Theoretical Computer Science 409 (2008) 511–520

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

On weighted balls-into-bins games

Petra Berenbrink [a,*], Tom Friedetzky [b], Zengjian Hu [a], Russell Martin [c]

[a] School of Computing Science, Simon Fraser University, Burnaby, B.C., V5A 1S6, Canada
[b] Department of Computer Science, University of Durham, Durham, DH1 3LE, UK
[c] Department of Computer Science, University of Warwick, Coventry, CV4 7AL, UK

ARTICLE INFO

Article history:
Received 21 February 2008
Received in revised form 1 September 2008
Accepted 5 September 2008
Communicated by P. Spirakis

Keywords:
Balls-into-bins games
Random resource allocation
Weighted balls

ABSTRACT

We consider the well-known problem of randomly allocating m balls into n bins. We investigate various properties of *single-choice games* as well as *multiple-choice games* in the context of *weighted balls*. We are particularly interested in questions that are concerned with the *distribution* of ball weights, and the *order* in which balls are allocated. Do any of these parameters influence the maximum expected load of any bin, and if yes, then how?

The problem of weighted balls is of practical relevance. Balls-into-bins games are frequently used to conveniently model load balancing problems. Here, weights can be used to model resource requirements of the jobs, i.e., memory or running time.

1. Introduction

The *balls-into-bins game*, also referred to as *occupancy problem* or *allocation process*, is a well known and much investigated model. The goal of a (static) balls-into-bins game is to sequentially allocate, at random, a set of m independent balls (tasks, jobs, ...) into a set of n bins (printers, servers, ...), such that the maximum number of balls in any bin is minimised. In the dynamic case, we do not have a fixed number of balls but rather new balls arrive over time (and existing ones may be removed).

In this paper, we are interested in *static sequential games*, where a fixed number of balls, m, are allocated one after the other; see [9] for an overview of balls-into-bins games in different settings. The classical *single-choice game* allocates each ball to a bin that is chosen independently and uniformly at random (i.u.r.). For $m = n$ balls and n bins the maximum *load* (maximum number of balls) in any bin is $\Theta(\log(n)/\log\log(n))$. More generally, for m balls and n bins the maximum load is $(m/n) + \Theta(\sqrt{m \log n/n})$. Surprisingly, the maximum load can be decreased dramatically by allowing every ball to i.u.r. choose a small number of $d > 1$ bins. The ball is then allocated to one of the least loaded of the d chosen bins. Then, the maximum load drops to $\Theta(\log\log(n)/\log(d))$ in the $m = n$ case (see [1]), and $(m/n) + \Theta(\log\log(n)/\log(d))$ in the general case, respectively (see [2]). Notice that the results cited above all hold *with high probability*[1] (w.h.p.). Following [1], we refer to the multiple-choice algorithm defined above as Greedy[d].

Most work done so far assumes that the balls are *uniform* and *indistinguishable*. In this paper, we concentrate on the *weighted case* where the i-th ball comes with a weight w_i. We define the *load* of a bin to be the sum of the weights of the balls allocated to it. In [5] the authors compare the maximum load of weighted balls-into-bins games with the maximum load of corresponding uniform games. They compare the maximum load of a game with m weighted balls with maximum weight 1 and total weight $W = w_1 + \cdots + w_m$ to a game with approximately $4W$ uniform balls. Basically, they show that the

* Corresponding author. Tel.: +1 49524246951.
E-mail address: petra@cs.sfu.ca (P. Berenbrink).

[1] We say an event A occurs with high probability, if $\Pr[A] \geq 1 - 1/n^{\alpha}$ for some constant $\alpha \geq 1$.

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2008.09.023

maximum load of the weighted game is not larger than the load of the game with uniform balls (which has a slightly larger total weight). Their approach can be used for a variety of balls-into-bins games and can be regarded as a general framework.

However, the results of [5] seem to be somewhat unsatisfactory. The authors compare the allocation of a (possibly huge) number of "small" weighted balls with an allocation of fewer but "heavier" uniform balls. Intuitively, it should be clear that it is better to allocate many "small" balls compared to fewer "big" balls. After all, the many small balls come with more random choices. The main goal of this paper is to get tighter results for the allocation of weighted balls, both for the single-choice and the multiple-choice game. To show our results we will use the majorisation technique introduced in [1].

1.1. Known results

Single-choice game. In [9] the authors give a tight bound on the maximum load of any bin when m *uniform* balls are allocated uniformly at random into n bins. [11] considers (among other results) the following problem: Assume the total weight of the balls W and the maximum ball weight $w_{\max}$ is fixed. Then the expected maximum load is is maximised for $W/w_{\max}$ balls of weight $w_{\max}$. In [7] Koutsoupias et al. consider the random allocation of weighted balls. Similar to [5], they compare the maximum load of an allocation of weighted balls to that of an allocation of a smaller number of uniform balls with a larger total weight. They repeatedly fuse the two smallest balls together to form one larger ball until the weights of all balls are within a factor of two of each other. They show that the bin loads after the allocation of the weighted balls are *majorised* by the loads of the bins after the allocation of the balls generated by the fusion process. Their approach also applies to more general games in which balls can be allocated into bins with nonuniform probabilities.

Multiple-choice game. During recent years much research has been done for games with multiple choices in different settings. See [9] for a nice overview. Here, we shall only mention the "classical" and most recent results, and the results for weighted balls.

The case where each of the m balls has unit weight has been studied extensively. Azar et al. [1] introduced Greedy[d] to allocate n balls into n bins. Their algorithm Greedy[d] chooses d bins i.u.r. for each ball and allocates the ball into a bin with minimum load. They show that after placing n balls the maximum load is $\Theta(\log\log(n)/\log(d)+1)$, w.h.p. Compared to single-choice games, this is an *exponential* decrease of the maximum load. Vöcking [12] introduced the Always-Go-Left protocol yielding a maximum load of $(\log\log n)/d$, w.h.p. In [2] the authors analyse Greedy[d] for $m \gg n$. They show that the maximum load is $m/n + \log\log(n)/\log d$, w.h.p. This shows that the multiple-choice process behaves inherently differently from the single-choice process, where it can be shown that the difference between the maximum load and the average load depends on m. They also show a *memorylessness* property of the Greedy process, i.e., whatever the situation is after allocation of some ball, after sufficiently many additional balls the maximum load of any bin can again be bounded as expected. Finally, Mitzenmacher et al. [10] show that a similar performance gain occurs if the process is allowed to store the location of the least loaded bin in memory.

In [5] the authors present a general framework for multiple choice games that relates the results of weighted balls-into-bins games back to game for uniform balls. They show that the maximum load of a game with m weighted balls with maximum weight 1 and total weight W is not larger than the one resulting by the allocation of $4W$ uniform balls with weight one. See [4] for more details.

In [6] the authors generalize the results of [2] to the weighted case where balls have weights drawn from an arbitrary weight distribution with a finite second moment. They show that the gap between the weight of the heaviest bin and the weight of the average bin is independent of the number balls thrown. Furthermore, if the fourth moment of the weight distribution is finite, the expected value of the gap is shown to be independent of the number of balls.

1.2. Model and definitions

We assume that we have m balls and n bins. In the following we denote the set $\{1,\ldots,m\}$ by $[m]$. Ball i has weight w_i for all $i \in [m]$. Let $w = (w_1,\ldots,w_m)$ be the *weight vector* of the balls. We assume $w_i > 0$ for all $i \in [m]$. $W = \sum_{i=1}^{m} w_i$ is the *total weight* of the balls. If $w_1 = \cdots = w_m$ we refer to the balls as *uniform*. In this case, we normalise the ball weights such that $w_i = 1$ for $\forall i \in [m]$.

The *load* of a given bin is the sum of the weights of all balls allocated to it. In the case of uniform balls the load is simply the number of balls allocated to the bin. The status of an allocation is described by a *load vector* $L(w) = (\ell_1(w),\ldots,\ell_n(w))$. Here, ℓ_i is the load of the i-th bin after the allocation of weight vector w. Whenever the context is clear we write $L = (\ell_1,\ldots,\ell_n)$. In some cases we consider the change that occurs in an allocation after allocating some number of additional balls. Then we define L_t to be the load vector after the allocation of the first t balls with weights $w_1,\ldots,w_t$ for $1 \le t \le m$. In many cases we will *normalise* a load vector L by assuming a non-increasing order of bin loads, i.e. $\ell_1 \ge \ell_2 \ge \cdots \ge \ell_n$. We then define $S_i(w) = \sum_{j=1}^{i} \ell_j(w)$ as the total load of the i highest-loaded bins. Again, when the context is clear we shall drop the "w" and write $S_i = \sum_{j=1}^{i} \ell_j$. Finally, in what follows, we let $\Omega = [n]$.

To compare two load vectors and also the balancedness of vectors of balls weights we use the concept of *majorisation*. First, we briefly review the notion of majorisation from [8].

Definition 1.1. For two normalised vectors $w = (w_1,\ldots,w_m) \in \mathbb{R}^m$ and $w' = (w'_1,\ldots,w'_m) \in \mathbb{R}^m$ with $\sum_{i=1}^{m} w_i = \sum_{i=1}^{m} w'_i$, we say that w' *majorises* w, written $w' \succ w$, if $\sum_{i=1}^{k} w'_i \ge \sum_{i=1}^{k} w_i$ for all $1 \le k \le m$.

Majorisation is a strict partial ordering between (normalised) vectors of the same dimensionality. Intuitively, vector v' *majorises* another vector v if v is "more spread out", or "more balanced", than v'. In the following, if we refer to a weight vector w that is more balanced than weight vector w', we mean that w' majorises w. We will use the term majorisation if we refer to load vectors.

Some examples are:

$$(1, \underbrace{0, \ldots, 0}_{m-1}) \succ (1/2, 1/2, \underbrace{0, \ldots, 0}_{m-2}) \succ \cdots \succ (\underbrace{1/(m-1), \ldots, 1/(m-1)}_{m-1}, 0) \succ (\underbrace{1/m, \ldots, 1/m}_{m}).$$

1.3. New results

In the next section, we first present some additional definitions that we will use later on in this paper. Section 3 is concerned with the single-choice game. In Theorem 3.1 we fix the number of balls and show that the expected maximum load is smaller for more balanced ball weight vectors. In more detail, we allocate two sets of balls into bins, where the first set has a more even weight distribution than the second one, i.e., the second corresponding weight vector majorises the first one. We show that the expected maximum load after allocating the first set is smaller than the one after allocation the second set. This also holds for the sum of the loads of the i largest bins. One could say that the *majorisation is preserved*: if one weight vector majorises another one, then we have the same order with respect to the resulting expected bin load vectors. Hence, uniform balls minimise the expected maximum load. Theorem 3.1 uses majorisation together with T-transformations (see the definition in the next section), thereby allowing us to compare sets of balls that only differ in *one* pair of balls.

Corollary 3.4 extends the results showing that the allocation of a large number of small balls with total weight W ends up with a smaller expected maximum load than the allocation of a smaller number of balls with the same total weight. We also show that the results are still true for many other random functions that are used to allocate the balls into the bins. Our results are much stronger than the ones of [7] since we compare arbitrary weight distributions with the same total weight. Compared to [7] we also allow for the same number of balls. In addition, we consider the entire load distribution and not only the maximum load.

Section 4 deals with multiple-choice games. The main result here is Theorem 4.6. It shows that, for sufficiently many balls, allocation of uniform balls is *not* necessarily better than allocation of weighted balls. It is better to allocate first the "big balls" and then some smaller balls on top of them, instead of allocating the same number of average sized balls. This result uses the memorylessness property of [2]. For fewer balls we show in Lemma 4.7 that the majorisation order is not generally preserved. Assume that we have two allocations $\mathcal{A}$ and $\mathcal{B}$, and that the load vector of allocation $\mathcal{A}$ is majorised by the load vector of allocation $\mathcal{B}$. Now, throwing only *one* additional ball into both allocations may reverse the majorisation order and suddenly $\mathcal{B}$ is majorised by $\mathcal{A}$ (or possibly the two load vectors could be noncomparable under the majorisation partial ordering).

The previous results mentioned for the single-choice game use the majorisation technique inductively. Unfortunately, it seems difficult to use T-transformations and the majorisation technique to obtain results for weighted balls in the multiple-choice game. We also present several examples showing that, for the case of a small number of balls with multiple-choices, the expected maximum load is not necessarily smaller if we allocate more evenly weighted balls.

2. Majorisation and *T*-transformations

In Section 1.2 we defined the concept of majorisation. In [1] Azar et al. use this concept for random processes. Here we give a slightly different definition adjusted for our purposes.

Definition 2.1 (*Majorisation*). Let w and w' be two weight vectors with m balls, and let Ω^m be the set of all possible random choices for Greedy applied on m balls. Define $w(\omega)$ (respectively, $w'(\omega)$) to be the allocation resulting from the choices $\omega \in \Omega^m$, and let $f : \Omega^m \longrightarrow \Omega^m$ be a *one-to-one* correspondence. Then we say that w' is majorised by w if there exists a function f such that for any $\omega \in \Omega^m$ we have $w(\omega) \succ w'(f(\omega))$.

A slightly weaker form of the majorisation is the *expected majorisation* defined below. We will use it in order to compare the allocation of two different load vectors with each other.

Definition 2.2 (*Expected Majorisation*). Let w and w' be two weight vectors with m balls, and let Ω^m be the set of all possible random choices. Let $L(w, \omega) = (\ell_1(w, \omega), \ldots, \ell_n(w, \omega))$ (resp., $L'(w', \omega) = (\ell_1(w', \omega), \ldots, \ell_n(w', \omega))$) be the normalised load vector that results from the allocation of w (respectively, w') using $\omega \in \Omega^m$. Let $S_i(w, \omega) = \sum_{j=1}^{i} \ell_j(w, \omega)$ and $S_i(w', \omega) = \sum_{j=1}^{i} \ell_j(w', \omega)$. Then we say that $L(w')$ is *expectedly majorised* by $L(w)$ if for all $i \in [n]$, we have $E[S_i(w)] \geq E[S_i(w')]$. (The expectation is over all possible n^m elements, selected uniformly at random, in Ω^m.)

Now we introduce a class of linear transformations on vectors called T-*transformations* which are crucial to our later analysis. We write

$$w \overset{T}{\Longrightarrow} w',$$

meaning that w' can be derived from w by applying one T-transformation. Recall that a square matrix $\Pi = (\pi_{ij})$ is said to be *doubly stochastic* if all $\pi_{ij} \geq 0$, and each row sum and column sum is one. Π is called a *permutation matrix* if each row and each column contains exactly one unit and all other entries are zero (in particular, a permutation matrix is doubly stochastic).

Definition 2.3 (*T-transformation*). A *T-transformation* matrix T has the form $T = \lambda I + (1-\lambda)Q$, where $0 \leq \lambda \leq 1$, I is the identity matrix, and Q is a permutation matrix that swaps exactly two coordinates. Thus, for some vector x of correct dimensionality, $xT = (x_1, \ldots, x_{j-1}, \lambda x_j + (1-\lambda)x_k, x_{j+1}, \ldots, x_{k-1}, \lambda x_k + (1-\lambda)x_j, x_{k+1}, \ldots, x_m)$.

T-transformations and majorisation are closely linked by the following lemma (see [8]).

Lemma 2.4. *For $w, w' \in \mathbb{R}^m$, $w \succ w'$ if and only if w' can be derived from w by successive applications of at most $m-1$ T-transformations.*

One of the fundamental theorems in the theory of majorisation is the following.

Theorem 2.5 (*Hardy, Littlewood and Pólya, 1929*). *For $w, w' \in \mathbb{R}^m$, $w \succ w'$ if and only if $w' = wP$, for some doubly stochastic matrix P.*

3. Weighted Single-choice Games

In this section we study the classical balls-into-bins game where every ball has only one random choice. Let w and w' be two m-dimensional weight vectors. Recall that $S_i(w)$ is defined to be the random variable counting the cumulative loads of the i largest bins after allocating w. $S_i(w, \omega)$ counting the cumulative loads of the i largest bins after allocating w using the random choices ω. In this section we show that, if there exist a majorisation order between two weight vectors w and w', the same order holds for $E[S_i(w)]$ and $E[S_i(w')]$. This implies that, if w majorises w', the expected maximum load after allocating w is larger than or equal to the expected maximum load after allocating w'.

The proof of the following theorem is due to an anonymous reviewer of the paper. Our original proof (see [3]) used the majorization technique.

Theorem 3.1. *If $w \succ w'$, then $E[S_i(w)] \geq E[S_i(w')]$ for all $i \in [n]$.*

Proof. Fix an arbitrary allocation $\omega \in \Omega^n$. We first prove a lemma which indicates that the function $S_i(w, \omega)$ is convex.

Lemma 3.2. *The function $S_i(w, \omega)$ is convex.*

Proof. Let v and v' be two m-dimensional vectors such that $w = (1-\lambda)v + \lambda v'$.

$$\begin{aligned}
S_i(w,\omega) &= \max_{A\subset[n],|A|=i} \sum_{1\leq j\leq m\,:\,\omega_j\in A} w_j \\
&= \max_{A\subset[n],|A|=i} \sum_{1\leq j\leq m\,:\,\omega_j\in A} ((1-\lambda)v_j + \lambda v'_j) \\
&\leq (1-\lambda) \max_{B\subset[n],|B|=i} \sum_{1\leq j\leq m\,:\,\omega_j\in B} v_j + \lambda \max_{B'\subset[n],|B'|=i} \sum_{1\leq j\leq m\,:\,\omega_j\in B'} v'_j \\
&= (1-\lambda)S_i(v,\omega) + \lambda S_i(v',\omega). \quad \blacksquare
\end{aligned}$$

Now we are ready to prove the theorem using an induction. If $w \succ w'$ then w' can be derived from w by $\ell \leq m-1$ T-transformations (see Lemma 2.4). Let

$$w \overset{T}{\Longrightarrow} w_1, \quad w_1 \overset{T}{\Longrightarrow} w_2, \quad \ldots \quad w_{\ell-1} \overset{T}{\Longrightarrow} w'.$$

Since $w \overset{T}{\Longrightarrow} w_1$ we have $w_1 = \lambda \cdot w + (1-\lambda)w \cdot P$, where P is a permutation matrix. The same holds for $w_1 \overset{T}{\Longrightarrow} w_2, \ldots, w_{\ell-1} \overset{T}{\Longrightarrow} w'$. Using Lemma 3.2, we get

$$\begin{aligned}
E[S_i(w_1)] &\leq \lambda E[S_i(w)] + (1-\lambda)E[S_i(wP)] \\
&= E[S_i(w)].
\end{aligned}$$

The second equation holds since $E[S_i(w)] = E[S_i(wP)]$ for any permutation matrix P. Again, we get the same inequality for all T-transformations $w_1 \overset{T}{\Longrightarrow} w_2, \ldots, w_{\ell-1} \overset{T}{\Longrightarrow} w'$, yielding $E[S_i(w)] \geq E[S_i(w')]$. $\blacksquare$

It is clear that the uniform weight vector is majorised by all other vectors with same dimension and same total weight. Using Theorem 3.1, we get the following corollary.

Table 1
Allocations $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$

Allocations	First $m/2$ balls		Last $m/2$ balls	
	Ball weights	Algorithm	Ball weights	Algorithm
$\mathcal{A}$	3	Greedy[d]	1	Greedy[d]
$\mathcal{B}$	2	Greedy[d]	2	Greedy[d]
$\mathcal{C}$	3	Optimal	1	Greedy[d]
$\mathcal{D}$	2	Optimal	2	Greedy[d]

Corollary 3.3. *Let $w = (w_1, \ldots, w_m)$, $W = \sum_{i=1}^{m} w_i$, and $w' = (\frac{W}{m}, \ldots, \frac{W}{m})$. For all $i \in [n]$, we have $E[S_i(w)] \geq E[S_i(w')]$.*

Proof. Note that $w' = wP$, where $P = (p_{ij})$ and $p_{ij} = 1/m\ \forall i, j \in [m]$. Clearly P is a doubly stochastic matrix. Hence by Lemma 2.5, $w \succ w'$. Consequently, from Theorem 3.1 we have $E[S_i(w)] \geq E[S_i(w')]$. ■

Theorem 3.1 also shows that an allocation of a large number of small balls with total weight W ends up with a smaller expected load than the allocation of a smaller number of balls with the same total weight. Note that in the next corollary the relation $w \succ w'$ must be treated somewhat loosely because the vectors do not necessarily have the same length, but the meaning should be clear, namely that $\sum_{i=1}^{j} w_i \geq \sum_{i=1}^{j} w'_i$ for all $j \in [m]$.

Corollary 3.4. *Let $w = (w_1, \ldots, w_m)$ and $W = \sum_{i=1}^{m} w_i$. Suppose that $w' = (w'_1, \ldots, w'_{m'})$ with $m \leq m'$, and also that $W = \sum_{i=1}^{m} w'_i$. If $w \succ w'$ we have $E[S_i(w)] \geq E[S_i(w')]$ for all $i \in [n]$.*

Proof. Simply add zeros to w until it has the same dimension than w'. ■

It is easy to see that we can generalise the result to other probability distributions that are used to choose the bins.

Corollary 3.5. *If $w \succ w'$, and the probability that a ball is allocated to bin b_i, $1 \leq i \leq n$, is the same for all balls, then we have $E[S_i(w)] \geq E[S_i(w')]$ for all $i \in [n]$.*

4. Weighted multiple-choice games

In the first sub-section we show that for multiple-choice games it is not always better to allocate uniform balls. For $m \gg n$ we construct a set of weighted balls that ends up with a smaller expected maximum load than a set of uniform balls with the same total weight. The second sub-section considers the case where m is not much larger than n. As we will argue in the beginning of that section, it appears that it may not be possible to use the majorisation technique to get tight results for the weighted multiple-choice game. This is due to the fact that the order in which weighted balls are allocated is crucial, but the majorisation order is not necessarily preserved for weighted balls in the multiple-choice game (in contrast to [1] for uniform balls). We discuss several open questions and give some weight vectors that result in a smaller expected maximum load than uniform vectors with the same total weight.

4.1. Large number of balls

We compare two allocations, $\mathcal{A}$ and $\mathcal{B}$, respectively. In $\mathcal{A}$ we allocate $m/2$ balls of weight 3 each and thereafter $m/2$ balls of weight 1 each, using the multiple-choice strategy. Allocation $\mathcal{B}$ is the uniform counterpart of $\mathcal{A}$ where all balls have weight 2. We show that the expected maximum load in $\mathcal{A}$ is strictly smaller than that in $\mathcal{B}$. We will use the *short term memory property* stated below in Lemma 4.1. See [2] for a proof. Basically, this property says that after allocating a sufficiently large number of balls, the load depends on the last poly(n) many balls only. If m is now chosen large enough (but polynomially large in n suffices), then the maximum load is (w.h.p.) upper bounded by $2m/n + \log\log n$. In the case of balls with weight 2, the maximum load is w.h.p. upper bounded by $2m/n + 2\log\log n$. Since [2] gives only upper bounds on the load, we can not use the result directly. We introduce two auxiliary allocations named $\mathcal{C}$ and $\mathcal{D}$, respectively. Allocation $\mathcal{C}$ is derived from Allocation $\mathcal{A}$, and $\mathcal{D}$ is derived from $\mathcal{B}$. The only difference is that in allocations $\mathcal{C}$ and $\mathcal{D}$ we allocate the first $m/2$ balls optimally (i.e. we always place the balls into the least loaded bins). In Lemma 4.5 we first show that the expected maximum loads of $\mathcal{A}$ and $\mathcal{C}$ will be nearly indistinguishable after allocating all the balls. Similarly, the maximum loads of $\mathcal{B}$ and $\mathcal{D}$ will be nearly indistinguishable. Moreover, we show that the expected maximum load in $\mathcal{D}$ is larger than that in $\mathcal{C}$. Then we can show that the expected maximum load in $\mathcal{A}$ is smaller than that in $\mathcal{B}$ (Theorem 4.6). For an overview of the four systems, we refer to Table 1.

To state the short memory property we need one more definition. For any two random variables X and Y defined jointly on the same sample space, the *variation distance* between $\mathcal{L}(X)$ (the "law", or distribution, of X) and $\mathcal{L}(Y)$ is defined as

$$\|\mathcal{L}(X) - \mathcal{L}(Y)\| = \sup_{A} |\Pr(X \in A) - \Pr(Y \in A)|,$$

where A is an arbitrary subset of the events. The following lemma is from [2, Corollary 1].

Lemma 4.1. *Suppose $L_0 = (\ell_1, \ldots, \ell_n)$ is an arbitrary normalised load vector describing an allocation of m balls into n bins. Define $\Delta = \ell_1 - \ell_n$ to be the maximum load difference in L_0. Let L'_0 be the load vector describing the optimal allocation of the same number of balls to n bins. Let L_k and L'_k, respectively, denote the vectors obtained after inserting k further balls to both allocations using the multiple-choice algorithm, Then for $k \geq n^5 \cdot \Delta$*

$$\|\mathcal{L}(L_k) - \mathcal{L}(L'_k)\| \leq k^{-\alpha}$$

where α is an arbitrary constant.

Intuitively, Lemma 4.1 indicates that given any configuration with maximum difference Δ, in $\Delta \cdot \text{poly}(n)$ steps the allocation "forgets" the difference, i.e., the allocation is nearly indistinguishable from the allocation obtained by starting from a completely balanced allocation. This is in contrast to the single-choice game requiring $\Delta^2 \cdot \text{poly}(n)$ steps in order to "forget" a load difference Δ (see [2]).

Lemma 4.2. *Suppose we allocate m balls to n bins using Greedy[d] with $d \geq 2$, $m \gg n$. Then the number of bins with load at least $m/n + i + \gamma$ is bounded above by $n \cdot \exp(-d^i)$, w.h.p, where γ denotes a suitable constant. In particular, the maximum load is w.h.p.*

$$\frac{m}{n} + \frac{\log\log n}{\log d} \pm \Theta(1).$$

Proof. The result that the maximum load is at most $m/n + \log\log n/\log d + \Theta(1)$ has been shown in [2]. To show the lower bound we first recall two results shown in [1]. First, let u and v be two positive integer vectors such that $u_1 \geq u_2 \geq \ldots \geq u_n$ and $v_1 \geq v_2 \geq \cdots \geq v_n$. Azar et. al show that if $u \succ v$, then also $u + e_i \succ v + e_i$, where e_i is the ith unit vector. Now let u, v be two vectors with same total weight. Denote by u' and v' the load vectors obtained by allocating a unit-size ball b into two allocations having initial loads u, v respectively. Then Azar et al. show the following theorem:

Theorem 4.3. *If $u \succ v$, there is a coupling of two allocations with respect to the allocation of b such that $u' \succ v'$.*

We consider two allocations $\mathcal{E}$ and $\mathcal{F}$. In $\mathcal{E}$ we allocate m balls into n bins using Greedy[d], while in $\mathcal{F}$, we first place $m - n$ balls optimally, and then allocate the remaining n balls by Greedy[d]. Clearly after allocating the first $m - n$ balls, the normalised load vector of $\mathcal{E}$ always majorises the normalised load vector of $\mathcal{F}$. Applying Theorem 4.3 on the last n balls, we see that $\mathcal{E} \succ \mathcal{F}$. Since the maximum load in $\mathcal{F}$ is known to be lower bounded by $m/n + \log\log n/\log d - \Theta(1)$ w.h.p [1], the same lower bound holds for the maximum load of $\mathcal{E}$. ∎

Let $L_i(\mathcal{A})$ (or $L_i(\mathcal{B}), L_i(\mathcal{C}), L_i(\mathcal{D})$) be the maximum load in Allocation $\mathcal{A}$ (respectively, $\mathcal{B}, \mathcal{C}, \mathcal{D}$) after the allocation of the first i balls. If we refer to the maximum load after the allocation of all m balls we will simply write $L(\mathcal{A})$ (or $L(\mathcal{B})$, $L(\mathcal{C})$, $L(\mathcal{D})$). Lemma 4.5 below compares the load of the four allocations described in Table 1. First, we give a lemma stating that, given two random variables, a small variation distance implies a small difference between their expectations.

Lemma 4.4. *Let X and Y be two discrete random variables sharing the same sample space. Let ζ be the maximum possible value of X and Y. Then,*

$$|E[X] - E[Y]| \leq \zeta \cdot \|\mathcal{L}(X) - \mathcal{L}(Y)\|.$$

Proof. Let $G = \{k |\Pr(X = k) > \Pr(Y = k)\}$, $S = \{k |\Pr(X = k) < \Pr(Y = k)\}$. Due to choice of G and S,

$$\sum_{k\in G} (\Pr(X = k) - \Pr(Y = k)) = \sum_{k\in S} (\Pr(Y = k) - \Pr(X = k)).$$

Hence,

$$\begin{aligned}
&|E[X] - E[Y]| \\
&= \left| \sum_{k} ((\Pr(X = k) - \Pr(Y = k)) \cdot k) \right| \\
&= \left| \sum_{k\in G} ((\Pr(X = k) - \Pr(Y = k)) \cdot k) - \sum_{k\in S} ((\Pr(Y = k) - \Pr(X = k)) \cdot k) \right| \\
&\leq \max \left\{ \sum_{k\in G} ((\Pr(X = k) - \Pr(Y = k)) \cdot k),\ \sum_{k\in S} ((\Pr(Y = k) - \Pr(X = k)) \cdot k) \right\} \\
&\leq \max \left\{ \zeta \cdot \sum_{k\in G} (\Pr(X = k) - \Pr(Y = k)),\ \zeta \cdot \sum_{k\in S} (\Pr(Y = k) - \Pr(X = k)) \right\} \\
&= \zeta \cdot \sum_{k\in G} (\Pr(X = k) - \Pr(Y = k)) \ \leq\ \zeta \cdot \sup_{A} |\Pr(X \in A) - \Pr(Y \in A)| \ =\ \zeta \cdot \|\mathcal{L}(X) - \mathcal{L}(Y)\|. \quad ∎
\end{aligned}$$

Lemma 4.5. *Let* $m = \Omega(n^6)$.

(a) $E[L(\mathcal{D})] - E[L(\mathcal{C})] \geq \frac{\log\log n}{\log d} - \Theta(1)$.
(b) $|E[L(\mathcal{A})] - E[L(\mathcal{C})]| \leq m^{-\beta}$, *where* β *is an arbitrary constant.*
(c) $|E[L(\mathcal{B})] - E[L(\mathcal{D})]| \leq m^{-\beta'}$, *where* β' *is an arbitrary constant.*

Proof. Part (a). The deviation of the maximum load from the average in Allocation $\mathcal{D}$ is exactly twice that of $\mathcal{C}$, or

$$E[L(\mathcal{D})] - \frac{2m}{n} = 2 \cdot \left(E[L(\mathcal{C})] - \frac{2m}{n}\right).$$

Hence,

$$E[L(\mathcal{D})] - E[L(\mathcal{C})] = E[L(\mathcal{C})] - \frac{2m}{n}.$$

By Lemma 4.2, the maximum load of Allocation $\mathcal{C}$ is at least $\frac{2m}{n} + \frac{\log\log n}{\log d} - \Theta(1)$ w.h.p. Hence,

$$E[L(\mathcal{D})] - E[L(\mathcal{C})] = E[L(\mathcal{C})] - \frac{2m}{n} \geq \frac{\log\log n}{\log d} - \Theta(1).$$

Part (b). For $0 \leq i \leq m/(2n)$, we define Allocation $\mathcal{A}_i$ as follows. In $\mathcal{A}_i$, we allocate the first $i \cdot n$ balls optimally and the rest $m - i \cdot n$ balls by Greedy[d] (clearly $\mathcal{A}_0 = \mathcal{A}$ and $\mathcal{A}_{m/(2n)} = \mathcal{C}$). Note that for $1 \leq i \leq m/(2n)$, the maximum load difference between $\mathcal{A}_i$ and $\mathcal{A}_{i-1}$ is $3n$. Since $m/2 = \Omega(n^6) > n^5 \cdot n$, by Lemma 4.1, the Greedy[d] algorithm has "short memory". In other words, after allocating all the m balls of A_{i-1} and A_i, both allocations will become almost indistinguishable. Moreover, we note that the variation distance of two random vectors is certainly no bigger than that of their respective maxima, thus

$$\|\mathcal{L}(L(\mathcal{A}_{i-1})) - \mathcal{L}(L(\mathcal{A}_i))\| \leq \left(\frac{m}{2}\right)^{-\alpha},$$

for an arbitrary constant α. Consequently

$$\|\mathcal{L}(L(\mathcal{A})) - \mathcal{L}(L(\mathcal{C}))\| \leq \sum_{i=1}^{m/2n} \|\mathcal{L}(L(\mathcal{A}_{i-1})) - \mathcal{L}(L(\mathcal{A}_i))\| \leq \left(\frac{m}{2}\right)^{-\alpha} \cdot \left(\frac{m}{2n}\right) \leq m^{-(\alpha-1)}.$$

It is clear that the maximal possible loads of both allocations $\mathcal{A}$ and $\mathcal{C}$ are $2m$ (if we allocate all the balls into one bin). By Lemma 4.4,

$$|E[L(\mathcal{A})] - E[L(\mathcal{C})]| \leq \left(\frac{m}{2}\right)^{-(\alpha-1)} \cdot 2m \leq m^{-\beta}$$

as long as we choose $\alpha = \frac{(1+\beta)\log_2 m + 1}{\log_2 m - 1} + 1$.

Part (c). This can be shown similar to part (b). ∎

Finally, we present the main result of this section, showing that uniform balls do not necessarily minimize the expected maximum load in the multiple-choice game.

Theorem 4.6. $E[L(\mathcal{B})] \geq E[L(\mathcal{A})] + \frac{\log\log n}{\log d} - \Theta(1)$.

Proof. Of course,

$$E[L(\mathcal{B})] - E[L(\mathcal{A})] = (E[L(\mathcal{D})] - E[L(\mathcal{C})]) - (E[L(\mathcal{A})] - E[L(\mathcal{C})]) + (E[L(\mathcal{B})] - E[L(\mathcal{D})]).$$

Since the difference between $(E[L(\mathcal{A})] - E[L(\mathcal{C})])$ and $(E[L(\mathcal{B})] - E[L(\mathcal{D})])$ is at most $m^{-\beta}$ (Lemma 4.5), we conclude that

$$E[L(\mathcal{B})] - E[L(\mathcal{A})] \geq \frac{\log\log n}{\log d} - \Theta(1) - m^{-\beta} - m^{-\beta'} \geq \frac{\log\log n}{\log d} - \Theta(1).$$ ∎

4.2. Majorisation Order for arbitrary values of m

In this section, we consider the Greedy[2] process applied on weighted balls, but most of the results can be generalised to the Greedy[d] process for $d > 2$. Just to remind you, in the Greedy[2] process each ball sequentially picks i.u.r. two bins and the current ball is allocated in the least loaded of the two bins (ties can be broken arbitrarily). This means, of course, that a bin with relative low load is more likely to get an additional ball than one of the highly loaded bins.

Another way to model the Greedy[d] process is the following: Assume that the load vector of the bins are normalised, i.e. $\ell_1 \geq \ell_2 \geq \cdots \geq \ell_n$. If we now place an additional ball into the bins, the ball will be allocated to bin i with probability

$(i^d - (i-1)^d)/n^d$, since all d choices have to be among the first i bins, and at least one choice has to be i. For $d = 2$ this simplifies to $(2i-1)/n^2$. Hence, in this fashion, the process can be viewed as a "one choice process", provided the load vector is re-normalised after the allocation of each ball. This means that the load distribution of the bins highly depends on the order in which the balls are allocated.

Unfortunately, the dependence of the final load distribution on the order in which the balls are allocated makes it very hard to get tight bounds using the majorisation technique together with T-transformations. Theorem 3.1 highly depends on the fact that we can assume that w_j and w_k (y_j and y_k) are allocated at the very end of the process, an assumption that can not be used in the multiple-choice game. In order to use T-transformations for multiple choice games, we would again need a result that shows that the majorisation order is preserved when we add more (similar) balls into the allocation. We need a result showing that if $\mathcal{A} \succ \mathcal{B}$ and we add an additional ball to both $\mathcal{A}$ and $\mathcal{B}$, after the allocation we still have $\mathcal{A}' \succ \mathcal{B}'$ (where $\mathcal{A}'$ and $\mathcal{B}'$ denote the new allocations with the one additional ball). While this is true for uniform balls (see [1]), this is not necessarily true for weighted balls and the multiple choice game. In the following sections we study the majorisation order for weighted multiple choice games, and the effect that the the allocation order or the number of balls have on the final load distribution.

Majorisation Order. The following easy example shows that the majorisation order need not be preserved for weighted balls in the nultiple-choice case. Let $\mathcal{A} = (7, 6, 5)$ and $\mathcal{B} = (7, 5.8, 5.2)$. If we now allocate one more ball with weight $w = 2$ into both systems (using the Greedy[2] protocol), with probability 5/9 the ball is allocated to the third bin in both allocations and we have $\mathcal{A}' = (7, 7, 6)$ and $\mathcal{B}' = (7.2, 7, 5.8)$, hence $\mathcal{B}' \succ \mathcal{A}'$. Alternatively, with probability 1/3 the ball is allocated to the second bin in each allocation resulting in load vectors $\mathcal{A}' = (8, 7, 5)$ and $\mathcal{B}' = (7.8, 7, 5.2)$. Finally, with probability 1/9 the ball is allocated to the first bin resulting in load vectors $\mathcal{A}' = (9, 6, 5)$ and $\mathcal{B}' = (9, 5.8, 5.2)$. In both cases we still have $\mathcal{A}' \succ \mathcal{B}'$. This shows that after the allocation of one additional ball using Greedy[2], the majorisation relation can turn around. Note that the load distributions of $\mathcal{A}$ and $\mathcal{B}$ are not "atypical", but they can easily come up using Greedy[2].

The next lemma gives another example showing that the majorisation relation need *not* be preserved for weighted balls in the multiple-choice game. The idea is that we can consider two allocations $\mathcal{C}$ and $\mathcal{D}$ where $\mathcal{C} \succ \mathcal{D}$, but by adding one additional ball (with large weight w), we then have $E[S_1(\mathcal{D}')] \geq E[S_1(\mathcal{C}')]$. It is easy to generalise the lemma to cases where w is not larger than the maximum bin load to show that the majorisation relation need not be preserved.

Lemma 4.7. *Let v and u be two (normalised) load vectors with $v \overset{T}{\Longrightarrow} u$ (so $v \succ u$). Let w be the weight of an additional ball with $w > v_1$. Let v', u' be the new (normalised) load vectors after allocating the additional ball into v and u. Then we have $E[S_1(u')] > E[S_1(v')]$.*

Proof. First we assume $v \overset{T}{\Longrightarrow} u$. Then, by the property of T-transformations, there must exist two bins with rank $j, k \in \mathbb{Z}^+$, $j < k$, such that $v_j > u_j > u_k > v_k$, and for $\forall i \neq j, k$, that $u_i = v_i$. Besides, we have $v_j - u_j = u_k - v_k > 0$. We observe that, since $w > v_1 \geq u_1$, the destination of the new ball immediately becomes the maximum loaded bin in both allocations. Since the probability to place the new ball on top of the i-th largest bin in both allocations is $\frac{i^d-(i-1)^d}{n^d}$ we get

$$
\begin{aligned}
E[S_1(u')] - E[S_1(v')] &= \sum_{i=1}^{n} \frac{i^d-(i-1)^d}{n^d} \cdot (u_i - v_i) \\
&= \frac{j^d-(j-1)^d}{n^d} \cdot (u_j - v_j) + \frac{k^d-(k-1)^d}{n^d} \cdot (u_k - v_k) \qquad (1) \\
&> 0. \qquad (2)
\end{aligned}
$$

Here (1) holds since $\forall i \notin \{j, k\}$, $u_i = v_i$. Inequality (2) is due to the facts that $j < k$ and $v_j - u_j = u_k - v_k > 0$. ∎

Remark: We feel it necessary to point out that the preceding lemma applies only to the largest elements of u' and v'. It is possible that after the allocation of the new ball we could have $E[S_2(u')] > E[S_2(v')]$ or the reverse inequality $E[S_2(v')] > E[S_2(u')]$.

For example, (using the Greedy[2] protocol) take $v = (7, 7, 3)$, $u = (7, 5, 5)$, and $w = 20$, and the first inequality holds. It is easy to check that $E[S_2(u')] = 32 > 31\frac{7}{9} = E[S_2(v')]$. On the other hand, using the vectors $v = (100, 1, 1)$, $u = (35, 34, 33)$, and a new ball having weight $w = 101$ we find that $E[S_2(v')] = 202 > 169\frac{4}{9} = E[S_2(u')]$. However, Lemma 4.7 tells us that $E[S_1(u')] \geq E[(S_1(v')]$ holds in both cases.

Lemma 4.7 and the example preceding that lemma both showed that a more unbalanced weight vector can end up with a smaller expected maximum load after the allocation of some additional (and similar) balls. However, in those cases we assumed that the number of bins is very small, or that one of the balls is very big. Simulation results show that for most weight vectors w, w' with $w \succ w'$ the expected maximum load after the allocation of w' is smaller than the one after the allocation of w. Unfortunately, we have been unable to show a result along these lines formally.

Order of the balls. Another interesting question concerns the order of allocating balls under the multiple-choice scenario. In the case that $m \geq n$, we conjecture that if all the balls are allocated in decreasing order, the expected maximum is the smallest among all possible permutations. This is more or less intuitive since if we always allocate bigger balls first, the

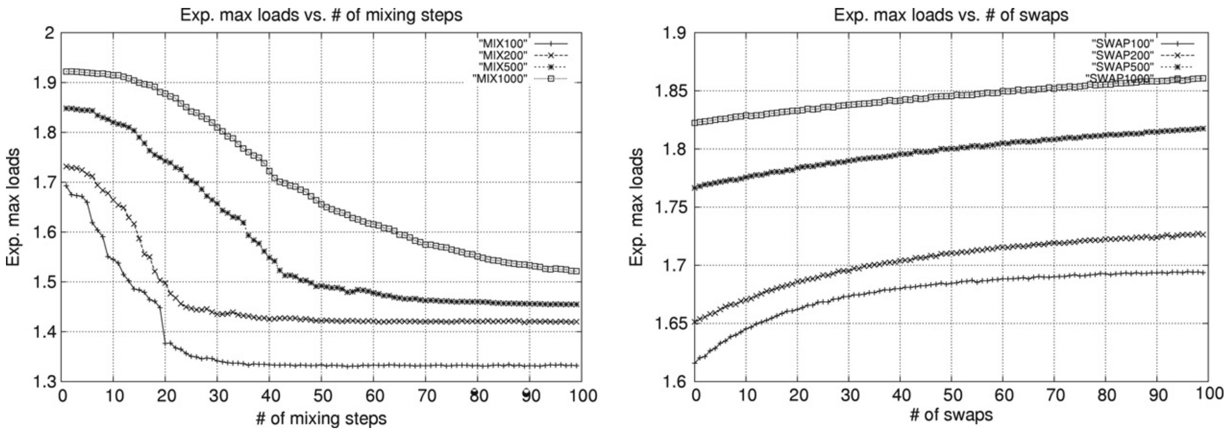

Fig. 1. Left: successive equalisation of weights. Right: successive swapping of weights.

chances would be low to place the remaining balls in those bins which are already occupied by the bigger balls. However, we still do not know how to prove this conjecture. We can answer the peer question: what about if we allocate balls in increasing order? The next observation shows that the increasing order does *not* always produce the worst outcome.

Observation 4.8. *Fix a set of weighted balls. The expected maximum load is not necessarily maximised by allocating the balls in increasing order.*

Proof. We compare two allocations $\mathcal{A}$ and $\mathcal{B}$ both with n bins. Let $w_{\mathcal{A}} = \{1, 2, 1, 5\}$, and $w_{\mathcal{B}} = \{1, 1, 2, 5\}$ be two weight vectors (sequences of ball weights). Notice that $w_{\mathcal{B}}$ is a monotonically increasing sequence while $w_{\mathcal{A}}$ is not. After allocating the first three balls, observe that the possible outcomes for $\mathcal{A}$ and $\mathcal{B}$ are $(2, 1, 1, 0, \dots 0)$, $(3, 1, 0 \dots, 0)$, $(2, 2, 0 \dots, 0)$ and $(4, 0, \dots 0)$. We can calculate the probabilities for $\mathcal{A}$ and $\mathcal{B}$ to end up in outcome $(2, 2, 0 \dots, 0)$ are $(1 - 1/n^2) \cdot 3/n^2$ and $(1 - 1/n^2) \cdot 1/n^2$, respectively. Moreover, notice both $\mathcal{A}$ and $\mathcal{B}$ have the same probability to end up in outcome $(2, 1, 1, 0, \dots 0)$ and $(4, 0, \dots, 0)$. Consequently, $\mathcal{B}$ has more (in fact, $(1 - 1/n^2) \cdot 2/n^2$) probability to end up in outcome $(3, 1, 0, \dots, 0)$ than $\mathcal{A}$, while $\mathcal{A}$ is more likely to end up in outcome $(2, 2, 0, \dots, 0)$. Hence, after allocating the first three balls, $\mathcal{B}$ certainly majorises $\mathcal{A}$. Since the last ball (with weight 5) is bigger than the loads of all bins in both $\mathcal{A}$ and $\mathcal{B}$ after allocating the first three balls, by Lemma 4.7 the expected maximum load after allocating $w_{\mathcal{A}}$ is bigger than that after allocating $w_{\mathcal{B}}$. ■

Many small balls. Another natural question to ask is the one we answered in Corollary 3.4 for the single-choice game. Is it better to allocate a large number of small balls compared to a smaller number of large balls with the same total weight? The next example shows again that the majorisation relation is not always maintained in this case.

Observation 4.9. *We consider two systems $\mathcal{A}$ and $\mathcal{B}$ both of n bins. Let $W_{\mathcal{A}} = (0, 2, 4, \dots, 2^{m-1})$ and $W_{\mathcal{B}} = (1, 1, 4, \dots, 2^{m-1})$ denote two allocations both of $m \geq 3$ balls. Note both systems are of same total weight and $W_{\mathcal{A}} \succ W_{\mathcal{B}}$, but if m is odd, the expected maximum load of $\mathcal{A}$ is smaller than $\mathcal{B}$.*

Proof. Clearly after allocating the first two balls System $\mathcal{A}$ majorises System $\mathcal{B}$. Besides, note that for both systems, the weight of every newly allocated ball is bigger than the sum of weights of all the balls allocated before. Hence, by Lemma 4.7, every time when a new ball is allocated, the majorisation relation would be "reversed". Hence, for any odd number $m \geq 3$, System $\mathcal{B}$ certainly majorises System $\mathcal{A}$. ■

To see this, when $m = 3$, simply by enumerating all cases we can get, the expected maximum load of $\mathcal{A}$ is $4 + 2/n^2$, which is smaller than that of $\mathcal{B}$ $(4 + 4/n^2 - 2/n^4)$.

We emphasize again that the initial majorisation relation is no longer preserved during the allocation. However, we still conjecture that in "most" cases the allocation of a large number of small balls is majorised by the one of a smaller number of large balls with the same total weight, but so far we have been unable to show formal results. The next section contains empirical results obtained by computer simulations examining some of the issue we have raised earlier.

4.3. *Simulation results*

In this section, we conduct an empirical study for the weighted multiple-choice balls-into-bins game. We allocate $m = n$ balls into n bins while the number of choices, d, is chosen to be 2. We examine cases in which n is set to 100, 200, 500, and 1000, respectively. Note it is not feasible to enumerate the huge number of possible allocations (which is $n^{m \cdot d}$) to calculate the exact expected maximum loads. Instead, we approximate them by taking the average maximum loads for a large number (specifically 100,000) number of iterations.

The goal of the first experiment is to demonstrate the following observation: the more balanced the ball weights are, the less the expected maximum load will be, after allocating all balls. In our experiment, we first randomly assign a weight in $(0, 1)$ to each ball. After that, we perform a few "mixing" steps, in which we choose two balls at random and equalise their weights, to make the overall weight vectors more balanced. We record the corresponding expected maximum loads vs. the number of mixing steps in the left part of Fig. 1.

Although the first observation above is almost always true, we still note that there do exist ball weight distributions which achieve smaller expected maximum load than their corresponding uniform ones, as shown in Theorem 4.6.

Next, we perform an experiment regarding the order of placing balls. We aim at showing that if we allocate balls in decreasing order of their weights, we would get the least expected maximum load. This seems intuitively likely since if we allocate big balls first, the small balls later are likely to fall into the holes left by the big ones. For the experiment, we first randomly assign each ball a weight in $(0, 1)$ and sort all ball weights by non-increasing order. Later, we perform a number of "swaps", i.e., we randomly choose two balls and exchange their weights, to get different ball arrangements. The right part of Fig. 1 shows the relation between the number of swaps and the corresponding expected maximum loads.

Clearly our experiment appears to support the conjecture that the decreasing order achieves the minimum expected maximum load. Unfortunately, we have not yet succeeded in proving this conjecture.

References

[1] Yossi Azar, Andrei Z. Broder, Anna R. Karlin, Eli Upfal, Balanced allocations, SIAM Journal on Computing 29 (1) (1999) 180–200.
[2] Petra Berenbrink, Artur Czumaj, Angelika Steger, Berthold Vöcking, Balanced allocations: The heavily loaded case, SIAM Journal on Computing 35 (6) (2006) 1350–1385.
[3] Petra Berenbrink, Tom Friedetzky, Zengjian Hu, Russell A. Martin, On weighted balls-into-bins games, in: Proc. of the 22nd Symposium on Theoretical Aspects of Computer Science, STACS, 2005, pp. 231-243.
[4] Petra Berenbrink, Randomized allocation of independent tasks, University of Paderborn, 2000.
[5] Petra Berenbrink, Friedhelm Meyer auf der Heide, Klaus Schröder, Allocating weighted jobs in parallel, Theory of Computing Systems 32 (3) (1999) 281–300.
[6] Kunal Talwar, Udi Wieder, Balanced allocations: The weighted case, in: Proc. of the 39th Annual ACM Symposium on Theory of Computing, STOC, 2007, pp. 256–265.
[7] Elias Koutsoupias, Marios Mavronicolas, Paul G. Spirakis, Approximate equilibria and ball fusion, Theory of Computing Systems 36 (6) (2003) 683–693.
[8] Albert W. Marshall, Ingram Olkin, Inequalities: Theory of Majorization and its Applications, Academic Press, 1979.
[9] Michael Mitzenmacher, Andrea W. Richa, Ramesh Sitaraman, The power of two random choices: A survey of techniques and results, in: Handbook of Randomized Computing, 2000.
[10] Michael Mitzenmacher, Balaji Prabhakar, Devarat Shah, Load balancing with memory, in: Proc. of the 43rd Annual IEEE Symposium on Foundations of Computer Science, FOCS, 2002, pp. 799–808.
[11] Peter Sanders, On the competitive analysis of randomized static load balancing, in: Proceedings of the first Workshop on Randomized Parallel Algorithms, RANDOM, 1996.
[12] Berthold Vöcking, How asymmetry helps load balancing, Journal of the ACM 50 (4) (2003) 568–589.