Information Processing Letters 131 (2018) 39–43

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Balanced allocation through random walk

Alan Frieze [a,*,1], Samantha Petti [b,2]

[a] Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213, United States
[b] School of Mathematics, Georgia Tech., Atlanta, GA 30313, United States

ARTICLE INFO

Article history:
Received 22 August 2017
Received in revised form 26 November 2017
Accepted 26 November 2017
Available online 28 November 2017
Communicated by B. Doerr

Keywords:
Balanced allocation
Random walk
Cuckoo hashing
On-line algorithms

ABSTRACT

We consider the allocation problem in which $m \le (1-\varepsilon)dn$ items are to be allocated to n bins with capacity d. The items $x_1, x_2, \dots, x_m$ 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 problem. We have m items that are to be allocated to n bins, where each bin has space for d items. The items $x_1, x_2, \dots, x_m$ 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 $p_i \ne q_i$ 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 [2]. Probabilistic bounds on the number of items m so that all m items can be inserted have been found by Cain, Sanders and Wormald [1] and independently by Fernholtz and Ramachandran [3].

Algorithmically, if $m \le d(1-\varepsilon)n$ where m, n grow arbitrarily large and $\varepsilon > 0$ is small and independent of n, then [2] prove the following:

1. If $d \ge 1 + \frac{\log(1/\varepsilon)}{1-\log 2}$ then w.h.p.[3] all the items can be 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 [9]) that was discussed in Fotakis, Pagh, Sanders and Spirakis [4]. 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 [8] and Fountoulakis, Panagiotou and Steger [5] gave $O((\log n)^{O(1)})$ time bounds for random walk insertion and more recently Frieze and Johansson [6] proved an $O(1)$ time bound on random walk insertion, for d sufficiently large.

The authors of [2] 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

* Corresponding author.
E-mail addresses: alan@random.math.cmu.edu (A. Frieze), spett3@gatech.edu (S. Petti).

1 Research supported in part by NSF grant DMS0753472.
2 This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1650044.

3 A sequence of events $(\mathcal{E}_n, n \ge 0)$ is said to occur with high probability (w.h.p.) if $\lim_{n\to\infty} \Pr[\mathcal{E}_n] = 1$.

https://doi.org/10.1016/j.ipl.2017.11.010
0020-0190/

$d = O(\log 1/\varepsilon)$. 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 x_i arbitrarily to one of them.
  if both bins p_i, q_i are saturated then **do**
    begin
    Choose b randomly from $\{p_i, q_i\}$; $y \to x_i$.
A   **repeat**
      Let x be a randomly chosen item from bin b.
      Remove x from bin b and replace it with item y.
      Let c be the other bin choice of item x.
      $y \leftarrow x$
      $b \leftarrow c$.
    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 $m \le (1-\varepsilon)dn$. Then for some absolute constant $M > 0$,*

$$\mathsf{E}[r_i] \le \frac{4M}{\varepsilon^2} \text{ w.h.p. for } i \in [m] \text{ provided}$$

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

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 $\mathsf{E}[r_i]$ 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 in D', then we place the edge in D' according to the algorithm RWI, which may require flipping edges in D'. This is repeated for all items. Note that D is a random directed graph with $(1-\varepsilon)dn$ edges. The undirected degree of each vertex in D is the same as in D'. However, the directed degrees will vary. Let D_t and D'_t denote the respective graphs after t edges have been inserted.

We compute the expected insertion time after $(1-\varepsilon)dn$ items have been added by analyzing D'. The expected time to add the next item is equal to the expected length of the following random walk in D'. Select two vertices x and y. If either is unsaturated no walk is taken, so we say the walk has length zero. Otherwise, pick a vertex at random from $\{x, y\}$ and walk "backwards" along edges oriented into the current vertex until an unsaturated vertex is reached. We call this the "replacement walk." As usual in a random walk, vertices may be revisited during a replacement walk. On the other hand, edges are crossed in a direction opposite to their orientation, which is unusual. Note also that after a vertex is visited for the second time, two of its edges will have the opposite direction to what they had originally. This small observation is crucial for the analysis below.

Let G denote the common underlying graph of D, D' obtained by ignoring orientation. In order to compute the expected length of the replacement walk, we analyze the structure of the subgraph G_S of G induced by a set S which contains all saturated vertices in G. In Section 3, we show that the expected number of connected components of size k among saturated vertices decays geometrically with k and that each component is a tree or contains precisely one cycle. In Section 4 we show that since the components of G_S are sparse and the number of components decays geometrically with size, the expected length of a replacement walk is constant.

3. Saturated vertices

In this section we describe the structure induced by G on the set of saturated vertices. Throughout this section, our observations rely only on information about the digraph D, and therefore are independent of the RWI algorithm. First we define a set S that is a superset of all saturated vertices.

Definition 1. Let A be the set of vertices of D with in-degree at least $d-1$ in D and $T_0 = \emptyset$. Given $A, T_0, \ldots T_k$, let T_{k+1} be all the vertices of $V \setminus (A \cup T_0 \cup T_1 \cup \cdots \cup T_k)$ with at least two neighbors in $A \cup T_0 \cup T_1 \cup \cdots \cup T_k$. Let $T = \bigcup T_i$ and $S = A \cup T$.

Alternatively, let S be the smallest set of vertices that contains A and is closed under adding nodes that have two neighbors in S.

Lemma 2. *The set S defined above contains all saturated vertices.*

Proof. We prove the statement by induction on S_t the set of saturated vertices after the t-th edge is added. Since $S_0 = \emptyset$, $S_0 \subseteq S$ vacuously. Assume $S_t \subseteq S$. Note that the