Balls into Non-uniform Bins

Petra Berenbrink[a], André Brinkmann[b], Tom Friedetzky[c], Lars Nagel[b]

[a]School of Computing Science, Simon Fraser University, Burnaby, B.C., V5A 1S6, Canada

[b]Zentrum für Datenverarbeitung, Johannes-Gutenberg-Universität Mainz, 55099 Mainz, Germany

[c]School of Engineering and Computing Sciences, Durham University, Durham DH1 3LE, United Kingdom

Abstract

Balls-into-bins games for uniform bins are widely used to model randomised load balancing strategies. Recently, balls-into-bins games have been analysed under the assumption that the selection probabilities for bins are not uniformly distributed. These new models are motivated by properties of many peer-to-peer (P2P) networks. In this paper we consider scenarios in which non-uniform selection probabilities help to balance the load among the bins. While previous evaluations try to find strategies for identical bins, we investigate heterogeneous bins where the "capacities" of the bins might differ significantly. We look at the allocation of m balls into n bins of total capacity C where each ball has d random bin choices. For such heterogeneous environments we show that the maximum load remains bounded by $\ln \ln(n)/\ln(d) + \mathcal{O}(1)$ *w.h.p.* if the number of balls m equals the total capacity C. Further analytical and simulative results show better bounds and values for the maximum loads in special cases.

Keywords: load balancing, randomized algorithms, balls-into-bins games

1. Introduction

In the standard balls-into-bins game, m unit-sized balls are allocated to n identical bins. It is assumed that every ball independently and uniformly at random chooses d bins and that it commits itself to a least-loaded of these bins. The goal of this strategy is to balance the load among the bins, by minimising the maximum number of balls allocated to any bin.

Balls-into-bins games are successfully used to model randomised load balancing strategies in networks and many other "real world" applications (see, e.g., [1,

Email addresses: `petra@cs.sfu.ca` (Petra Berenbrink), `brinkman@uni-mainz.de` (André Brinkmann), `tom.friedetzky@dur.ac.uk` (Tom Friedetzky), `nagell@uni-mainz.de` (Lars Nagel)

2, 3, 4, 5] for applications of randomisation strategies). In these cases, balls represent requests or data items, while bins model servers or some form of storage. Most of the previous papers assume the same uniform capacity (or size) for all bins and uniform bin probabilities. The goal is to balance the load in a way that each bin receives approximately the same number of balls. We should point out that when we refer to a bin's "capacity" / "size" then we do not mean to imply the existence of a maximum "volume", or load threshold (as in e.g. bin packing); the reader should think more in terms of "speed", "bandwidth" or "compression ratio". The precise notion that we use throughout this paper is simply that when a ball of size s is placed into a bin of capacity c, then the "effective" load that this bin experiences is $\ell = s/c$.

Standard balls-into-bins games assume that the probability of a bin to be selected by a ball is the same for all bins. It is unfortunately very difficult to maintain this property in distributed environments without centralised control. P2P environments like Chord or CAN [2, 1], e.g., are unable to map peers evenly to their address space, making some bins more likely to be selected than others [6, 7, 8]. Byers *et al.* [7, 9] extended the model, assuming that the probability for a bin to be selected within a random experiment is not uniform over all bins. Their underlying process still tries to balance the number of balls as evenly as possible over the set of bins.

However, in many practical applications, some bins can handle a much larger load than others, under-utilizing stronger bins under these constraints. In the variant of balls-into-bins games that we consider, it is assumed that the bins are not uniform, but that they come with an integer capacity, as outlined above.

Let the total capacity C be the sum of the capacities of all bins. The natural probability for a bin to be chosen would be either $1/n$, that is uniform, or c_i/C, proportional to the bin's capacity c_i. We will analyse the latter case for $d \geq 2$ and show that the maximum load is $\ln\ln(n)/\ln(d) + \mathcal{O}(1)$ *w.h.p.* (Theorem 3). Furthermore, we will investigate cases in which the maximum load is constant (Theorems 1 and 2). In some cases changing the probability distribution leads to much better results (Theorem 5).

1.1. Related Work

There is a vast number of papers dealing with balls-into-bins games in their many different settings. We restrict our attention to major results and previous work that is relevant to the results presented in this paper.

In the standard game in which each ball chooses d bins *i.u.r.*, the maximum load can be bounded by $\ln\ln(n)/\ln(d) + \Theta(1)$ if $m = n$ balls are thrown [10]. In case $m \gg n$ the deviation of the load from the average m/n is also $\ln\ln(n)/\ln(d) + \Theta(1)$ and thus independent of the number of balls m [11].

Recently, several papers have examined the case in which bins are not chosen *i.u.r.* The motivation for these models comes from the properties of peer-to-peer networks like Chord, which use Consistent Hashing to distribute requests (balls) over computers (bins) [6, 2]. There the computers and requests are mapped to random points on a ring and the requests are assigned to the closest computer on the ring in an anti-clockwise direction. Therefore, each bin is responsible for one arc on the ring. The maximum arc length can be up to a factor of $\log(n)$ larger than the average arc length.

Byers *et al.* [7, 9] successfully apply the power-of-two-choices paradigm to this setting by letting each request randomly choose $d \geq 2$ points and allocate itself to a peer of lowest load. Although the maximum arc length can be up to $\log(n)$ times larger than the average one, the maximum load of every peer is still bounded by $\ln\ln(n)/\ln(d) + \Theta(1)$, *w.h.p.*, for $m = n$. Hence, the work of Byers *et al.* shows that this imbalance does not lead to a shift in the maximum load for the case $m = n$.

Wieder [12] demonstrates that in the scenario of Byers *et al.* the maximum difference between the loads grows with m. Thus, for $m \gg n$ the bounds are not as tight as in the standard case [11]. However, if the number of choices d is allowed to (slowly) grow with the deviation in the probability distribution, the maximum load is again bounded by $\frac{m}{n} + \mathcal{O}(\ln\ln(n))$ (which complies with [11]). The presented bounds are tight in a way that a smaller d leads to a deviation of the load linear in m.

Kenthapadi and Panigrahy [13] and Godfrey [14] analyse graph-based models in which the random choices are non-uniform and dependent. In [13] the 2-choice game is considered with the restriction that balls can only choose bins that are connected by an edge in an underlying graph G. The authors assume that each edge in the graph has the same probability to be chosen and show that the maximum load does not deviate much from the maximum load in the standard 2-choice game provided that G is (almost) n^{ϵ}-regular where ϵ is a large enough constant. In [14] Godfrey generalises the model to the d-choice game by assuming that the underlying graph is a d-uniform multi-hypergraph (which is even allowed to change with each ball). A ball can only choose sets of d bins that correspond to hyperedges in its hypergraph. Again, each hyperedge has the same probability to be chosen. Godfrey investigates under which circumstances the maximum load in any bin is 1 *w.h.p.* The authors of [15] consider the same model and improve the results of [14].

The case of heterogeneous bin sizes has been considered in the related field of selfish load balancing (e.g., [16, 17]), but to our knowledge nobody has analysed it for multiple-choice games. Such games are mentioned by Wieder [12] to motivate his work about multiple-choice games with heterogeneous probabilities. He suggests to choose the bins' probabilities proportional to their capacities. In this paper we will analyse this particular case and variations of it.

1.2. Our Contributions

All previous results assume that each bin has a uniform capacity and that the balls should be distributed as evenly as possible. In contrast, we assume that the system consists of heterogeneous bins where each bin i can have an arbitrary, integral capacity c_i and the objective is to balance the load of each bin, which is defined as the number of balls inside this bin divided by its capacity. If not stated otherwise, we assume that a bin's probability to be chosen is proportional to its size.

In the analytical part of this paper, we assume that we have n bins with a total capacity of C and $m = C$ balls. Hence, the optimal maximum load is one. The main analytical result from Section 3 shows that the maximum load of a bin is $\ln\ln(n)/\ln(d) + \mathcal{O}(1)$ (Theorem 3) if $d \geq 2$. Hence, the maximum load does not grow with an increasing capacity. We even show that the maximum load becomes constant if almost all bins are big, *i.e.*, have size $\Omega(\ln(n))$ (Theorem 1). Provided that we can choose a different probability distribution, a constant maximum load can be achieved even if there is only a constant fraction of $\Theta(\ln\ln(n))$-sized bins (Theorem 5). The proof of this theorem uses Observation 2, which states that if all bins have the same capacity $\bar{c}$, the maximum load is bounded by $(m/n + \mathcal{O}(\ln\ln(n)))/\bar{c}$ *w.h.p.*

Based on a simulation environment, we arrange and simulate bin arrays with varying parameters in Section 4 and compare our analytical results with the experiments. There we also consider settings that we do not analyse, most notably the heavily loaded case and systems with a small number of bins.

2. Model and Definitions

We assume bins to be non-uniform. Each bin comes with a positive integer capacity which we also refer to as size. We denote the capacities / sizes of the n bins by $c_1, \ldots, c_n$, and let $C = \sum_{i=1}^{n} c_i$.

We usually allocate $m = C$ balls into our system of n bins and assume that each ball has $d \geq 2$ choices. In the following we say that a bin is *chosen* or that a ball *chooses* a bin if we refer to the d choices of a ball. When a bin actually receives the ball, then we say that the bin *gets* or *is allocated* the ball.

The load balancing protocol (see Algorithm 1) greedily tries to minimise the maximum load within the set B of the d chosen bins after a ball has been

Algorithm 1 Load Balancing Protocol

```
1: for all balls do
2:     Independently choose a set B of d bins at random
3:     Determine the set B_opt ⊆ B of bins that would have the lowest load after
       allocating the ball
4:     Determine the maximum capacity c_max of the bins in B_opt
5:     Remove all bins b_j with capacity c_j < c_max from B_opt
6:     i.u.r. choose a bin from B_opt and allocate the ball to it
7: end for
```

allocated. It therefore determines the subset $B_{opt} \subseteq B$ with the smallest load after a possible allocation and *i.u.r.* allocates the ball to one of the bins with the highest capacity from it. We will show within the analysis that it is beneficial to move the load into the direction of these bigger bins.

We say that if m_i balls are allocated to a bin b_i of capacity $c_i \geq 1$, then this bin's load is $\ell_i = \frac{m_i}{c_i}$. Usually we will assume that the probability of bin b_i with capacity c_i being chosen is $\frac{c_i}{C}$ and therefore proportional to c_i. If we use other probability distributions, we will clearly point this out.

The *height* of a ball is the load of the bin it is allocated to directly after its allocation. If a ball is allocated to a bin with prior load ℓ_i and capacity c_i then its height will be $(\ell_i + 1)/c_i$. The *load vector* of an allocation of balls into n bins is a vector $L = (\ell_1, \ldots, \ell_n)$, where ℓ_i is the load of bin i as defined above. The corresponding *normalised load vector* $\bar{L} = (\bar{\ell}_1, \ldots, \bar{\ell}_n)$ consists of the elements of L in non-increasing order (one may think of this as sorting the bins according to their loads whilst reassigning indices accordingly).

To make our proofs more accessible we will occasionally imagine that each bin of capacity c does actually consist of c many unit-sized *slots* (the protocol is entirely unaware of this). Hence, the total number of slots equals the total capacity C of the bins. For a fixed slot $i \in \{1, \ldots, C\}$ let $b(i)$ denote the unique bin to which slot i belongs. We sometimes pretend that the balls randomly choose a slot instead of a bin, which is again simply an analytical tool that the algorithm is entirely unaware of. Note that the probability to choose a fixed slot will then still be proportional to its bin's capacity.

When thinking in terms of slots rather than bins we have to determine how balls are allocated to the slots of a bin. To do so we define the *slot load vector*

$$S = (\underbrace{s_{1,1}, \ldots, s_{1,c_1}}_{\text{bin } 1}, \underbrace{s_{2,1}, \ldots, s_{2,c_2}}_{\text{bin } 2}, \ldots, \underbrace{s_{n,1}, \ldots, s_{n,c_n}}_{\text{bin } n}),$$

where $s_{i,j}$ is the j-th slot of the i-th bin. We may drop the two-dimensional indices and instead use $\{1, \ldots, C\}$ as index set where convenient. Let i be a bin with slots $s_{i,1}, \ldots s_{i,c_i}$. We assume that the slots of bins are filled in a round-robin fashion. If a ball chooses one of the slots from bin i and bin i contains ℓ balls, we assume that i's first (leftmost) $\ell \bmod c_i$ slots contain one ball more than the remaining slots. A *normalised slot load vector* $\bar{S}$ has length C and an entry for every slot. Similar to normalised load vectors we assume that S is sorted in decreasing order. Also, whenever we have slots with the same (slot) load but whose "host bins" have different loads, we place the one belonging to the bin with higher (bin) load before the other one with smaller (bin) load. Note that in a normalised slot load vector a bin's slots are not necessarily in successive positions. As a small example consider two bins a and b with 4 slots each and load 2.5 and 2.75. The normalised slot load vector is $3, 3, 3, 3, 3, 2, 2, 2$, belonging to bins b, b, b, a, a, b, a, a.

If we allocate m balls into n bins, L_i $(\bar{L}_i, S_i, \bar{S}_i)$ is defined as the load vector (normalised load vector, slot load vector, and normalised slot load vector, respectively) after the allocation of the i-th ball.

To compare the load of two load vectors or two slot vectors we use the notion of *majorisation*.

Definition 1 (Majorisation $\succeq$). *Given two vectors $U = (u_1, \ldots, u_\ell)$ and $V = (v_1, \ldots, v_\ell)$ we say that U majorises V if and only if $\sum_{i=1}^{k} \bar{u}_i \geq \sum_{i=1}^{k} \bar{v}_i$ for all $k = 1, \ldots, \ell$ where $\bar{u}_i$ and $\bar{v}_i$ are the i-th entries of the normalised vectors $\bar{U}$ and $\bar{V}$, respectively. We then write $U \succeq V$.*

3. Analysis

The structure of this section is as follows. The main contribution of this section is Theorem 3, upper-bounding the maximum load of any bin in a system of heterogeneous bins. We start by showing Observation 1 that bounds the load of big bins as well as the height of balls that have at least one big bin among their choices. Lemma 1 shows that load distributions achieved by systems with solely unit-sized bins dominate those achieved by systems with heterogeneous bins with the same total capacity. This Lemma will then be used to prove Theorem 1 and 2, showing under which circumstances (that is, number of small bins vs. number of big bins) we may achieve constant maximum load, and Theorem 3. Observation 2 bounds the maximum load for uniform bin arrays in the heavily loaded case. Finally, using this observation Theorem 5 shows better results for the case in which one can choose the bins' probabilities oneself.

Define a bin to be *big* if its capacity is at least $r \cdot \ln(n)$ for some constant r, and *small* otherwise. With $\mathcal{B}_b$ we denote the set of balls that have at least one big bin among their choices and with $\mathcal{B}_s$ the remaining balls that probe only small bins. $\ell_{max}^{(b)}$ ($\ell_{max}^{(s)}$) is the maximum load in any bin if only the loads of balls from set $\mathcal{B}_b$ ($\mathcal{B}_s$) are counted. Furthermore C_b is the capacity of the big bins and C_s the capacity of the small bins. Hence, $C = C_b + C_s$.

First we bound $\ell_{max}^{(b)}$:

Observation 1. *Consider the d-choice game in which $m = C$ balls are thrown into n bins with total capacity C. Fix any constant $k > 0$. If $k \leq \frac{r}{3} - 1$, the load in every big bin will be at most 4 with probability at least $1 - n^{-k}$. This implies that $\ell_{max}^{(b)} \leq 4$ with probability at least $1 - n^{-k}$.*

Proof. First of all, in this proof the load of a slot is the *inherited* load which is the load of the bin the slot belongs to. Hence, all slots belonging to one bin have the same load.

Consider a big bin b_i with $2 \cdot c_i$ balls, *i.e.* load 2. First we show that the probability for this bin to get a ball b is at most c_i/m. As the number of balls equals the number of slots, at least half of the slots have a load less than 2 when b is allocated. If more than $m/2$ slots had at least 2 balls, then the total number of balls would exceed $2 \cdot m/2 = m = C$. Let A_ℓ be the event that b chooses ℓ big bins and $d-\ell$ small bins, and let B be the event that b is allocated to b_i. As a worst case assumption, to upper bound $\mathbf{Pr}[B \mid A_\ell]$ we assume that the ball is not allocated to any of the small bins. Then

$$\mathbf{Pr}[B \mid A_\ell] \leq \ell \cdot \frac{c_i}{m} \cdot \left(\frac{1}{2}\right)^{\ell-1} \leq \frac{c_i}{m}$$

for $1 \leq \ell \leq d$ and $\mathbf{Pr}[B \mid A_\ell] = 0$ for $\ell = 0$. Furthermore

$$\mathbf{Pr}[B] = \sum_{\ell=1}^{d} \mathbf{Pr}[B \mid A_\ell] \cdot \mathbf{Pr}[A_\ell] \leq \frac{c_i}{m} \cdot \sum_{\ell=1}^{d} \mathbf{Pr}[A_\ell] \leq \frac{c_i}{m}.$$

The expected number of balls hitting the big bin b_i is therefore at most c_i after m balls. Let X_i be the number of balls that have been allocated to b_i. Then, using Chernoff bounds (with $\epsilon = 1$), we obtain

$$\begin{aligned}
\mathbf{Pr}[X_i \geq 2 \cdot c_i] &= \mathbf{Pr}[X_i \geq (1+\epsilon) \cdot c_i] \leq e^{-\epsilon^2 \cdot c_i/3} \leq e^{-r \cdot \ln(n)/3} \\
&= n^{-r/3} \leq n^{-k-1}
\end{aligned}$$

Hence, for r chosen suitably, with probability at least $1-n^{-k-1}$, bin b_i is chosen by at most $2 \cdot c_i$ many balls and the load can be upper-bounded by

$$\ell_i \leq 2 + \frac{X_i}{c_i} \leq 2 + \frac{2 \cdot c_i}{c_i} = 4.$$

Since there are at most n big bins, the probability that (at least) one of them exceeds load 4 is bounded by $n \cdot n^{-k-1} = n^{-k}$.

For the second part of the lemma, note that we assumed that all balls that choose at least one big bin will not be allocated to any of the small bins. Since under these circumstances the maximum load of big bins is still *w.h.p.* at most 4, no ball of $\mathcal{B}_b$ will choose a small bin unless its load is smaller than 4. Hence, no ball of $\mathcal{B}_b$ will have a height of more than 4. This implies $\ell_{max}^{(b)} \leq 4$. $\square$

Lemma 1. *Let P be a d-choice process on n non-uniform bins with total capacity C, and let Q be a d-choice process on C unit-sized bins. Then the maximum load in P is stochastically dominated by the maximum load in Q.*

Proof. We show this result by coupling. Since the number of bins is different in both processes, we define the state space as the set of normalised slot load vectors. The slot load vectors in P and Q have equal length because the total capacity C is the same in both processes. We assume that both processes randomly choose d slots. In the case of P the ball could be allocated to any slot of the bin which receives it, but we assume that the ball is allocated to the rightmost slot. Note that this is only a renumbering of the slots of the bins.

As the process starts with empty slot load vectors, S^P is majorised by S^Q in the beginning. S^P will remain stochastically dominated by S^Q if an order-preserving coupling of the two processes exists. This would already imply the statement of the lemma because the maximum load in P would also remain dominated due to $\bar{\ell}_1^P \le \bar{s}_1^P \le \bar{s}_1^Q = \bar{\ell}_1^Q$.

Let S_j^P and S_j^Q denote the slot load vectors after the j-th ball. For the coupling we have to show that for every ball j there exists a bijection between the random bin choices of P and Q such that $S_j^P \preceq S_j^Q$ implies $S_{j+1}^P \preceq S_{j+1}^Q$. Let $h_1 \le h_2 \le \ldots \le h_d$ be the indices of the d randomly chosen slots in the normalized slot vector of process Q. Then Q will allocate the ball into slot h_d. For the moment let us assume that P will use the rightmost slot of the bin that contains slot h_d. Hence, P will use the slot h_d itself or a slot on the right side of h_d in the normalised slot load vector. It follows from Claim 2.4 in [12] that $S_{j+1}^P \preceq S_{j+1}^Q$.

It remains to prove that P allocates the ball to a least loaded bin by choosing the bin that contains h_d. (i) Note that we compare the least loaded slots of the bins in question and that the slots of a bin are filled in a round-robin fashion. Therefore, if one slot has a strictly smaller load than another slot, then the same is true for the according bins. (ii) Recall that we added to the definition of the normalised slot load vector that slots of the same load are ordered by the loads of the respective bins in decreasing order. Hence, even if two slots have the same load, then the slot with the higher index belongs to the bin of lesser (or equal) load. □

Before we state the first theorem, we prove the following lemma, which will be repeatedly used in the theorem's proof.

Lemma 2. *Define $X_s := |\mathcal{B}_s|$ and let Y count the number of times in which a ball from $\mathcal{B}_s$ falls into a non-empty bin. Then*

1. $$\mathbf{Pr}\left[\, X_s \ge k \,\right] \le \left(\frac{e \cdot C_s^2}{k \cdot C}\right)^k$$

2. $$\mathbf{Pr}\left[\, Y \ge \lambda \mid X_s = k \,\right] \le \left(\frac{e \cdot k^3}{\lambda \cdot C_s^2}\right)^\lambda .$$

Proof. First we show Part 1. Recall that $\mathcal{B}_s$ denotes the set of balls that have all d choices among small bins. Note that $C_s \le c \cdot (n \cdot \ln(n))^{2/3} = o(n)$ implies that there are $\Theta(n)$ big bins and that therefore $m \ge n \cdot \ln(n)/2$.

A ball belongs to $\mathcal{B}_s$ if all d choices are directed to small bins. Hence, the probability for a ball to be in $\mathcal{B}_s$ is

$$p_s = \left(\frac{C_s}{C}\right)^d \le \left(\frac{C_s}{C}\right)^2 .$$

For the number X_s of such balls we therefore obtain

$$\mathbf{Pr}\left[\, X_s \geq k \,\right] = \mathbf{Pr}\left[\, B(m, p_s) \geq k \,\right] \leq \left(\frac{e \cdot C}{k}\right)^k \cdot p_s^k \leq \left(\frac{e \cdot C_s^2}{k \cdot C}\right)^k$$

where $B(m, p_s)$ is the binomial distribution in parameters m (number of trials) and p_s (success probability of each trial), and $X_s \sim B(m, p_s)$. The first inequality in Eq. 1 uses the well-known bound $\binom{n}{k} \leq \left(\frac{en}{k}\right)^k$ with $e = \exp(1)$.

Now we show Part 2. We assume $X_s \leq k$ where k is taken over from Part 1. The remaining task is to bound the maximum load of the game in which k balls are allocated into the small bins with a total capacity of C_s. Lemma 1 states that the maximum load of this process is dominated by the maximum load of the process $\mathcal{P}$ that allocates k balls to C_s unit-sized bins. Therefore, it is sufficient to show a constant bound for the maximum load in $\mathcal{P}$.

Recall that Y count the number of collisions, that is, the number of times in which a ball from $\mathcal{B}_s$ falls into a non-empty bin. For each case, we will show that Y is constant *w.h.p.* which already implies a constant maximum load. Let Y_i, $i \in [k]$, denote binary random variables such that $Y_i = 1$ if the i-th ball from $\mathcal{B}_s$ collides with one of the previous balls and $Y_i = 0$ otherwise. Observe that $Y = \sum_{i=1}^{|\mathcal{B}_s|} Y_i$. The collision probability $p_c := \mathbf{Pr}\left[\, Y_i = 1 \,\right]$ for ball i is upper-bounded by

$$p_c = \mathbf{Pr}\left[\, Y_i = 1 \,\right] \leq \left(\frac{i-1}{C_s}\right)^d \leq \left(\frac{i-1}{C_s}\right)^2 < \left(\frac{|\mathcal{B}_s|}{C_s}\right)^2 .$$

For the number of collisions Y we obtain

$$\mathbf{Pr}\left[\, Y \geq \lambda \mid |\mathcal{B}_s| = k \,\right] \leq \mathbf{Pr}\left[\, B(k, p_c) \geq \lambda \,\right] \leq \left(\frac{e \cdot k}{\lambda}\right)^\lambda \cdot p_c^\lambda \leq \left(\frac{e \cdot k^3}{\lambda \cdot C_s^2}\right)^\lambda .$$

□

Theorem 1. *Consider the d-choice game in which $m = C$ balls are allocated into n bins with total capacity C. Let κ be an arbitrary constant, and assume that either*
(1) $m \geq n^2$ or
(2) $C_s \leq c \cdot (n \cdot \ln(n))^{2/3}$ for an arbitrary positive constant c.
Then $\ell_{max} \leq 6\kappa$ with probability at least $1 - n^{-\kappa}$.

Proof. Since $\ell_{max} \leq \ell_{max}^{(b)} + \ell_{max}^{(s)}$ and since Observation 1 states that $\ell_{max}^{(b)}$ is constant, it remains to show that $\ell_{max}^{(s)}$ is also constant.

In Lemma 2 we calculate upper bounds for $|\mathcal{B}_s|$ and $\ell_{max}^{(s)}$, as a function of both m and C_s. Here we consider six cases, depending on the values of C_s and m. The first three cases imply statement (1), the last three cases imply statement (2). To apply Lemma 2 we will choose k so that $\mathbf{Pr}\left[\, X_s \geq k \,\right] \leq n^{-\alpha}$ for any constant

α (provided that n is large enough). In two cases we will be able to choose k as a (small) multiple of α which already implies a constant maximum load.

Case 1: $\mathbf{C} \geq \mathbf{n^2}$ *and* $\mathbf{C_s} \in [\mathbf{1}, \mathbf{n^{3/4}}]$. In this case we use Lemma 2(1) to show that the size of $\mathcal{B}_s$ is w.h.p. at most a constant and clearly $\ell_{max}^{(s)} \leq |\mathcal{B}_s|$.

$$\mathbf{Pr}\left[\, X_s \geq k \,\right] \leq \left(\frac{e \cdot (C_s)^2}{k \cdot C}\right)^k \leq \left(\frac{e \cdot n^{3/2}}{k \cdot n^2}\right)^k = \left(\frac{e}{k \cdot n^{1/2}}\right)^k \leq \frac{1}{n^\kappa}$$

for $k \geq 2\kappa + 1$. From this we can directly derive that $\ell_{max}^{(s)} \leq 2 \cdot \kappa + 1 = \mathcal{O}(1)$ with probability $1 - n^{-\kappa}$.

Case 2: $\mathbf{C} \geq \mathbf{n^2}$ *and* $\mathbf{C_s} \in [\mathbf{n^{3/4}}, \mathbf{n}]$.

In this case we first use Lemma 2(1) to show that $|\mathcal{B}_s|$ is at most $\ln(n)$, w.h.p.

$$\begin{aligned} \mathbf{Pr}\left[\, X_s \geq \ln(n) \,\right] &\leq \left(\frac{e \cdot (C_s)^2}{\ln(n) \cdot C}\right)^{\ln(n)} \leq \left(\frac{e \cdot n^2}{\ln(n) \cdot n^2}\right)^{\ln(n)} = n^{-\ln\ln(n)+1} \\ &\leq n^{-\kappa}/2 \end{aligned}$$

Now we use Lemma 2(2) to show that $\ell_{max}^{(s)}$ is w.h.p. constant if the size of $|\mathcal{B}_s|$ is at most $\ln(n)$.

$$\mathbf{Pr}\left[\, Y \geq \lambda \;||\mathcal{B}_s| \leq \ln(n) \,\right] \leq \left(\frac{e \cdot \ln^3(n)}{\lambda \cdot n^{3/2}}\right)^\lambda \leq n^{-\kappa}/2$$

for $\lambda \geq \kappa$. Clearly, We have

$$\begin{aligned} \mathbf{Pr}\left[\, Y \geq \lambda \,\right] &= \mathbf{Pr}\left[\, Y \geq \lambda \;||\mathcal{B}_s| \leq \ln(n) \,\right] \cdot \mathbf{Pr}\left[\, |\mathcal{B}_s| \leq \ln(n) \,\right] \\ &\quad + \mathbf{Pr}\left[\, Y \geq \lambda \;||\mathcal{B}_s| > \ln(n) \,\right] \cdot \mathbf{Pr}\left[\, |\mathcal{B}_s| > \ln(n) \,\right] \\ &\leq \mathbf{Pr}\left[\, Y \geq \lambda \;||\mathcal{B}_s| \leq \ln(n) \,\right] + \mathbf{Pr}\left[\, |\mathcal{B}_s| > \ln(n) \,\right] \end{aligned}$$

Thus, with a probability of $1 - n^{-\kappa}$, we have $\ell_{max}^{(s)} \leq \kappa$.

Case 3: $\mathbf{m} \geq \mathbf{n^2}$ *and* $\mathbf{C_s} \in [\mathbf{n}, \mathbf{n} \cdot \mathbf{r} \cdot \ln(\mathbf{n})]$.

Here we use Lemma 2(1) again to show that $|B_s| \leq (r \cdot \ln(n))^3$, w.h.p. Recall, a bin is called big if its capacity is at least $r \cdot \ln n$.

$$\begin{aligned}
\mathbf{Pr}\left[\, X_s \geq (r \cdot \ln(n))^3 \,\right] &\leq \left(\frac{e \cdot (C_s)^2}{(r \cdot \ln(n))^3 \cdot C}\right)^{(r \cdot \ln(n))^3} \\
&\leq \left(\frac{e \cdot (n \cdot r \cdot \ln(n))^2}{(r \cdot \ln(n))^3 \cdot n^2}\right)^{(r \cdot \ln(n))^3} \\
&= \left(\frac{e}{r \cdot \ln(n)}\right)^{(r \cdot \ln(n))^3} = \left(\frac{n}{r^{\ln(n)} \cdot n^{\ln\ln(n)}}\right)^{r^3 \cdot \ln^2(n)} \\
&\leq n^{-\kappa}/2
\end{aligned}$$

for $r \geq 1$. Hence, using Equation 1 we get

$$\mathbf{Pr}\left[\, Y \geq \lambda \mid |\mathcal{B}_s| \leq (r \cdot \ln(n))^3 \,\right] \leq \left(\frac{e \cdot (r \cdot \ln(n))^9}{\lambda \cdot n^2}\right)^{\lambda} \leq n^{-\kappa}/2$$

for $\lambda \geq \kappa/2$. Similar to the last case we have $\ell_{max}^{(s)} \leq 2\kappa$ with a probability of $1 - n^{-\kappa}$.

Case 4: $\mathbf{C} \geq \mathbf{n} \cdot \ln(\mathbf{n})/\mathbf{2}$ *and* $\mathbf{C_s} \in [\mathbf{1}, (\mathbf{n} \cdot \ln(\mathbf{n}))^{\mathbf{5/12}}]$.

$$\mathbf{Pr}\left[\, X_s \geq k \,\right] \leq \left(\frac{e \cdot (n \cdot \ln(n))^{5/6}}{k \cdot 1/2 \cdot n \cdot \ln(n)}\right)^{k} = \left(\frac{2e}{k \cdot (n \cdot \ln(n))^{1/6}}\right)^{k} \leq n^{-\kappa}$$

for $k \geq 6\kappa$. This immediately yields $\ell_{max}^{(s)} \leq 6 \cdot \kappa$ with probability $1 - n^{-\kappa}$.

Case 5: $\mathbf{C} \geq \mathbf{n} \cdot \ln(\mathbf{n})/\mathbf{2}$ *and* $\mathbf{C_s} \in [(\mathbf{n} \cdot \ln(\mathbf{n}))^{\mathbf{5/12}}, (\mathbf{n} \cdot \ln(\mathbf{n}))^{\mathbf{7/12}}]$.
Again, we first show that $|\mathcal{B}_s| \leq (n \cdot \ln^2(n))^{1/6}$, w.h.p.

$$\begin{aligned}
\mathbf{Pr}\left[\, X_s \geq (n \cdot \ln^2(n))^{1/6} \,\right] &\leq \left(\frac{e \cdot (C_s)^2}{(n \cdot \ln^2(n))^{1/6} \cdot C}\right)^{(n \cdot \ln^2(n))^{1/6}} \\
&\leq \left(\frac{e \cdot (n \cdot \ln(n))^{7/6}}{(n \cdot \ln^2(n))^{1/6} \cdot 1/2 \cdot n \cdot \ln(n)}\right)^{(n \cdot \ln^2(n))^{1/6}} \\
&= \left(\frac{2e}{(\ln(n))^{1/6}}\right)^{(n \cdot \ln^2(n))^{1/6}} \leq n^{-\kappa}/2.
\end{aligned}$$

Again,

$$\begin{aligned}
\mathbf{Pr}\left[Y \geq \lambda \mid X_s < (n \cdot \ln^2(n))^{1/6}\right] &\leq \left(\frac{e \cdot k^3}{\lambda \cdot (C_s)^2}\right)^{\lambda} \\
&\leq \left(\frac{e \cdot (n \cdot \ln^2(n))^{1/2}}{\lambda \cdot (n \cdot \ln(n))^{5/6}}\right)^{\lambda} \\
&= \left(\frac{e \cdot \ln^{1/6}(n)}{\lambda \cdot n^{1/3}}\right)^{\lambda} \leq n^{-\kappa}/2
\end{aligned}$$

for $\lambda \geq 4\kappa$. Thus, with a probability of $1 - n^{-\kappa}$ we have $\ell_{max}^{(s)} \leq 4\kappa$.

Case 6: $\mathbf{C} \geq \mathbf{h} \cdot \mathbf{n} \cdot \ln(\mathbf{n})$ *and* $\mathbf{C_s} \in [(\mathbf{n} \cdot \ln(\mathbf{n}))^{\mathbf{7/12}}, \mathbf{c} \cdot (\mathbf{n} \cdot \ln(\mathbf{n}))^{\mathbf{2/3}}]$.
Again we first show (using Lemma 2(1) again) that w.h.p. $|\mathcal{B}_s| \leq (n \cdot \ln^2(n))^{1/3}$.

$$\begin{aligned}
\mathbf{Pr}\left[X_s \geq (n \cdot \ln^2(n))^{1/3}\right] &\leq \left(\frac{e \cdot (C_s)^2}{(n \cdot \ln^2(n))^{1/3} \cdot C}\right)^{(n \cdot \ln^2(n))^{1/3}} \\
&\leq \left(\frac{e \cdot c^2 \cdot (n \cdot \ln(n))^{4/3}}{(n \cdot \ln^2(n))^{1/3} \cdot 1/2 \cdot n \cdot \ln(n)}\right)^{(n \cdot \ln^2(n))^{1/3}} \\
&= \left(\frac{2e \cdot c^2}{\ln^{1/3}(n)}\right)^{(n \cdot \ln^2(n))^{1/3}} \leq n^{-\kappa}/2.
\end{aligned}$$

Now Lemma 2(2) gives us

$$\begin{aligned}
\mathbf{Pr}\left[X_c \geq \lambda\right] &\leq \left(\frac{e \cdot k^3}{\lambda \cdot (C_s)^2}\right)^{\lambda} \leq \left(\frac{e \cdot n \cdot \ln^2(n)}{\lambda \cdot (n \cdot \ln(n))^{7/6}}\right)^{\lambda} \\
&= \left(\frac{e \cdot \ln^{5/6}(n)}{\lambda \cdot n^{1/6}}\right)^{\lambda}.
\end{aligned}$$

for $\lambda \geq 7\kappa$. Similar to the last case, this implies that with a probability of $1 - n^{\kappa}$ $\ell_{max}^{(s)} \leq 6 \cdot \kappa$. □

Theorem 2. *Let $d \in \mathbb{N}$ with $d \geq 2$. Consider the d-choice game in which C balls are allocated into n bins with total capacity C. Let $\kappa \in (0, \frac{r}{3} - 1]$ (with r being the constant of the definion of "bigness" of a bin), and assume*

$$C_s \leq C^{\frac{d-1}{d}} \cdot (\log C)^{1/d}.$$

Then, with a probability of at least $1 - n^{-\kappa}$,

$$\ell_{max} \leq 2(\kappa + 4).$$

Proof. Notice that $C_s \leq r \cdot n \cdot \ln(n)$ where r is a positive constant (see the definition of smallness of a bin). Recall that the probability that a ball sends all d requests to only small bins is $(C_s/C)^d$. Let X_s denote the number of balls sending all d queries to small bins. Then

$$E[X_s] = C \cdot \left(\frac{C_s}{C}\right)^d = \frac{C_s^d}{C^{d-1}}.$$

Case 1: $\mathbf{C_s = 0}$. In this case all bins are big, and we can simply apply Observation 1 and find $\ell_{max} \leq 4$ with probability at least $1 - 1/n^\kappa$ whenever $0 < \kappa \leq \frac{r}{3} - 1$.

Case 2: $\mathbf{1 \leq C_s \leq C^{1/4}}$. In this case, the probability that a ball sends all d requests to small bins is

$$\left(\frac{C_s}{C}\right)^d \leq \left(\frac{C^{1/4}}{C}\right)^d = C^{-\frac{3}{4}d} \leq C^{-\frac{3}{2}}$$

(the last inequality holds as $d \geq 2$). The probability that there are k or more balls in $\mathcal{B}_s$ is

$$\begin{aligned}\mathbf{Pr}\left[\, X_s \geq k \,\right] &\leq \binom{C}{k} \cdot C^{-\frac{3}{2}k} \leq \left(\frac{Ce}{k}\right)^k \cdot C^{-\frac{3}{2}k} \\ &= C^{k-\frac{3}{2}k} \cdot \left(\frac{e}{k}\right)^k \leq C^{-k/2} \leq n^{-k/2} \leq n^{-(\kappa+2)}\end{aligned}$$

for $k \geq \max\{e, 2(\kappa+2)\} = 2(\kappa+2)$ (also recall that $C \geq n$). Of those balls sending all d queries to small bins, no small bin will receive more than k balls with probability at least $1 - 1/n^{\kappa+2}$.

According to Observation 1, the additional load of small bins due to balls that mix requests between small and big bins is at most four with probability $1 - 1/n^{\kappa+2}$, so long as $\kappa+2 \leq \frac{r}{3} - 1$. Hence, with probability at least $1 - 1/n^{\kappa+1}$, the maximum load of the small bins is at most $k+4$. For the maximum load of the big bins we can use Observation 1 again, and conclude that with a probability of at least $1 - 1/n^\kappa$, the maximum load of *any* bin is no greater than $4+k$ for any $k \geq 2(\kappa+2)$.

Case 3: $\mathbf{C^{1/4} < C_s \leq C^{\frac{d-1}{d}} \log^{1/d} C}$. In this case,

$$E[X_s] = \frac{C_s^d}{C^{d-1}} \leq \frac{\left(C^{\frac{d-1}{d}} \log^{1/d} C\right)^d}{C^{d-1}} = \frac{C^{d-1} \log C}{C^{d-1}} = \log C.$$

After applying Chernoff's inequality we obtain $\mathbf{Pr}\left[\, X_s > 2\log C \,\right] < n^{-\ell}$ for any constant $\ell > 0$ (this is because $C \geq n$, and we obtain an inversely polynomial probability already for $n/\log n$.

Let Y_s count the number of balls from $\mathcal{B}_s$ that fall into bins that already contain at least one ball of $\mathcal{B}_s$. Then,

$$\begin{aligned}
\mathbf{Pr}\left[Y_s \geq k \right] &\leq \binom{2\log C}{k} \cdot \left(\frac{2\log C}{C^{1/4}}\right)^{dk} \\
&\leq \left(\frac{2e\log C}{k}\right)^{k} \cdot \left(\frac{2\log C}{C^{1/4}}\right)^{dk} \\
&\leq \frac{\log^k(C) \cdot 2^{dk} \cdot \log^{dk} C}{C^{dk/4}} \\
&\leq \frac{C^d}{C^{dk/4}} = \frac{1}{C^{d(k/4-1)}} \leq \frac{1}{C^{k/2-2}}
\end{aligned}$$

provided that $k \geq 2 \cdot e$ and C is large enough. For $k \geq 2(\kappa + 4)$ we obtain

$$\mathbf{Pr}\left[Y_s \geq k \right] \leq \frac{1}{C^{k/2-2}} \leq \frac{1}{C^{\kappa+2}} \leq \frac{1}{n^{\kappa+2}}.$$

Hence, with a probability of at least $1 - 1/n^{\kappa+2}$ the maximum load of the small bins is at most k for $k \geq \max\{2e, 2(\kappa+4)\} = 2(\kappa+4)$. The remainder of this case may be dealt with as in Case 2 above.

In summary, we find that for any positive $\kappa < \frac{r}{3} - 1$ with a probability of at least $1 - 1/n^{\kappa}$, $\ell_{max} \leq 4 + k$ for $k \geq 2(\kappa + 4)$. □

Theorem 3. *Consider the d-choice game in which, for any constant $k \geq 1$, $m = C = n^k$ balls are allocated into n bins with total capacity C. Then,* w.h.p., *the maximum load is bounded by*

$$\frac{\ln\ln(n)}{\ln(d)} + \mathcal{O}(1).$$

Proof. Lemma 1 compares the process in which m balls are allocated into n bins of total capacity C with the process that throws m balls into C unit-sized bins and states that the maximum load of the former is stochastically dominated by the maximum load of the latter. By applying Theorem 1.1 of [10] on the standard game with m balls and $m = C$ bins, we obtain a bound on the maximum load that is also valid for the first process. *W.h.p.*, the maximum load is

$$\ell_{max} \leq \frac{\ln\ln(m)}{\ln(d)} + \mathcal{O}(1) \leq \frac{\ln\ln(n^k)}{\ln(d)} + \mathcal{O}(1) = \frac{\ln\ln(n)}{\ln(d)} + \mathcal{O}(1).$$

□

The next observation considers the game in which all bins have capacity $\bar{c}$ and in which m balls are allocated to n bins. The maximum load equals the maximum load of the standard game (in which m balls are allocated into n bins of capacity one) divided by $\bar{c}$. The observation follows from the main result in [11], namely a tight analysis for multiple- choice algorithms allocating an arbitrarily large number of unit-size balls into unit-size bins:

Theorem 4 (Corollary 1.4 in [11])**.** *If m balls are allocated into n bins using* GREEDY$[d]$ *with $d \geq 2$, then the number of balls in the fullest bin is $\frac{m}{n} + \frac{\ln \ln n}{\ln d} \pm O(1)$, w.h.p. (that is, the maximum height above average is $\frac{\ln \ln n}{\ln d} \pm O(1)$, w.h.p.).*

Our corresponding result is as follows.

Observation 2. *Assume we allocate m balls into n bins with capacity $\bar{c}$ each. For $d \geq 2$ the maximum load is* w.h.p.

$$\ell_{max} = \frac{1}{\bar{c}} \cdot \left(\frac{m}{n} + \mathcal{O}(\ln \ln(n)) \right).$$

For $m = n \cdot \bar{c}$ in particular we obtain

$$\ell_{max} = \frac{1}{\bar{c}} \cdot \left(\frac{n \cdot \bar{c}}{n} + \mathcal{O}(\ln \ln(n)) \right) = 1 + \frac{\mathcal{O}(\ln \ln(n))}{\bar{c}}.$$

Proof. Since all capacities are the same, namely $\bar{c}$, the loads are computed in the same way for all bins and every ball adds the same load to the total load regardless of where it is allocated. Therefore the allocation process equals that of the standard game in which all bins have capacity 1. For the number of balls in the fullest bin the bounds given in [18, 11] can be applied. Finally we get the load by dividing by the bin's capacity $\bar{c}$. ☐

The following corollary follows directly from the last observation.

Corollary 1. *If $\bar{c} \in \Omega(\ln \ln(n))$ and if $m = k \cdot n \cdot \bar{c}$ for some arbitrary k, the maximum load is $k + \mathcal{O}(1)$,* w.h.p.

Theorem 5. *Let $k > 0$ and $0 < \alpha \leq 1$ be constants. Consider the game in which $\alpha \cdot n$ bins have capacity $q(n)$ and all other bins have capacity smaller $q(n)$. If $q(n) \in \Omega(\ln \ln(n))$, then there is a probability distribution over the bins such that the maximum load will be constant* w.h.p. *after the allocation of $m = k \cdot C$ balls.*

Proof. Assign probability $\frac{1}{\alpha \cdot n}$ to all bins with capacity $q(n)$ and probability 0 to all others. Ignoring the bins with probability 0, we may consider this a game of $m = k \cdot C \leq k \cdot n \cdot q(n)$ balls and $\alpha \cdot n$ bins. Applying Observation 2 we obtain

$$\begin{aligned} \ell_{max} &\leq \frac{1}{q(n)} \cdot \left(\frac{m}{\alpha \cdot n} + \mathcal{O}(\ln \ln(\alpha \cdot n)) \right) \\ &\leq \frac{1}{q(n)} \cdot \left(\frac{k \cdot n \cdot q(n)}{\alpha \cdot n} + \mathcal{O}(\ln \ln(n)) \right) \\ &\leq \frac{k}{\alpha} + \frac{\mathcal{O}(\ln \ln(n))}{q(n)} \leq \frac{k}{\alpha} + \mathcal{O}(1) = \mathcal{O}(1). \end{aligned}$$

□

The last result implies that in some cases much better results for the maximum load are possible if one can choose the probabilities oneself.

4. Simulations

The purpose of the simulations in this section is two-fold. On the one hand we consider the games analysed in the previous section and demonstrate that the asymptotic bounds behave well in practice. On the other hand we introduce settings and evaluate the performance of the approach in models not covered previously in this paper. Among others we will present experiments indicating that our results also hold for a very small number of bins and for the heavily loaded case, which are, of course, settings important for many practical applications. Whereas the main focus is on the maximum load in the analytical section, we often consider complete distributions here.

In order to obtain meaningful results, the experiments are usually repeated 10,000 times, and the values plotted are the average values. If not stated otherwise, the probabilities are proportional to the capacities and the number of balls equals the total capacity. Within this section, we use the term large bins slightly different from the use of big bins within the evaluation section, as we do not formally define their capacity as being bigger than $r \cdot \ln n$ for a specific value of r. Nevertheless, we use the term large bins for bin sizes, where we expect a behavior similar to big bins in the evaluation section.

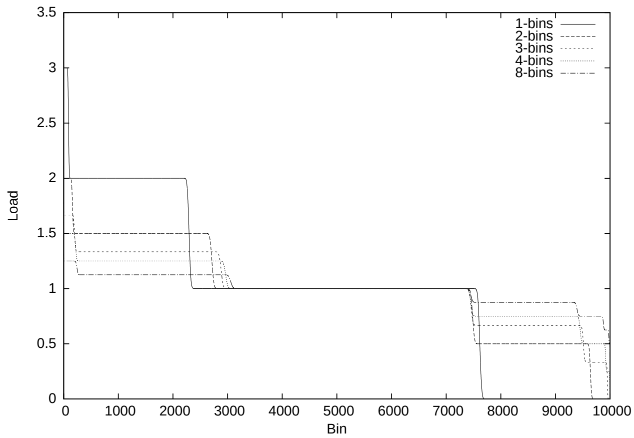

Figure 1: Uniform bins

4.1. Uniform Bins

This section analyses the influence of larger capacities on the load distribution for uniform bins. The setting is similar to the evaluation of the heavily loaded case presented in [11], as having bigger, but uniform capacities with $\forall c_i : c_i = c > 1$ and throwing $m = c \cdot n$ balls leads to the same distribution as the classical balls-into-bins strategy, as Algorithm 1 becomes identical to the standard d-choice game presented by Azar *et al.* [10].

In the first experiment, we have $n = 10{,}000$ bins, $d = 2$ and the uniform capacities have been set in the different experiments to $c = 1, 2, 3, 4, 8$. The capacities cover the interesting range between $\ln\ln(n) \approx 2.22$ and $\ln(n) \approx 9.21$.

In Figure 1 we plot the normalised load distribution (with load given by number of balls divided by capacity) of the entire bin vector for the five different (uniform) capacities. In the figure, "x-bins" refers to bins of capacity x.

According to Observation 2 the maximum load is $1 + \mathcal{O}(\ln\ln(n))/c$ for capacity $c \geq 2$ and $m = C = c \cdot n$. And in fact in our simulations the maximum load is very close to $1 + \ln\ln(n)/c$ for $c = 2, 3, 4, 8$ and close to $\ln\ln(n)/\ln(2)$ for $c = 1$

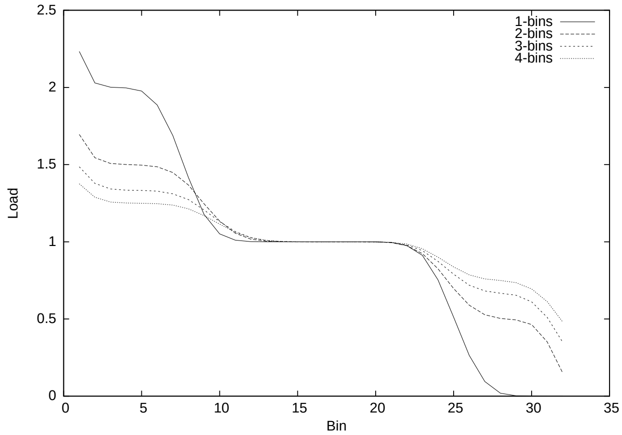

Figure 2: 32 uniform bins. Load distribution for C balls

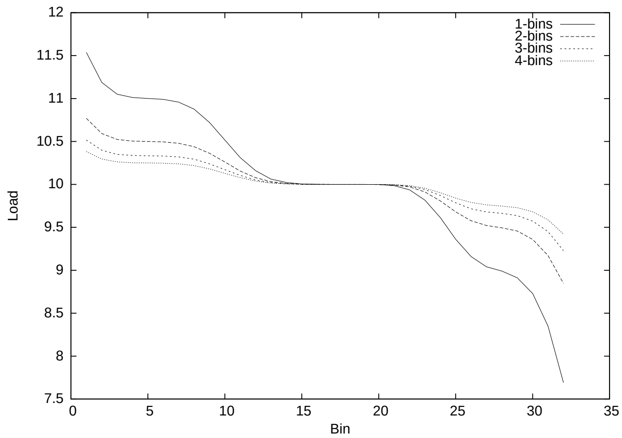

Figure 3: 32 uniform bins. Load distribution for $10 \cdot C$ balls

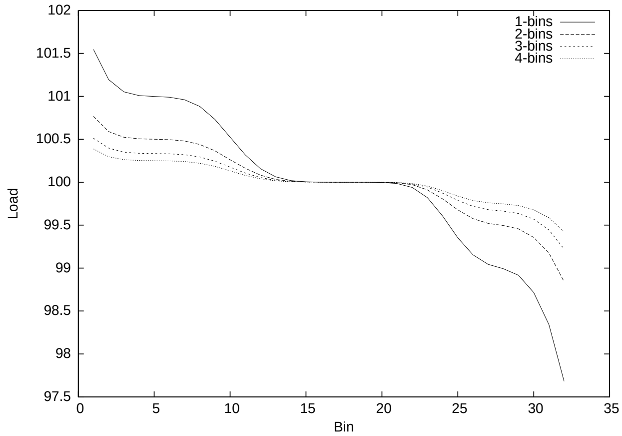

Figure 4: 32 uniform bins. Load distribution for $100 \cdot C$ balls

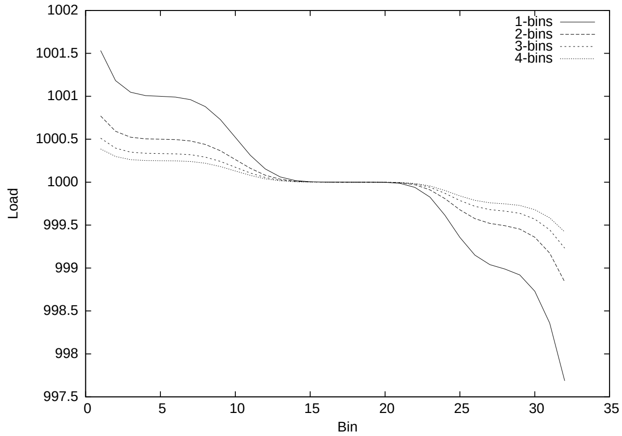

Figure 5: 32 uniform bins. Load distribution for $1{,}000 \cdot C$ balls

(see [10] or Theorem 3).

In Figure 2, 3, 4 and 5 we consider a smaller set of 32 uniform bins. The experiments show how an increase in the number of balls, m, affects the load distribution for different capacities $C = c \cdot n$ for $c = 1, 2, 3, 4$. The four plots show, top to bottom, left to right, the load distributions over the entire array of $n = 32$ bins, for $m = C, 10 \cdot C, 100 \cdot C, 1{,}000 \cdot C$ respectively. Observe how the absolute deviation from the average load m/n remains essentially invariant. In fact the curves for $m = 10 \cdot C, 100 \cdot C, 1{,}000 \cdot C$ look identical and suggest that this absolute deviation is independent of the number of balls, which corresponds to the theoretical results for the heavily loaded case in [11] with all bins and balls being uniform.

4.2. Load Distribution in Mixed Arrays

In this section we look at heterogeneous bin arrays. We assume that the number of balls m equals the total capacity C (unless stated otherwise) and that the bins' probabilities are proportional to their capacities. We fix the number of bins and increase the total capacity of the system.

In this scenario it is plausible that an increase of the total capacity leads to a decrease in the maximum load because the bigger bins draw balls and a ball in a big bin adds little to the total load. We will present a few simulations that substantiate this assumption. Moreover, we will analyse which type of bins are likely to hold the biggest load.

Figure 6 depicts the results from an experiment with a fixed number of $n = 1{,}000$ bins. We mix small bins of capacity 1 with large bins of capacity 10. The fraction of large bins is depicted on the x-axis and varies from 0% to 100%. The figure

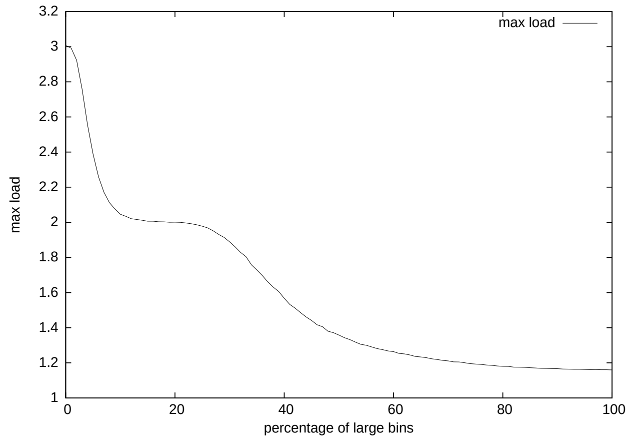

Figure 6: Bins of size 1 and 10. Maximum load depending on fraction of large bins.

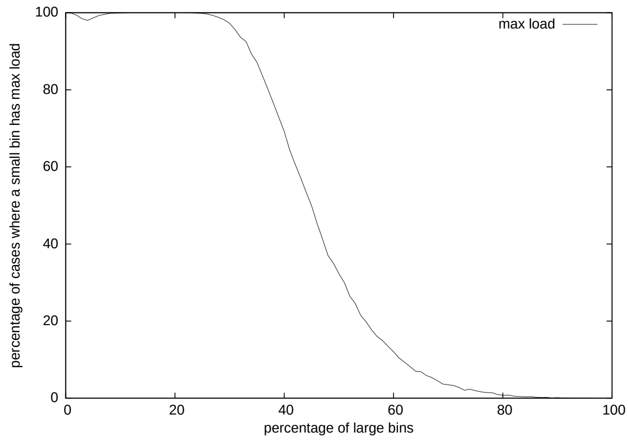

Figure 7: Bins of size 1 and 10. Location of maximally loaded bin.

shows how the maximum load changes when we increase the fraction of large bins and hence the total capacity. We can see clearly that, as expected, the maximum load decreases as the proportion of large bins increases.

Observe the slow decrease between 10% and 30% in Figure 6. The phenomenon can be linked to another one known to occur in standard (uniform) *balls-into-bins* games. There, the maximum load can be observed to remain invariant for a relatively large number of successive balls, during whose placement the *number* of maximally loaded bins grows, but not their (individual) loads. An image often used to describe this effect is that of horizontally growing a plateau. Clearly, as the probability for a ball to query only bins of maximum load (members of the plateau) grows proportionally with the width of the plateau, the maximum load will eventually increase by one, and a new plateau will be formed.

The large bins start to exert a pull effect from the beginning of our experiment, very quickly decreasing the maximum load from 3 to 2. Then the maximum load remains more or less unchanged until the large bins reach a fraction of 25 percent. Afterwards the fraction of large bins increases considerably until they are able to "pull" enough balls from the small bins such that the maximum load of the small bins drops to 1 (see also Figure 7). This happens when approximately 90% of the balls hit at least one big bin. From there on, the maximum load stays in one of the large bins and slowly decreases to 1.2.

Figure 7 gives a different view on and some explanations for the same experiment. The plot shows, for each point on the curve, the fraction of 1,000 independent runs in which a small bin of capacity 1 was among the maximally loaded. The maximum load is more likely to be in one of the small bins as long as the pull effect from the large bins is not too strong. With about 45% large bins the fraction of small bins containing the maximum load drops under 50%. Then the probability to choose at least one big bin is already $4{,}500/(4{,}500 + 550) > 0.89$. The figure suggests that small bins have the maximum load as long as the number of big bins is small, and that the plateau in Figure 6 coincides with the area where the maximum load is moving from the small bins to the large bins.

It is interesting to analyse the small dent around 2% in Figure 7. One reason for this phenomenon is the relatively high maximum load among the large bins, which results from insufficient load balancing. The load is poorly balanced among the large bins because most of the time a big bin is chosen, it is chosen together with a small bin. The second reason is that deviations in the maximum load of the small bins are more substantial than deviations in the maximum load of the big bins. In this particular case, the maximum load of the small bins is either 2 or 3 and the maximum load of the large bins usually slightly below 2, but not always. Thus, with a small probability, the maximum load is among the large bins.

The results presented in Figure 6 and 7 have shown how the maximum load moves from the small bins to the larger bins for an environment with two different bin sizes. Figure 8 and 9 now present results for environments with more than two bin sizes, where the maximum load is a function of the overall capacity.

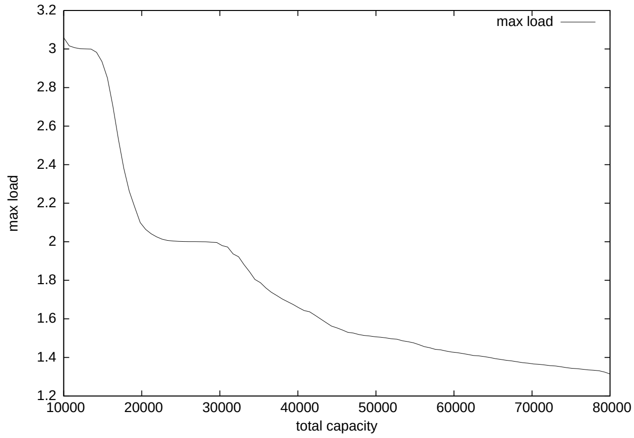

Figure 8: Randomised bin sizes. Maximum load depending on system capacity.

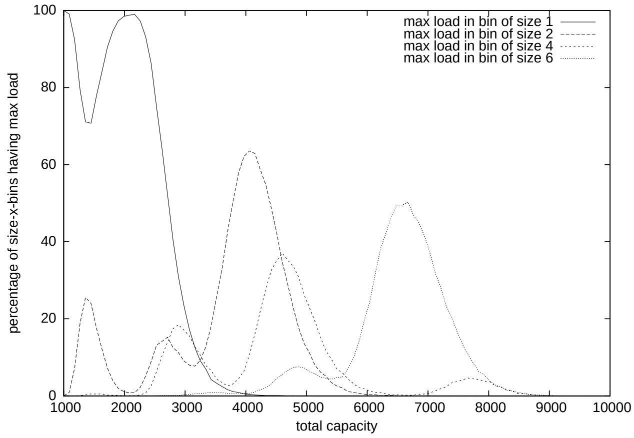

Figure 9: Randomised bin sizes. Location of maximally loaded bin.

In this case, the results are not obtained by gradually increasing the fraction of large fixed-size bins. Instead, we determine each bin's capacity using a random process in which, for a desired total capacity $C = c \cdot n$ (with c between 1 and 8) the size of each bin is determined by $1 + X$ where X is a binomially distributed random variable with $X \sim \mathbf{Bin}(7, \frac{c-1}{7})$, where 7 is the number of Bernoulli experiments and $(c-1)/7$ is the probability in each of the experiments. Notice that the total capacity will in general not be precisely equal to $c \cdot n$, but it can be shown, theoretically and experimentally, that it will be very close to it with high probability. The number of balls is always the same as the respective total capacity ($m = C$). The result is very similar to the previous experiment. While increasing the total capacity, the maximum load rapidly decreases.

The shapes of Figure 6 and Figure 8 seem to be similar, nevertheless, it could be seen that Figure 8 is less smooth and that multiple smaller plateaus try to emerge. These smaller plateaus can be explained by Figure 9. It can be seen that the smallest load is in the beginning in bins of size 1. This can be easily explained, as most of the bins have in this case a capacity of 1. The situation changes, when the capacity of the system becomes bigger than 2,500. In this case, the maximum load is moving to bins of capacity 2. This has two reasons: Firstly, the bigger bins generate a pull from the bins of size 1 to the bigger bins. Secondly, the number of size 1 bins decreases with increasing system capacity. The maximum load starts to decrease again at a system's capacity of 3,000. In this case, the bins of size 3 or 4 are dominating and receive this maximum load. Figure 8 and 9 nicely show that it makes sense to build heterogeneous systems and to add bigger bins to an already existing systems to achieve better load balance. The dents in Figure 9 can be explained in the same way as the dents for Figure 7.

This behaviour can also be seen in the plots of Figure 10 and 11. They show load

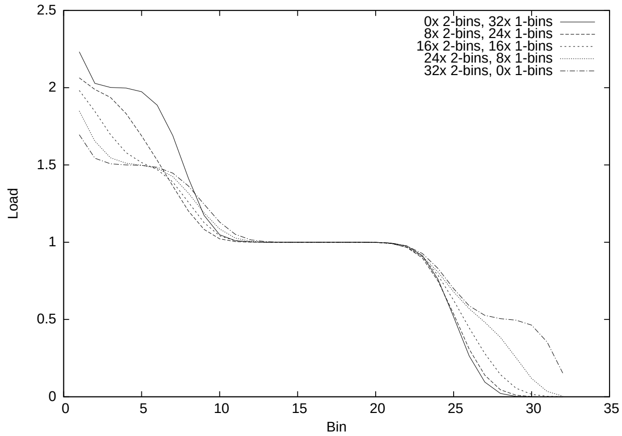

Figure 10: 32 bins of capacity 1 and 2

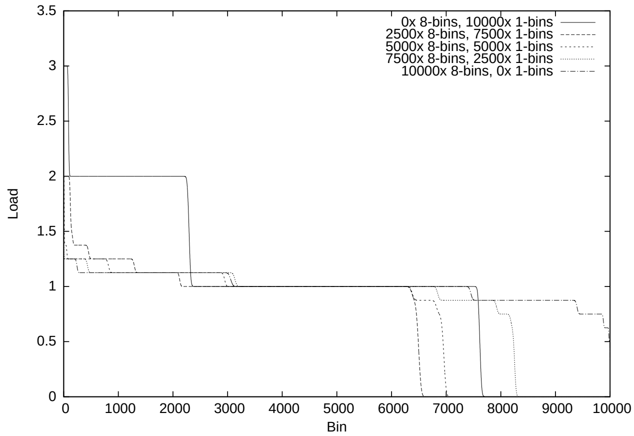

Figure 11: 10,000 bins of capacity 1 and 8

distributions for different ratios of small bins and large bins. Here, load stands for the average over 10,000 repetitions of the same experiment. We consider two cases: In the first plot we have only 32 bins, and the bin sizes are 1 and 2. In the second plot there are 10,000 bins, and the bin sizes are 1 and 8. We observe in both plots: The more large bins we have, the more even the load distribution becomes.

The experiment depicted in Figure 12 and 13 equals the one in Figure 11 as we consider the same ratios of small and large bins; sizes 1 and 8 respectively. We provide the results this time in two separate plots that complement each other. The left part shows only the bins of size 8, the right part only the bins of size 1. Notice that the curves do not generally span the entire width of the figures as there are simply not in general $n = 10{,}000$ bins of a given size available. Again, load denotes the average load over 10,000 repetitions.

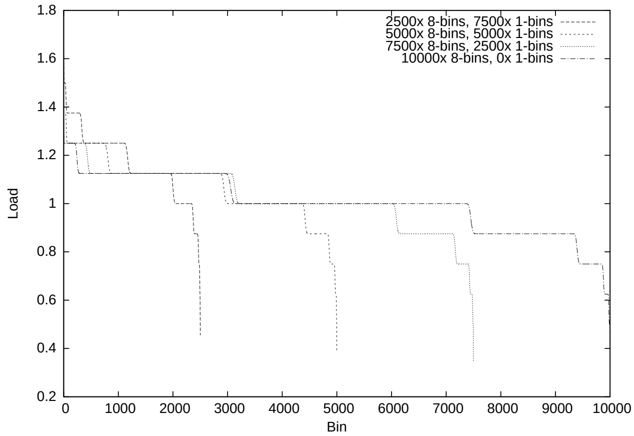

Figure 12: Bins of capacities 1 and 8. Load for bins of capacity 8.

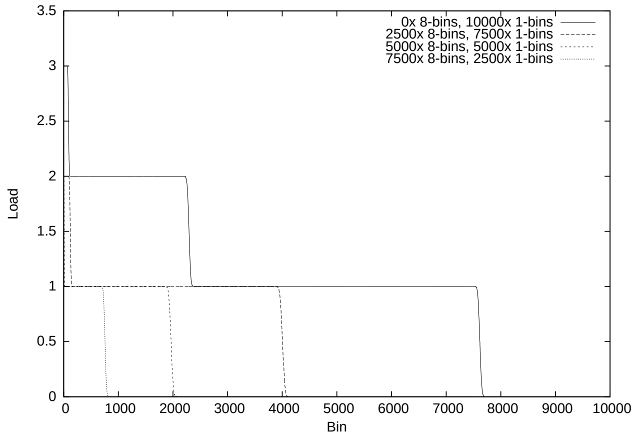

Figure 13: Bins of capacities 1 and 8. Load for bins of capacity 1.

Observation 1 and Theorem 3 predict a constant load in the large bins and higher loads in some small bins. We can observe that the asymptotic bounds behave very well in our experiment.

4.3. Dynamically Increasing the Number of Bins

This subsection discusses an issue to be observed in cloud computing and high performance computing (HPC) environments. The environment starts with a small number of storage systems and grows over time. New disks are bought in batches and each generation of disks is bigger than the previous one. Nevertheless, new disks do not replace the old ones, but the old disks remain in the system to improve both capacity and bandwidth. In our experiments, we scale the environment from two hard disks (bins) up to 1,000 disks. Each increase is by 20 disks. We assume two growth models: linear and exponential.

In the linear growth model, the i-th batch of new disks is bigger than the $(i-1)$-st batch by some constant offset. In the simulations, we start with capacity 2 for the first batch, and increase by $a = 1$, $a = 2$, $a = 4$, or $a = 6$ in each new batch. We perform an entire run of the experiment, from two disks all the way up to 1,000, for each of those values of a. The outcome can be seen in Fig. 14. The diagram also contains a graph for the "baseline setting" in which no growth at all is assumed but all disks have capacity 2.

In the exponential growth model, the i-th batch of new disks is bigger than the $(i-1)$-st batch by some constant factor. In the simulations, we start with capacity 2 for the first batch, and increase by a factor of $b = 1.005$, $b = 1.1$, $b = 1.2$, or $b = 1.4$. Again, we perform an entire run of the experiment, from two disks all the way up to 1,000, for each of those values of b. The outcome can be seen in Fig. 15. The diagram again also contains a graph for the "baseline setting" in which no growth at all is assumed but all disks have capacity 2.

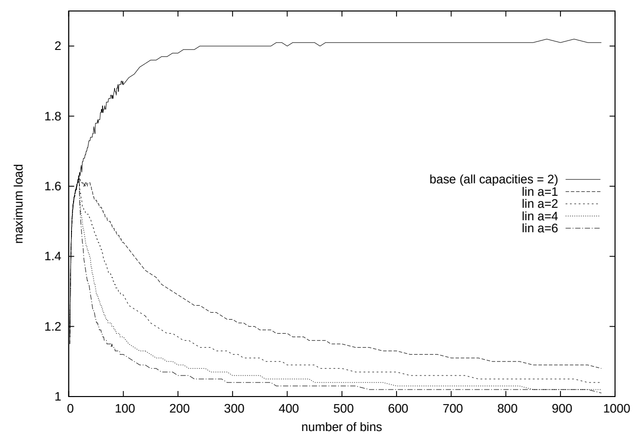

Figure 14: Linear growth between generations

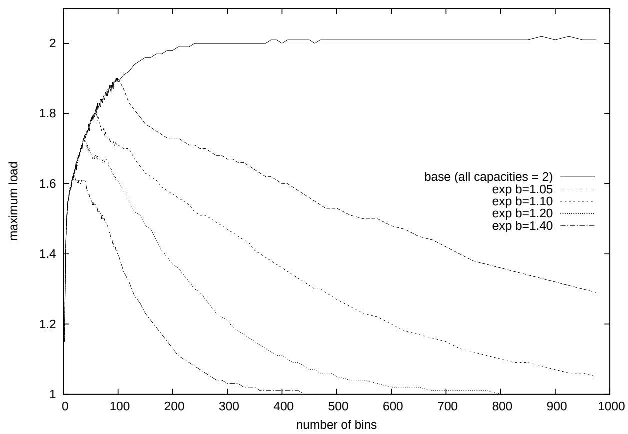

Figure 15: Exponential growth between generations

In both settings, linear and exponential, on adding a new batch of disks the simulation is started from scratch (data could of course be reallocated instead), and, most importantly, whatever the capacity distribution and total system capacity C are, we always allocate precisely C many balls. This results in an average load (#balls/capacity) of 1, and a maximum load of 1 is optimal in any of the settings.

As expected, it can be observed that the exponential growth model is a little slow to take off, but once the capacities of new batches are significant (and the factor is large enough, i.e., not 1.005) it clearly outperforms the linear growth model. It is notable that either model, with any parameter, exhibits a decreasing maximum load as function of the system size, quite unlike the baseline model.

Although in our experiments we allocate data from scratch when new disks arrive, it should be pointed out that a number of algorithms have been proposed and implemented which are able to perform a reorganization with minimum overhead (please see, e.g., [19, 20, 21, 5]).

4.4. The Heavily Loaded Case

In Figure 2, 3, 4 and 5 we have already seen an example for the heavily loaded case ($m \gg C$) when we simulated the uniform game in which all bins have the same capacity. We observed that, in accordance with Observation 2, the difference between the maximum load and the average load $\frac{m}{n}$ is independent of the number of balls m. In this section we find indication that the same holds if the bins have random capacities.

In the experiment that is depicted in Figure 16 we fix $n = 10{,}000$ as well as a total capacity CAP, a multiple of n. We then generate individual bin capacities such that the (expected) total capacity is equal to the prescribed capacity CAP, using an approach similar to that in Section 4.2. For each fixed value of CAP, we throw 100×CAP many balls into the systems and at certain points throughout this process plot the current deviation of the maximum load from the average load as a function of the number of balls currently in the system (that is, we measure this quantity after the $(i \cdot \mathrm{CAP})$-th ball for $i = 1, 2, \ldots, 100$).

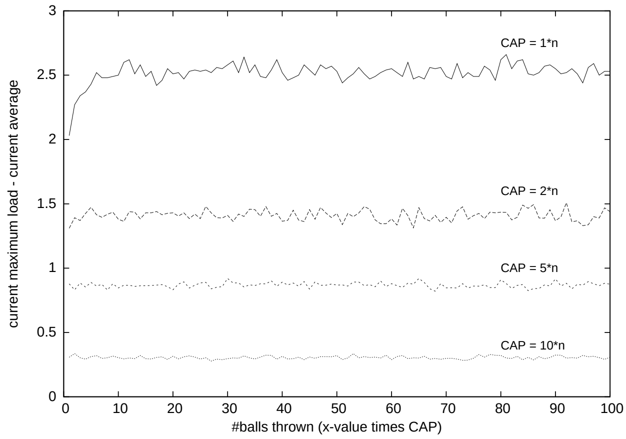

Figure 16: Heavily loaded, deviation of maximum from average

The plot shows curves for a variety of CAP-values. What we see is essentially a bundle of parallel lines, indicating that indeed the deviation of the maximum load from the average does not grow with the number of balls thrown, apparently regardless of the underlying total capacity. The positions of the lines also match our intuition and predictions as the lines get closer to zero as the total capacity increases, meaning the maximum load approaches the average load for large capacities. Notice that the curves slightly jiggle up and down. One might not expect such behaviour when tracing a term depending on the maximum load (which ought to be monotonic). However, the maximum load of each individual time step is still based on a random process, which fluctuates over time.

4.5. Optimal Probability Distribution

So far the probabilities were chosen to be proportional to the capacities. This is a natural approach and works well if the differences between the capacities are small. However, if this is not the case, it might be beneficial to use another strategy and alter the probabilities. Theorem 5 shows, for instance, that in certain cases in which a constant fraction of all capacities is of order $\ln\ln(n)$, a constant maximum load can be achieved by simply ignoring the low-capacity bins.

Let us consider the following setting: The number of bins is $n = 100$, half of them have capacity one and the other half (integer) capacity x, for $2 \le x \le 14$. The number of balls is $m = C = \sum_{i=1}^{n} c_i = \frac{n}{2} + x \cdot \frac{n}{2} = 50 \cdot (x+1)$, and the probability of a bin that has capacity c is set to $c^t/C(t)$ where $C(t) = \sum_{i=1}^{n} c_i^t$. Note that the probabilities sum up to 1 and that bins with the same capacity have the same probability. Since we have only two different capacities, all probabilities

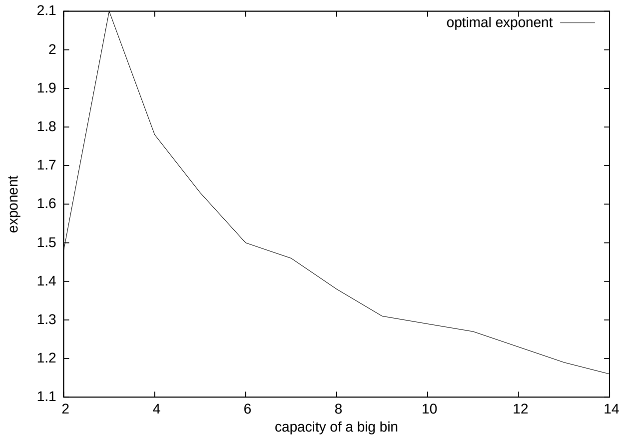

Figure 17: Optimal exponent for different capacities

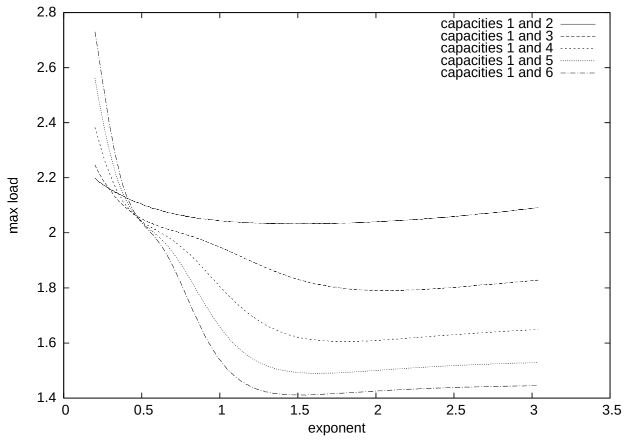

Figure 18: Max load for different exponents and capacities

are fixed as soon as the probability for one bin is set. For this probability, however, we can choose any value in the open interval $\left(0, \frac{2}{n}\right)$[1].

The question is, given x, what is the optimal exponent t? Figure 17 shows our experimental results for the simulation of the random allocation according to the altered probability distribution. For every capacity $c \in \{1, 2, ..., 14\}$ and every exponent $t \in \{1, 1.005, ..., 3\}$ the maximum load is averaged over 1,000,000 repetitions, and the best values for t are used in the plot. It shows that the

[1]Let p denote the probability for a bin with capacity $c \in \{1, x\}$ where $x \in \{2, 3, ...\}$. Since $\frac{c^t}{1+x^t}$ can take any value in $(0, 1)$, it follows:

$$p = \frac{c^t}{C(t)} = \frac{c^t}{\frac{n}{2} \cdot 1^t + \frac{n}{2} \cdot x^t} = \frac{2}{n} \cdot \frac{c^t}{1 + x^t} \in \left(0, \frac{2}{n}\right)$$

optimal exponent can differ considerably from 1. For the array in which 50 bins have capacity 1 and 50 bins capacity 3, the optimal exponent is about 2.1.

5. Conclusions

We have analysed the multiple-choice game with unit-sized balls and heterogeneous bins assuming that a bin's load is determined by the number of balls it contains divided by its capacity.

First we assumed that the probabilities of the bins to be selected by a ball are proportional to their capacities and that the number of balls equals the total capacity of the bins. For the maximum load we obtained a bound that is not worse than the one in the standard game [10], which is a special case of our model (all capacities set to one). For certain settings, we have been able to show that the maximum load can be even reduced to a constant by having a set of heterogeneous bins, as bigger bins are able to attract balls and help to reduce the load of smaller bins.

The experiments indicate that the asymptotic bounds are tight and that the strategy can even be employed in applications with a small number of bins, making it applicable in practical environments.

Interestingly and surprising, our analytical and experimental results show that it can be beneficial to choose different probability distributions over the heterogeneous bins.

To the best of our knowledge, even though it is of great practical relevance, this model has not previously been examined in a formal way. Future work could address the problem of finding a general upper bound for the maximum load in the heavily loaded case. Additionally, it would be interesting to further analyse the problem of choosing the best probability distribution for a given heterogeneous bin array.

References

[1] S. Ratnasamy, P. Francis, M. Handley, R. M. Karp, S. Shenker, A scalable content-addressable network, in: Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM), 2001, pp. 161–172.

[2] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. Kaashoek, F. Dabek, H. Balakrishnan., Chord: a scalable peer-to-peer lookup protocol for internet applications, IEEE/ACM Transactions on Networking 11 (1) (2003) 17 – 32.

[3] D. Fotakis, R. Pagh, P. Sanders, P. G. Spirakis, Space efficient hash tables with worst case constant access time, Theory Comput. Syst. 38 (2) (2005) 229–248.

[4] V. The Lam, M. Mitzenmacher, G. Varghese, Carousel: Scalable logging for intrusion prevention systems, in: Proceedings of the 7^{th} USENIX Symposium on Networked Systems Design and Implementation (NSDI), 2010, pp. 361–376.

[5] A. Miranda, S. Effert, Y. Kang, E. L. Miller, A. Brinkmann, T. Cortes, Reliable and randomized data distribution strategies for large scale storage systems, in: Proceedings of the 18^{th} International Conference on High Performance Computing (HiPC), 2011.

[6] D. R. Karger, E. Lehman, F. T. Leighton, R. Panigrahy, M. S. Levine, D. Lewin, Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web, in: Proceedings of the 29^{th} ACM Symposium on Theory of Computing (STOC), 1997, pp. 654–663.

[7] J. W. Byers, J. Considine, M. Mitzenmacher, Simple load balancing for distributed hash tables, in: Proceedings of the 2^{nd} International Workshop on Peer-to-Peer Systems (IPTPS), 2003, pp. 80–87.

[8] M. Bienkowski, M. Korzeniowski, F. Meyer auf der Heide, Dynamic load balancing in distributed hash tables, in: Proceedings of the 4^{th} International Workshop on Peer-to-Peer Systems (IPTPS), 2005, pp. 217–225.

[9] J. W. Byers, J. Considine, M. Mitzenmacher, Geometric generalizations of the power of two choices, in: Proceedings of the 16^{th} ACM Symposium on Parallel Algorithms and Architectures (SPAA), 2004, pp. 54–63.

[10] Y. Azar, A. Broder, A. Karlin, E. Upfal, Balanced allocations, in: Proceedings of the 26^{th} ACM Symposium on Theory of Computing (STOC), 1994, pp. 593–602.

[11] P. Berenbrink, A. Czumaj, A. Steger, B. Vöcking, Balanced allocations: the heavily loaded case, in: Proceedings of the 32^{nd} ACM Symposium on Theory of Computing (STOC), 2000, pp. 745–754.

[12] U. Wieder, Balanced allocations with heterogenous bins, in: Proceedings of the 19^{th} ACM Symposium on Parallel Algorithms and Architectures (SPAA), 2007, pp. 188–193.

[13] K. Kenthapadi, R. Panigrahy, Balanced allocation on graphs, in: Proceedings of the 17^{th} ACM-SIAM Symposium on Discrete Algorithms (SODA), 2006, pp. 434–443.

[14] B. Godfrey, Balls and bins with structure: Balanced allocations on hypergraphs, in: Proceedings of the 19^{th} ACM-SIAM Symposium on Discrete Algorithms (SODA), 2008, pp. 511–517.

[15] P. Berenbrink, A. Brinkmann, T. Friedetzky, L. Nagel, Balls into bins with related random choices, J. Parallel Distrib. Comput. 72 (2) (2012) 246 – 253.

[16] E. Even-Dar, A. Kesselman, Y. Mansour, Convergence time to nash equilibria, in: Proceedings of the 30^{th} International Colloquium on Automata, Languages and Programming (ICALP), 2003, pp. 502–513.

[17] C. P. J. Adolphs, P. Berenbrink, Distributed selfish load balancing with weights and speeds, in: Proceedings of the 31^{st} ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC), 2012, pp. 135–144.

[18] M. Raab, A. Steger, "Balls into Bins" - a simple and tight analysis, in: Proceedings of the 2^{nd} International Workshop Randomization and Approximation Techniques in Computer Science (RANDOM), 1998, pp. 159–170.

[19] A. Brinkmann, K. Salzwedel, C. Scheideler, Compact, adaptive placement schemes for non-uniform requirements, in: Proceedings of the 14^{th} ACM Symposium on Parallel Algorithms and Architectures (SPAA), 2002, pp. 53–62.

[20] R. J. Honicky, E. L. Miller, Replication under scalable hashing: A family of algorithms for scalable decentralized data distribution, in: Proceedings of the 18^{th} IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2004.

[21] S. A. Weil, S. A. Brandt, E. L. Miller, D. D. E. Long, C. Maltzahn, Ceph: A scalable, high-performance distributed file system., in: Proceedings of the 7^{th} Symposium on Operating Systems Design and Implementation (OSDI), 2006, pp. 307–320.