IBM Research Report

Perfectly Balanced Allocation With Estimated Average Using Approximately Constant Retries

Sourav Dutta
IBM Research Division
IBM India Research Lab
Plot - 4, Block - C, ISID Campus, Vasant Kunj
New Delhi - 110070, India.

Souvik Bhattacherjee
IBM Research Division
IBM India Research Lab
Plot - 4, Block - C, ISID Campus, Vasant Kunj
New Delhi - 110070, India.

Ankur Narang
IBM Research Division
IBM India Research Lab
Plot - 4, Block - C, ISID Campus, Vasant Kunj
New Delhi - 110070, India.

IBM Research Division
Almaden - Austin - Beijing - Delhi - Haifa - T.J. Watson - Tokyo - Zurich

LIMITED DISTRIBUTION NOTICE: This report has been submitted for publication outside of IBM and will probably be copyrighted is accepted for publication. It has been issued as a Research Report for the early dissemination of its contents. In view of the transfer of copyright to the outside publisher, its distribution outside of IBM prior to publication should be limited to peer communications and specific requests. After outside publication, requests should be filled only by reprints or legally obtained copies of the article (e.g., payment of royalties). Copies may be requested from IBM T.J. Watson Research Center, Publications, P.O. Box 218, Yorktown Heights, NY 10589 USA (email: reports@us.ibm.com). Some reports are available on the internet at http://domino.watson.ibm.com/library/CyberDig.nsf/home.

Abstract

Balanced allocation of online balls-and-bins has long been an active area of research for efficient load balancing and hashing applications. There exists a large number of results in this domain with different settings, such as parallel allocations [1], multi-dimensional allocations [5], weighted balls [4] etc. For sequential multi-choice allocation where m balls are thrown into n bins with each ball choosing d (constant) bins independently uniformly at random, the maximum load of a bin is $O(\log\log n) + m/n$ with high probability [3]. This offers the current best known allocation scheme. However, for $d = \Theta(\log n)$, the gap obtained is of $O(1)$ [9]. A similar constant gap bound has been established for parallel allocations with $O(\log^* n)$ communication rounds [12].

In this paper we propose a novel multi-choice allocation algorithm, *Improved D-choice with Estimated Average* (IDEA) achieving a constant gap with high probability for the sequential single-dimensional online allocation problem with constant d. We achieve a maximum load of $\lceil m/n \rceil$ with high probability for constant d choice scheme with *approximately* constant number of retries or rounds per ball. We also show that the bound holds even for an arbitrary large number of balls, $m >> n$. Further, we generalize this result to the weighted case, where the balls have weights drawn from an arbitrary weight distribution with finite variance. We show that the gap in this case is also independent of the number of balls thrown and is a constant. We also propose that $IDEA$ provides constant gap bound w.h.p. in the multi-dimensional scenario for $m = n$.

1 Introduction

A central research area in the domain of randomized algorithms is the *occupancy problem* for *balls-into-bins* processes [2, 7, 3, 12, 13]. The framework of the problem involves the analysis of the online allocation, wherein a set of independent balls is to be assigned to a set of bins. The occupancy problem helps to model several realistic problems into a formal mathematical structure, and hence opens an active area of work in probability theory as well as in computer science.

In the classical "balls-into-bins" problem, m balls are sequentially thrown into n bins, where each ball is placed into one of the bins independently and uniformly at random (i.u.r.). The natural question then is to analyze the maximum load in any of the bins. Mapping the problem to the application domain, we may consider the balls to be jobs or tasks and the bins to be servers. The problem then reduces to scheduling the jobs with balanced load allocations among the servers.

Probably one of the earliest applications of randomized load balancing is in the context of $hashing$. For the chaining method during hash clash, the length of the lists in the hash buckets are a measure of the retrieval complexity. For a uniform hash function, the length of the lists follow the same distribution as the number of balls in a bin in this case.

The advent of parallel and distributed systems required efficient online load balancing among the servers to improve the throughput of the system. Dependence on a centralized environment for uniform load balancing is highly undesirable for such systems due to high communication complexity. With the introduction of the *Cloud Computing* paradigm, the placement of virtual machines (VMs) on servers provided a new dimension to the applicability of the randomized balanced allocation study.

Other applications such as the design of Multimedia or Data Servers use disk arrays where a data unit is partitioned and stored in a distributed fashion. These applications demand even (balanced) access of the disks on retrieval [15] and Karp in [11] discusses applications in video-on-demand (termed *k-orientability* [7]). The balls into bins problem accurately describes these applications only when the balls have uniform weights. Other applications assume the loads to be of different weights to model its various dimensions.

This paper tackles the problem of sequential online allocation of balls into bins. Assuming we have n bins and m balls arriving one at a time are to be thrown into these bins, the problem is to devise an efficient algorithm such that the allocation of the balls is nearly balanced among all the bins. In formal terms, the load in each of the bins should be as close to the average, (m/n) as possible. We initially study the case of single-dimensional sequential placement of uniform weighted balls into bins problem and then extend it for the general weighted case. Finally we also observe that $IDEA$ provides the same result w.h.p. for multi-dimensional balls-into-bins problem for $m = n$.

In this context, we define *Gap* to be the difference between the heaviest loaded bin and the average load. The currently best known algorithm bounds Gap to $O(\log \log n)$ with high probability using the symmetric *d-choice* placement strategy [2, 13]. In the d-choice method, each ball selects d bins i.u.r. among the n bins and is allocated to the least loaded bin among them. It is well-known that if $d = \Theta(\log n)$ choice, the gap is $O(1)$ [9].

In this paper we propose a novel algorithm, *Improved D-choice with Estimated Average*, ($IDEA$) for efficient placement of the balls in the bins. We prove that this technique provides a ***constant*** Gap with high probability (w.h.p.) even when d is kept constant, albeit with an approximately constant number of retries or rounds per ball. We further extend the result to show that the guarantee also holds true for the heavily loaded case, i.e. $m >> n$ w.h.p. Our technique is different from the typical greedy d-choice process in that it places the ball in the bin that has load equal-to or lower than the *estimated average* of that bin. Using *approximately* constant number of retries such a bin can be found for each ball and hence the load in each bin tends towards the estimated average which also tends towards the actual average, resulting in constant upper bound on the gap. Our strategy is also different from the typical asymmetric strategy [17] where in case of tie over the load, the leftmost bin gets the ball. Our result can have profound implication both theoretically and practically on the online load balancing algorithms.

The outline of the paper is as follows: Section 2 presents an introduction to the known works and results in this domain. In Section 3 we propose the detailed outline of the $IDEA$ algorithm for allocating the balls into the bins. Section 4 provides the theoretical proof for bounding the Gap to a constant quantity with high probability. Section 5 provides insights into the working of the $IDEA$ algorithm. Section 6.1 depicts its extension for the general weighted balls case, and Section 6.2 exhibits similar results for the multi-dimensional scenario. Finally, Section 7 concludes the paper.

2 Related Work

The study of "balls-into-bins" problem dates back to the study of hashing by Gonnet. He showed that when n balls are thrown into n bins i.u.r., the fullest bin has an expected load of $(1 + o(1)) \log n / \log \log n$ [10]. The maximum loaded bin in this approach was shown to be $O(\log n / \log \log n)$ w.h.p. [8]. It was also shown that for $m \geq n \log n$ balls, a bin can have a maximum load of $m/n + \Theta(\sqrt{m \log n / n})$.

Azar et al. [2] showed that if the balls chose sequentially from $d \geq 2$ bins i.u.r. (called *Greedy[d]* algorithm) and greedily selected the bin currently with the lowest load, the Gap could be bounded by $O(\log \log n / \log d)$ w.h.p. However, the solution worked only for the case when $m = n$. They also showed that the bound is stochastically optimal, i.e. any other greedy approach using the placement information of the previous balls to place the current ball majorizes to their approach. However, if the alternatives are drawn from separate groups with different rules for tie breaking, it results in different allocations. [17] presents such an *asymmetric* strategy and using witness tree based analysis proves that this leads to an improvement in the load balance to $O(\frac{\log \log(n)}{d \log(\phi_d)})$ w.h.p. where, ϕ_2 is the golden ratio and ϕ_d is a simple generalization. Our algorithm is different from both these techniques in that it uses the *estimated gap* as the criterion for choosing the bin and makes potentially multiple retries, where in each retry d bins are chosen i.u.r.

For the heavily loaded case, $m >> n$, the bound of $O(\log \log n / \log d)$ w.h.p. was later proven in [3] using sophisticated techniques in two main high level steps. In the first step, they show that when the number of balls is polynomially bounded by the number of bins the gap can be bounded by $O(\ln \ln(n))$, using the concept of layered induction and some additional tricks. In particular, they consider the entire distribution of the bins in the analysis (while in typical $m = O(n)$ case the bins with load smaller than the average could be ignored). In the second step, they extend this result to general $m >> n$ case, by showing that the multiple-choice processes are fundamentally different from the classical single-choice process in that they have *short memory*. This property states that given some initial configuration with gap Δ, after adding $poly(n)$ more balls the initial configuration is *forgotten*. The proof of the short memory property is done by analyzing the mixing time of the underlying Markov chain describing the load distribution of the bins. The study of the mixing time is via a new variant of the coupling method (called *neighboring coupling*). It was also shown that when $d = \Theta(\log n)$ the gap becomes $O(1)$ [9].

Cole et al. [6] showed that the two-choice paradigm can be applied effectively in a different context, namely, that of routing virtual circuits in interconnection networks with low congestion. They showed how to incorporate the two-choice approach to a well-studied paradigm due to Valiant for routing virtual circuits to achieve significantly lower congestion.

Kunal et.al. [16] prove that for weighted balls (weight distribution with finite fourth moment) and $m >> n$, the expected gap is independent of the number of balls and is less than n^c, where c depends on the weight distribution. They first prove the weak gap theorem which says that w.h.p $Gap(t) < t^{2/3}$. Since in the weighted case the d choice process is not dominated by the one choice process, they prove the weak gap theorem via a potential function argument. Then, the *short memory theorem* is proved. While in [3] the short memory theorem is proven via coupling, [16] uses similar coupling arguments but defines a different distance function and use a sophisticated argument to show that the coupling converges.

The $(1+\beta)$-choice scheme [14] proved that if a ball chooses with $\beta \in (0, 1)$ probability the least loaded bin of $d = 2$ randomly chosen bin, and otherwise i.u.r. a single bin, the Gap becomes independent of m and is given by $O(\log n/\beta)$.

In the parallel setting, [12] showed that a constant bound on the gap is possible with $O(\log^* n)$ communication rounds. Adler et.al. [1] consider parallel balls and bins with multiple rounds. They present analysis for $O(\frac{\log\log(n)}{\log(d)})$ bound on the gap (for $m = O(n)$) using $O(\frac{\log\log(n)}{\log(d)} + O(d))$ rounds of communication.

For offline balls-into-bins problem, using maximum flow computations it was shown that the maximum load of a bin w.h.p. is $\lceil m/n \rceil + 1$. [7] showed that for $m > cn \log n$ balls, where c is a sufficiently large constant, a perfect distribution of the balls was possible w.h.p. However, no such similar result is found in the literature for the online sequential case for constant d choice.

Mitzenmacher et. al. in [5] addresses both the single choice and d-choice paradigm for multidimensional balls and bins under the assumption that the balls are uniform D-dimensional (0, 1) vectors, where each ball has exactly f populated dimensions. They show that the gap for multidimensional balls and bins, using the two-choice process, is bounded by O(log log(nD)). We provide a better bound of $O(1)$ w.h.p. for $m = n$ case.

In this paper, we study a novel online sequential allocation algorithm for balls-into-bins based on a constant *d-choice* strategy and prove a constant gap bound both for $m = n$ and the heavily loaded case $m >> n$ along with for the general weighted balls and multi-dimensional scenario.

3 The $IDEA$ Algorithm

In this section we discuss the working of the *Improved D-choice with Estimated Average* ($IDEA$) algorithm. We consider there are n bins and m balls which arrive in an online fashion. We initially assume that the balls are of uniform weights and are numbered according to the order of their arrival. In hashing applications, the number of the balls based on their arrival order plays no role in assisting better or faster retrieval. Hence, this assumption does not decrease the complexity of the problem at hand. Later we also provide a blueprint of the case when such a numbering of the balls in not allowed and the weighted balls case with the weights of the balls drawn from an arbitrary distribution with finite variance.

```
Algorithm 1: IDEA Algorithm
  Require: Number of bins ($n$), Number of balls ($m$) and Maximum iteration ($\gamma$)
  Ensure: Balanced Allocation of Balls-into-Bins

    for all bin $B_i$, $i \in [1, n]$ do
      Initialize the load, $L_{B_i}$ and estimated average, $\hat{A}_{B_i}$ to 0
    end for
    for all ball $b_j$, $j \in [1, m]$ do
      $loop \leftarrow 0$
      while $loop \leq \gamma$ do
        Choose $d$ bins, $C = \{Bin_1, Bin_2, \cdots Bin_d\}$ i.u.r. from the $n$ bins
        if set $C$ contains at least one bin with negative or zero estimated gap, $\widehat{Gap}_{Bin_i} = L_{Bin_i} - \hat{A}_{Bin_i}$
        then
          Break while
        end if
        $loop \leftarrow loop + 1$
      end while
      Place ball $b_j$ in the bin, $B \in C$ having the lowest estimated gap, $\widehat{Gap}_B$
      $L_B \leftarrow L_B + 1$
      for all bins, $Bin_i \in C$ do
        if $\hat{A}_{Bin_i} > \lceil j/n \rceil$ then
          $flag \leftarrow 1$
        else
          $flag \leftarrow 0$
        end if
        if $flag = 0$ then
          $\hat{A}_{Bin_i} \leftarrow \hat{A}_{Bin_i} + 1/d$
        end if
      end for
    end for
```

Given each bin has an accurate knowledge of the average number of balls in the system, m/n it is easy to distribute the balls so as to obtain a perfectly balanced allocation. $IDEA$ operates on the above principle, where each bin independently calculates a fairly good estimate of the current average number of balls in the system. Each bin is then loaded nearly equal to its estimated average value. In the remainder of this section we show how each bin independently estimates its average which we later prove, with a high probability, to be very close to the actual average, m/n. We also show that each bin is then loaded close to its estimated average value, giving a maximum load of $\lceil m/n \rceil$ with a constant gap allocation w.h.p.

The $IDEA$ algorithm initially works as in the d-choice algorithm. On arrival of a ball b_j, it i.u.r. chooses d bins (d is constant) as its possible candidates for placement. Each bin, $B_i, i \in [1, n]$ is characterised by two parameters: (i) *Current Load*, L_i^j, and (ii) *Current Estimated Average*, $\hat{A}_i^j$. For each bin we define its *estimated gap*, $\widehat{Gap}_i^j$ as the difference between its current load and its current estimated average. Formally, $\widehat{Gap}_i^j = L_i^j - \hat{A}_i^j$.

The ball b_j is then allocated to the bin having the lowest value of $\hat{Gap}_i^j$ among the d chosen bins. Given the definition of Gap (in Section 1) we would like to place the ball in a bin with *negative* or *zero* $\hat{Gap}$. This would ensure that the loads in the bins be close to their estimated average values and thus lead to a lower Gap. Hence, if in the d choice a ball selects no bin with negative or zero $\hat{Gap}_i$, it re-chooses its candidate d bins. To boost the probability of a ball choosing a bin having such $\hat{Gap}_i$, this re-choosing will be carried out γ times, where γ will later be shown to be approximately a constant.

The current estimated average for each of the d bins finally selected by the ball is then incremented by $1/d$. In the next paragraph we discuss the selection of such an increment value. We intuitively argue that for each bin if $\hat{A}_i$ is finally close to the actual average (m/n) w.h.p., and its load L_i is nearly equal to its estimated average, the overall Gap in the system will be minimized and the maximum load of a bin will be $\lceil m/n \rceil$. The pseudo-code of $IDEA$ algorithm is shown in Algorithm 1.

The probability that a bin is chosen by a ball in its d choice is given by d/n. So when n balls arrive a bin will be chosen d times on expectation. For each such choice the estimated average of the bin is incremented by $1/d$ (Algorithm 1). Hence, its final estimated average will be 1, which is indeed the actual average of the system. However, from Lemma1 we observe that a bin might be chosen $d \log n$ times or lesser w.h.p. Since we increase the estimated average by $1/d$, the estimated average may increase beyond 1 in such cases. Hence the estimated average of a bin may be greater that 1 in two situations:
(i) Not more than n balls have arrived, but the bin has been chosen close to $d \log n$ times, or
(ii) More than n balls have arrived.
For case (i), the estimated average of the bin should still remain 1, while in the other case, the estimated average should be increased as usual. It is here that the numbering of the balls come into effect. If the estimated average of a bin goes beyond 1 and the next ball which selects this bin has a number less than n, the bin knows that it may be chosen $d \log n$ times and hence refrains from increasing its estimated average until a ball with number more than n selects it. Similarly when the estimated average of a bin increases beyond $\alpha, \alpha \in \mathbb{N}$, it checks if the next ball selecting it has a number greater than αn. Thus the balls communicate their numbers as well while choosing the d candidate bins.

However in the scenario where numbering of the balls is forbidden, to differentiate between the two cases, we use the sampling technique among the bins. A bin with estimated average just above α, in this case chooses $\log n$ bins i.u.r. and communicates with them for their estimated average. If the average of the estimated averages of the sampled bins are significantly lower than that of this bin (by some threshold, T), the bin comprehends that case (i) has happened, i.e., it is receiving more than d balls out of n balls and thus refrains from increasing its estimated average. However, if the average of the estimated averages are close to that of the communicating bin (difference is less than T), the bin decides that more than αn balls are arriving and increases its estimated value as usual. The probability that the error in the sampled average is greater than a constant ϵ is given by $\frac{1}{n}$ for $\log n$ sampled choice (*sampling theorem*). Hence w.h.p. of $1 - \frac{1}{n}$ we obtain the right decision for each bin.

Hence, we find that $IDEA$ dynamically adapts its estimated average to be closer to the actual average of the system. In either case, the estimated average of a bin is increased by at most 1 for every n balls.

4 Theoretical Framework

In this section, we provide a theoretical proof of the constant gap performance of the $IDEA$ algorithm. First, we bound the number of balls that may select each bin. We then establish that each ball in the $IDEA$ algorithm chooses at least one bin having negative $\hat{Gap}$ with a high probability, which makes the load of each bin converge to its estimated average value. Finally, we bound the Gap of the system to a constant value w.h.p. We assume m balls to arrive in an online fashion and there are n bins.

Lemma 1. *If each ball chooses d bins i.u.r. out of n bins, each bin is chosen by $\frac{md}{n}$ balls on expectation, and by at most $\frac{md}{n}\log n$ balls with high probability.*

Proof. Define $Y_1, Y_2, \cdots Y_m$ to be indicator random variables corresponding to balls $b_1, b_2, \cdots, b_m$ respectively. Let $Y_i = 1$ represent the event that the ball b_i chose bin B as one of its d candidate bins, otherwise $Y_i = 0$, $\forall i \in [1, m]$. Since the balls choose d bins i.u.r., the probability that bin B is chosen among the d bins, or $\Pr(Y_i) = 1$, is given by d/n. Let X be a random variable depicting the number of balls that chose B among its d candidate bins. Hence, $X = \sum_{i=1}^{m} Y_i$. The expected value of X is,

$$E[X] = E[\sum_{i=1}^{m} Y_i] = \sum_{i=1}^{m} E[Y_i] = \sum_{i=1}^{m} \frac{d}{n} = \frac{md}{n} \qquad \text{[By Linearity of Expectation]} \tag{1}$$

Applying Chernoff's bound on X we obtain,

$$P(X > (1+\delta)E[X]) < \frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}$$

$$\therefore P(X > (1+\delta)\frac{md}{n}) < \frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}$$

Substituting $\delta = \log n - 1$ we have,

$$P(X > \frac{md}{n}\log n) < \frac{e^{\log n - 1}}{(\log n)^{\log n}} = \frac{n}{e(\log n)^{\log n}} \tag{2}$$

Let $y = (\log n)^{\log n}$. Hence, $\log y = \log n \log\log n$. We have,

$\Rightarrow \log(y/n) = \log n\,(\log\log n - 1) = \log n\,(\log\log n - \log\log e^e)$

For large values of n, $\log\log(n/e^e) \geq 1$, giving $\log(y/n) \geq \log n$. Therefore, we have $y > n^2$.

Substituting in Eq. (2),

$$P(X > \frac{md}{n}\log n) < \frac{1}{en} \tag{3}$$

Hence, bin B is chosen by at most $\frac{md}{n}\log n$ balls with a high probability of $1 - \frac{1}{en}$. □

Lemma 2. *At any iteration, the estimated average of each bin is approximately equal to the current average with high probability.*

Proof. We assume here that Z balls have already arrived and have been placed among the n bins. The number of balls that chose bin B among its d candidates is $\frac{Zd}{n}$ on expectation, since each bin can be chosen by a ball with a probability of $\frac{d}{n}$. The number of such balls is also bounded by $\frac{Zd}{n}\log n$ with high probability (by Lemma 1). However, a bin does not increment its estimated average by more than d times for every n balls. For each choice the bin B increases its estimated average by $\frac{1}{d}$. Hence the current value of $\hat{A_B}$ is given by,

$$\hat{A_B} = \frac{Zd}{n} \cdot \frac{1}{d} = \frac{Z}{n} \text{ , which is the current average.}$$

Hence, the estimated average $\hat{A}$ of any bin is nearly equal to the actual average w.h.p. □

Observation 1. *The variance of the estimated average of a bin B for n balls is,*

$$Var[\hat{A_B}] = Var[\frac{X}{d}] = Var[\frac{1}{d}.\sum_{i=1}^{n} Y_i] = \frac{1}{d^2}\sum_{i=1}^{n} Var[Y_i]$$

$$= \frac{1}{d^2}.n\frac{d}{n}(1-\frac{d}{n}) = \frac{1}{d} - \frac{1}{n} \qquad \textit{[From Lemma 1]}$$

Lemma 3. *The sum of the estimated gap, $\hat{Gap}$ over all the bins at any time is zero.*

Proof. Each ball chooses d candidate bins uniformly and randomly and is finally allocated to the bin having the lowest estimated gap. Hence for all the d chosen bins, $IDEA$ increases their estimated average by $1/d$. The load of $d-1$ bins which do not receive the ball remains same, and thus their estimated gap decreases by the above factor. However, for the bin in which the ball is placed, its load increases by 1 and its estimated gap increases by $1 - 1/d$. Hence the sum of change of the estimated gap over the d chosen bins in any iteration is $1 - 1/d + (d-1)(-1/d) = 0$. □

Lemma 4. *The number of bins having a* zero *or* negative *estimated gap, $\hat{Gap}$ is $\Theta(n)$.*

Proof. Let there be α bins with positive $\hat{Gap}$, β bins with negative estimated gap, and θ bins having 0 estimated gap. Hence, $\alpha + \beta + \theta = n$. We would like to establish a lower bound on $\beta + \theta$. In order to have minimum number of bins with negative or zero $\hat{Gap}$, the value of the gap should be minimum for bins with a positive gap and maximum for bins with a negative gap. The minimum positive estimated gap for a bin is $Z(1 - \frac{d-1}{d})$ when $Z(d-1)$ balls have arrived in the system, of which only Z balls have been committed into the bin. The maximum negative estimated average that a bin may have in this case is $-\frac{Z(d-1)}{d}$. Hence,

$$\alpha.(Z(1 - \frac{d-1}{d})) + \beta.(-\frac{Z(d-1)}{d}) + \theta.0 = 0 \qquad \text{[From Lemma 3]}$$
$$\therefore \alpha = \beta(d-1)$$

As $\alpha + \beta + \theta = n$, we have $d\beta + \theta = n$. So, the number of bins with zero or negative $\hat{Gap}$ is $\Theta(n)$. □

Lemma 5. *Each ball chooses at least one bin having negative estimated gap among its* d *choices w.h.p. in γ rounds.*

Proof. Each ball selects independently and uniformly at random d candidate bins for its placement among the n bins. Hence the probability that bin B_i is chosen as a candidate for ball b_j is, $P_i^j = \binom{n-1}{d-1}/\binom{n}{d} = \frac{d}{n}$. Let there be c bins with zero or negative $\hat{Gap}$. The probability that neither of these bins are selected as candidate by a ball $= \binom{n-c}{d}/\binom{n}{d}$. The ball may re-select its candidates at most γ times. Therefore, the probability that neither of the c bins are selected in any of the γ tries $= \left(\binom{n-c}{d}/\binom{n}{d}\right)^\gamma$. Hence the probability that at least one bin with negative $\hat{Gap}$ is selected in the γ iteration is given by,

$$P(\text{at least one selected}) = 1 - \left(\frac{\binom{n-c}{d}}{\binom{n}{d}}\right)^\gamma \approx 1 - \frac{1}{2^{d\gamma}} \qquad \text{[Assuming } c = n/2 \text{ from Lemma 4]} \qquad (4)$$

For $d = 2$ and $\gamma = 2$, we obtain a probability of around 0.94. However, with $\gamma = \log n$, the probability becomes nearly $1 - \frac{1}{n}$. Further, we can show that approximately constant number of retries suffice.

Let the number of bins with positive gap at any point of time be $n^{1-\epsilon}$, where $0 \leq \epsilon \leq 1$. The probability P_{bneg} with which a bin with a zero or negative gap is chosen in γ iterations is given by,

$$P_{bneg} = 1 - \left(\frac{\binom{n^{1-\epsilon}}{d}}{\binom{n}{d}}\right)^\gamma$$

For a zero or a negative bin to be chosen with a high probability, we need $P_{bneg} \geq 1 - \frac{1}{n^\phi}$, where $\phi > 0$. Hence for $\left(1 - (\frac{n^{1-\epsilon}}{n})^{d\gamma}\right) > 1 - \frac{1}{n^\phi}$. Thus, $\gamma > \frac{\phi}{d\epsilon}$. Hence, at least one such bin is chosen by each ball in approximately constant γ re-polls or rounds per ball w.h.p. □

In the next lemma, we show that in practice only a couple of retries are needed to get a bin with zero or negative estimated gap.

Lemma 6. *The expected number of rounds, γ per ball to find a bin with zero or negative estimated gap is constant.*

Proof. Let p_i denote the probability that we find a zero or a negative bin at iteration i. Therefore, we have

$$p_i = \left(\prod_{1}^{i-1} P_{pos}\right) \cdot P_{neg} = \prod_{1}^{i-1} \frac{1}{2^d} \cdot \left(1 - \frac{1}{2^d}\right) = \frac{2^d - 1}{2^{id}}$$

where P_{pos} is the probability of selecting a bin with a positive estimated gap and P_{neg} is the probability of selecting a bin with a zero or negative gap. The expected number of rounds per ball, γ to find a zero or a negative gap is given by,

$$E[\gamma] = \sum ip_i = \left(2^d - 1\right) \sum_{i=1}^{\frac{\log n}{d}} \frac{i}{2^{id}} \tag{5}$$

Let,

$$S(i) = \sum_{i=1}^{\frac{\log n}{d}} \frac{i}{2^{id}} \tag{6}$$

$$\therefore \frac{S(i)}{2^d} = \sum_{i=1}^{\frac{\log n}{d}} \frac{i}{2^{(i+1)d}} \tag{7}$$

Subtracting Eq. (7) from Eq. (6), we have

$$\left(1 - \frac{1}{2^d}\right) S(i) = \frac{1}{2^d} - \frac{\log n}{d 2^{1+\log n}} + \frac{1}{2^d \left(2^d - 1\right)}$$

$$\therefore S(i) \approx \frac{2^d}{\left(2^d - 1\right)^2} \tag{8}$$

Substituting Eq. (8) in Eq. (5), we have

$$E[\gamma] \approx 1 + \frac{1}{2^d - 1} \tag{9}$$

$$\Rightarrow E[\gamma] < 2$$

□

Lemma 7. *The load of each bin tends to its estimated average.*

Proof. $IDEA$ places each ball into a bin with zero or negative $\hat{Gap}$, with high probability $1 - \frac{1}{n^\phi}$ (Lemma 5) using γ retries. When a ball is placed in a bin, its $\hat{Gap}$ increases. Thus, the probability that this bin will again get a ball lowers. On the other hand, the bins that had been chosen but the ball was not placed in them have a decrease in their estimated gap. Hence, the probability that a ball is placed in them increases. So, a bin with a negative or zero $\hat{Gap}$ has a higher probability of a ball being allocated to it, whereby its estimated gap tends towards 0 (in case of negative estimated gap-ed bins). On the other hand, bins with positive estimated gap receive a ball with low probability even when chosen as candidates, and their estimated gap decreases towards 0. Hence, we observe that the estimated gap of any bin tends towards 0. Since, estimated gap is the difference of the load and the estimated average of a bin and the gap tends to zero, the load of the bins becomes nearly equal to their estimated average w.h.p. □

Theorem 1. *The maximum load in any bin is $\lceil m/n \rceil + \Theta(1)$ w.h.p using the $IDEA$ allocation algorithm for the sequential, on-line and unweighted balls-into-bins problem.*

Proof. Using the above lemmas we observe that the estimated average of each bin finally becomes $\lceil m/n \rceil$ and the load in each bin is equal to its estimated average w.h.p. Hence the maximum load in any bin is $\lceil m/n \rceil + \Theta(1)$ w.h.p. □

Corollary 1. *The $IDEA$ algorithm provides a perfectly balanced allocation with constant gap.*

Proof. Since the maximum loaded bin has a load of $\lceil m/n \rceil + \Theta(1)$ w.h.p. (Theorem 1), the *Gap* is of $\Theta(1)$ providing a perfectly balanced allocation for the balls-into-bins problem with constant gap. □

5 Discussion

We note that the *Greedy[d]* algorithm can also retry γ times to find a bin of even lower total number of balls that what it could do in a single round. Still, the distribution of the balls in bins will be different than the $IDEA$ algorithm because the $IDEA$ algorithm explicitly uses the *expected gap* to make the decision of where the ball is placed. The key question is can the Greedy[d] algorithm give a constant gap and the answer is negative for a single retry because of the well known lower bound of $O(\ln \ln(n))$ [2], while for multiple retries γ has to be $\Theta(log(n))$ [9] to achieve a constant gap. $IDEA$ however requires only constant (< 2) retries in the expectation (Lemma 6), to achieve the constant gap. Further, it requires $\gamma = \frac{\phi}{d\epsilon}$ retries with high probability (Lemma 5).

A bin, B is chosen by d balls among n balls on expectation. However, the bin may be chosen αd times, $0 \leq \alpha \leq 1$ among the first ρ balls that arrive. As such, the $Greedy[d]$ choice algorithm will place the balls in empty or lesser loaded bins if available. In the remaining balls, B is chosen $(1 - \alpha)d$ times. Now, for large values of α, even if all these balls are placed in it, B will have a load far less than the average of the system. So the Gap increases. However, for $IDEA$ with large α values, the estimated average for B will be large and hence its estimated gap will be significantly lower than the other bins. So, it has a higher probability of a ball being allocated to it. Thus, when the remaining balls arrive and a small fraction of them are placed in B, its load will still be closer to the actual average as compared to the d-choice algorithm. This sensitivity towards skewness in the random choices also enables $IDEA$ to arrive at a better allocation than the d-choice.

6 Extended Framework

6.1 Weighted Case

In this section we consider the weighted case of the balls-into-bins problem where the balls have weights drawn from a distribution χ with an expected weight W^*, such that the weight of any ball W has a finite variance and can be bounded by $(W^* - k) \leq W \leq (W^* + k)$, where k is a constant. We apply the $IDEA$ algorithm and show that the gap is also constant w.h.p. in such scenarios.

Theorem 2. *The maximum load in any bin is $W^*(\lceil m/n \rceil + \Theta(1))$ w.h.p using the $IDEA$ allocation algorithm for the sequential, on-line and weighted balls-into-bins problem.*

Proof. Reworking the lemmas stated in Section 4 we observe that the estimated average of each bin converges to $W^* \lceil m/n \rceil$ and that the load in each bin tends to its estimated average w.h.p. Hence the maximum load in any bin is given by $W^*(\lceil m/n \rceil + \Theta(1))$ w.h.p. The complete proofs of the lemmas for the weighted case is provided in Appendix A. □

Corollary 2. *The $IDEA$ algorithm provides a perfectly balanced weighted allocation with constant gap even for the general weighted case of the Balls-into-bins problem.*

Proof. From Theorem 2 we observe that as the maximum load is $W^*(\lceil m/n \rceil + \Theta(1))$. Hence $IDEA$ provides a perfectly balanced allocation for the weighted case w.h.p. having a constant gap of $W^*\Theta(1)$. □

6.2 Multi-Dimensional Case

In this section, we consider the multidimensional (md), variant of the balls and bins problem. One multidimensional variant, proposed by [5] is as follows: Consider throwing m balls into n bins, where each ball is a uniform D-dimensional (0-1) vector of weight f. Here, each ball has exactly f non-zero entries chosen uniformly among all $\binom{D}{f}$ possibilities. The average load in each dimension for each bin is given as mf/nD.

Let $l(a, b)$ be the load in the dimension a for the b^{th} bin. The gap in a dimension (across the bins) is given by $gap(a) = max_b l(a, b) avg(a)$, where $avg(a)$ is the average load in the dimension a. The maximum gap across all the dimensions, $max_a gap(a)$, then determines the load balance across all the bins and the dimensions. Thus, for the multidimensional balanced allocation problem, the objective is to minimize the maximum gap (across any dimension). We refer to the multidimensional ball as md-ball and the multidimensional bin as md-bin.

In another variation of multidimensional balanced allocation the constraint of uniform distribution for populated entries is removed. Here again, each ball is a D dimensional 0-1 vector and each ball has exactly f populated dimensions, but these populated dimensions can have an arbitrary distribution. In the third variation that is most general of the three, the number of populated dimensions, f, may be different across the balls, where f then is a random variable with an appropriate distribution.

Each md-ball has f populated dimensions, where f could be constant across the balls or a random variable with a given distribution. Let, $s_i(t)$ denote the sum of the loads (minus corresponding dimension averages) across all D dimensions for the bin i at time t, expressed as $s_i(t) = \sum_{d=1}^{D} x_i^d$. This reduces the problem to that of the *scalar weighted case*. The $IDEA$ algorithm works based on the sum of the dimensions for each bin. Also, for each choice of the bin, its estimated average is now incremented by $\frac{f}{d}$.

Theorem 3. *For the multi-dimensional scenario, the $IDEA$ algorithm provides a constant gap for uniform distribution of the f populated dimensions for each ball with $m = n$.*

Proof. Following the analysis in Section 6.1, the Gap in the system is bounded by $\Theta(1)$. Hence, the difference of the number of balls in the maximum bin and the actual average of the system is constant. For $m = n$, the average is 1 and so the number of balls in the maximum bin is also a constant. Given a uniform distribution of the f populated dimensions of each ball over D, the Gap is bounded by $\Theta(1)$. □

7 Conclusions

This paper proposes the *Improved D-choice with Estimated Average*, $IDEA$ algorithm which w.h.p. provides a perfectly balanced allocation for the sequential, online and uniform weighted balls-into-bins problem. We propose a better metric for greedy placement of the balls using the estimated average of the system for each bin. We show that for a constant d choice and approximately constant number of rounds per ball, the maximum loaded bin in $IDEA$ is $\lceil m/n \rceil + \Theta(1)$ w.h.p. This result holds for $m = n$ case as well as the heavily loaded scenario where $m >> n$. We also extends the solution for the general weighted case (with $m >> n$) to show similar results for balls with weights taken from an arbitrary distribution with finite variance and for the multi-dimensional case with $m = n$ for uniform distribution of f populated dimensions over the D total dimensions.

The application of $IDEA$ in parallel scenario along with its performance for the multi-dimensional balls-into-bins problem in the heavy case provides an interesting future work in this domain.

References

[1] M. Adler, S. Chakrabarti, M. Mitzenmacher, and L. Rasmussen. Parallel Randomized Load Balancing. In *STOC*, pages 238–247, 1995.

[2] Y. Azar, A. Z. Broder, A. R. Karlin, and E. Upfal. Balanced Allocations. *SIAM J. of Computing*, 29(1):180–200, 1999.

[3] P. Berenbrink, A. Czumaj, A. Steger, and B. Vöcking. Balanced Allocations: The Heavily Loaded Case. *SIAM J. of Computing*, 35(6):1350–1385, 2006.

[4] P. Berenbrink, F. Meyer auf der Heide, and K. Schröder. Allocating Weighted Jobs in Parallel. *Theor. Comput. Syst.*, 32:361–386, 1999.

[5] A. Broder and M. Mitzenmacher. Multidimensional Balanced Allocations. In *SODA*, pages 195–196, 2005.

[6] R. Cole, B. Maggs, F. M. auf der Heide, M. Mitzenmacher, A. Richa, K. Schroder, R. Sitaraman, and B. Vocking. Randomized protocols for low congestion circuit routing in multi-stage interconnection networks. In *Thirteith Annual Symposium ACM symposium on the Theory of Computing*, pages 378 – 388, May 1998.

[7] A. Czumaj, C. Riley, and C. Scheideler. Perfectly Balanced Allocations. In *Workshop on Randomization and Approximation Techniques in Computer Science*, pages 240–251, 2003.

[8] D. Dubhashi and D. Ranjan. Balls and Bins: A Study in Negative Dependencies. *Random Structures and Algorithms*, 13:99–124, 1996.

[9] P. B. Godfrey. Balls and Bins with Structure: Balanced Allocations on Hypergraphs. In *SODA*, 2008.

[10] G. H. Gonnet. Expected Length of the Longest Probe Sequence in Hash Code Searching. *J. of ACM*, 28(2):289–304, 1981.

[11] R. M. Karp. Random graphs, random walks, differential equations and the probabilistic analysis of algorithms. In *STACS*, pages 1–2, 1998.

[12] C. Lenzen and R. Wattenhofer. Tight Bounds for Parallel Randomized Load Balancing. In *STOC*, 2001.

[13] M. Mitzenmacher. *The Power of Two Choices in Randomized Load Balancing*. PhD thesis, University of California at Berkeley, 1996.

[14] Y. Peres, K. Talwar, and U. Wieder. The (1+β)-choice process and weighted balls-into-bins. In *SODA*, pages 1613–1619, 2010.

[15] P. Sanders, S. Egner, and J. Korst. Fast concurrent access to parallel disks. *Algorithmica*, 35(1):21–55, 2003.

[16] K. Talwar and U. Wieder. Balanced allocations: The Weighted Case. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, STOC '07, pages 256–265, New York, NY, USA, 2007. ACM.

[17] B. Vocking. How asymmetry helps load balancing. In *FOCS*, pages 131 – 141, 1999.

A Theoretical Framework for the Weighted Case

In this section, we provide a theoretical proof of the constant gap performance of the weighted version of the $IDEA$ algorithm. We follow the same proof sketch as in the case of ball with unit weight. Further, we too assume here m balls and n bins, $m \gg n$.

Lemma 8. *If each weighted ball chooses d bins i.u.r. out of n bins, each bin is chosen by $\frac{md}{n}$ balls on expectation, and by at most $\frac{md}{n} \log n$ weighted balls with high probability.*

Proof. Similar to Proof of Lemma 1. □

Lemma 9. *At any iteration, the estimated average of each bin is approximately equal to the current average w.h.p.*

Proof. We assume here that Z balls have already arrived and have been placed among the n bins. The number of balls that chose bin B among its d candidates is $\frac{Zd}{n}$ on expectation, since each bin can be chosen by a ball with a probability of $\frac{d}{n}$. The number of such balls is also bounded by $(1 + \log n)\frac{Zd}{n}$ with high probability (by Lemma 1). However, a bin does not increment its estimated average by more than d times when n balls are thrown. For each ping a bin B receives, it increases its estimated average by $\frac{W}{d}$, which is bounded by $\frac{W^*-k}{d} \le \frac{W}{d} \le \frac{W^*+k}{d}$. Hence the current value of $\hat{A_B}$ is given by,

$$\hat{A_B} = \frac{Zd}{n} \cdot \frac{W^* \pm k}{d} = \frac{Z\,(W^* \pm k)}{n} \text{ , which is the current average.}$$

Hence, the estimated average $\hat{A}$ of any bin is nearly equal to the actual average w.h.p. □

Lemma 10. *The sum of the estimated gap, $\hat{Gap}$ over all the bins at any time is zero.*

Proof. Each ball chooses d candidate bins uniformly and randomly and is finally allocated to the bin having the lowest estimated gap. Hence for all the d chosen bins, their estimated average increases by W/d. The load of $d-1$ bins which do not receive the ball remains same, and thus their estimated gap decreases by the above factor. However, for the bin in which the ball is placed, its load increases by 1 and its estimated gap increases by $W\left(1-\frac{1}{d}\right)$. Hence the sum of change of the estimated gap over the d chosen bins in any iteration is $W\left(1-\frac{1}{d}\right) + (d-1)\frac{-W}{d} = 0$. □

Lemma 11. *The number of bins having a* zero *or* negative *estimated gap, $\hat{Gap}$ is $\Theta(n)$.*

Proof. Let there be α bins with positive $\hat{Gap}$, β bins with negative estimated gap, and θ bins having 0 estimated gap. Hence, $\alpha + \beta + \theta = n$. We would like to establish a lower bound on $\beta + \theta$. In order to have minimum number of bins with negative or zero $\hat{Gap}$, the value of the gap should be minimum for bins with a positive gap and maximum for bins with a negative gap. The minimum positive estimated gap for a bin is $ZW_{min} - \frac{1}{d}\sum_{i=1}^{Z(d-1)} W_i \approx Z(W^* \pm k)\left(1-\frac{d-1}{d}\right)$ when $Z(d-1)$ balls have arrived in the system, of which only Z balls have been committed into the bin. We have $W_{min} = min\{W_1, W_2, \ldots, W_{Z(d-1)}\}$. The maximum negative estimated average that a bin may have in this case is $-\frac{\sum_{i=1}^{Z(d-1)} W_i}{d} \approx -\frac{Z(d-1)(W^* \pm k)}{d}$. Hence,

$$\alpha.(Z(W^* \pm k)(1 - \frac{d-1}{d})) + \beta.(-\frac{Z(W^* \pm k)(d-1)}{d}) + \theta.0 = 0 \qquad \text{[From Lemma 3]}$$
$$\therefore \alpha = \beta(d-1)$$

Further, $\alpha + \beta + \theta = n$. Hence, d$\beta + \theta = n$. So, the number of bins with zero or negative $\hat{Gap}$ is $\Theta(n)$. □

Lemma 12. *Each ball chooses at least one bin having negative estimated gap among its d choices w.h.p. in γ rounds.*

Proof. Similar to Proof of Lemma 5. □

Lemma 13. *The expected number of rounds, γ per ball to find a bin with zero or negative estimated gap is constant.*

Proof. Similar to Proof of Lemma 6. □

Lemma 14. *The load of each bin tends to its estimated average.*

Proof. Similar to Proof of Lemma 7. □