} \exp\left(-\frac{\varepsilon^2 dk}{M}\right)
$$

$$
= \frac{2}{1 - e^{-\varepsilon^2/M}}
$$

$$
\leq \frac{4M}{\varepsilon^2}
$$

\[\square\]

5 Conclusion

We have proved an $O(1)$ bound on the expected insertion time of a natural random walk insertion algorithm. The next step in the analysis of this algorithm is to reduce the dependence of $d$ on $\varepsilon$.

References


