Randomized Algorithms (RA-MIRI): Assignment #2

Luís Chávez, Antonio Peso Vilella

November 2025

1 Introduction

Consider a collection of m bins and n balls. For each ball, we draw a certain number $d \geq 1$ of bins, uniformly at random from $\{1, \dots, m\}$ and with replacement, and use some rule (deterministic or randomized) to choose in which of the d bins we put the ball.

Let $X_i(n)$ be the load (amount of balls) of the bin $i \in \{1, \dots, m\}$ when allocating a total of n balls. In an ideal scenario, we would expect to find a balance of loads in every bin (i.e. $X_i(n) = n/m$), so we will study the *gap* between the maximum load and the ideal scenario. In particular, we want to study the evolution of

$$G_n = \max_{1 \leq i \leq m} \left\{ X_i(n) - \frac{n}{m} \right\}$$

In this experimental study, we will consider different allocation strategies of the n balls in the m bins. In particular, the strategies to follow will be:

1. Single Ball Setup: a ball is allocated in a single bin using a random bin or set of bins ($d \in \{1, \dots, 6\}$. The allocation will depend on the amount of bins selected and other parameters (e.g. $\beta \in [0, 1]$).
2. Batch Setup: a batch of b balls is allocated using "outdated" information, i.e. the decision is taken based on the loads before any of the balls in the batch are allocated.
3. Partial Information: the allocation decision will be taken based on partial information of the loads of the bins. In particular, we will choose a bin based on the position of the load with respect to the median ($k = 1$) and the percentiles 25% and 75% ($k = 2$).

2 Single Ball Setup

In this setup, we will experiment with different strategies to allocate a single ball in a bin. The different strategies are:

- one-choice: a single bin will be taken at random and the ball will be allocated in it.
- d-choice: d bins are chosen at random (with replacement). The ball will be allocated in the bin with the smallest loss. In case of a draw, one will be chosen at random.
- $(1+\beta)$-choice: given $\beta \in (0,1)$, we define a Bernoulli random variable X with parameter β (i.e. $X \sim Bernoulli(\beta)$). If $X = 0$ we will use the one-choice strategy and if $X = 1$ we will use the d-choice strategy.

It is easy to see that the first and second strategies are equivalent to the choices of $\beta = 0$ and $\beta = 1$, respectively.

The *one-choice* and *two-choice* strategies have been extensively studied. We expect to see a lower gap in the two-choice experiment following the well-known "power of two random choices" paradigm.

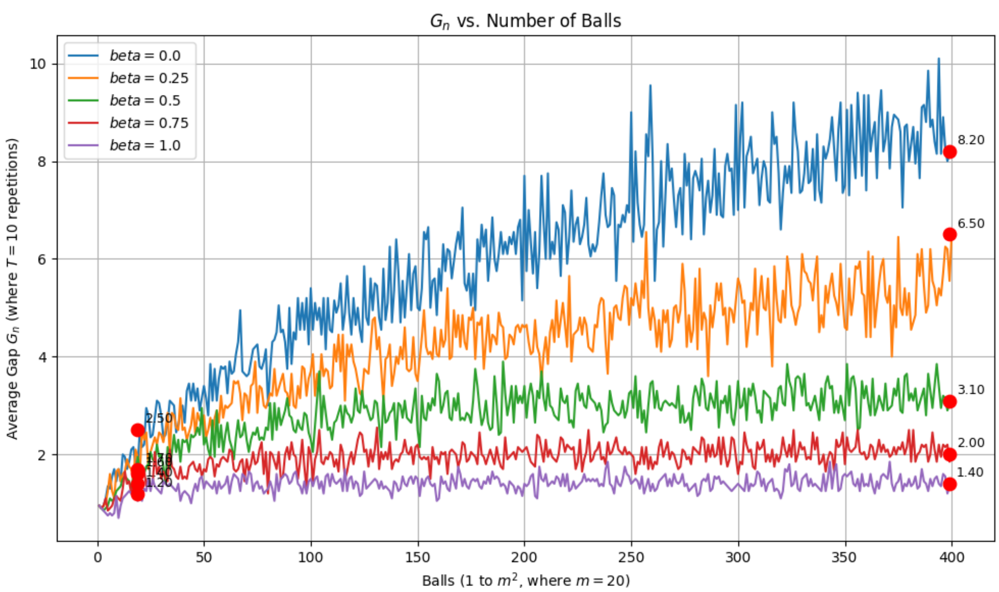

Figure 1: Evolution of the average gap G_n using the single ball setup with $\beta \in \{0, 0.25, 0.5, 0.75, 1\}$ for one-choice and two-choice, $m = 20$ bins and $T = 10$ repetitions.

In Figure 1 we can observe that the closer the strategy is to the two-choice strategy, the smaller the average gap growth. This growth is almost non-existent in the two-choice strategy, where the gap stays almost stable as the number of balls grows to $n = m^2$. The two-choice strategy allows for a much more balanced allocation of the balls, using the power of randomized choices and the up-to-date information of the loads.

Another important take from Figure 1 is the width that the strategies present as the number of balls grows. In the one-choice strategy (and for lower values of β), the variance grows as the number of balls approaches the heavy-load scenario, while in the two-choice strategy (and for higher values of β) the variance remains almost constant. It is expected to get a variation of the gap in all strategies, since the allocation is based on random choices, but it is clear that the use of information (two-choice) makes a difference both in the expected value and variance of the gap.

3 Batch Setup

In this setup, we will repeat the same experiments as in Section 2 with a delay in the update of the information. In particular, the information on the loads of the bins will not be updated until all the balls in the batch have been allocated (i.e. every ball in the batch will "choose" a bin, but the load of that bin will not be updated until all the balls in the batch have "chosen" a bin).

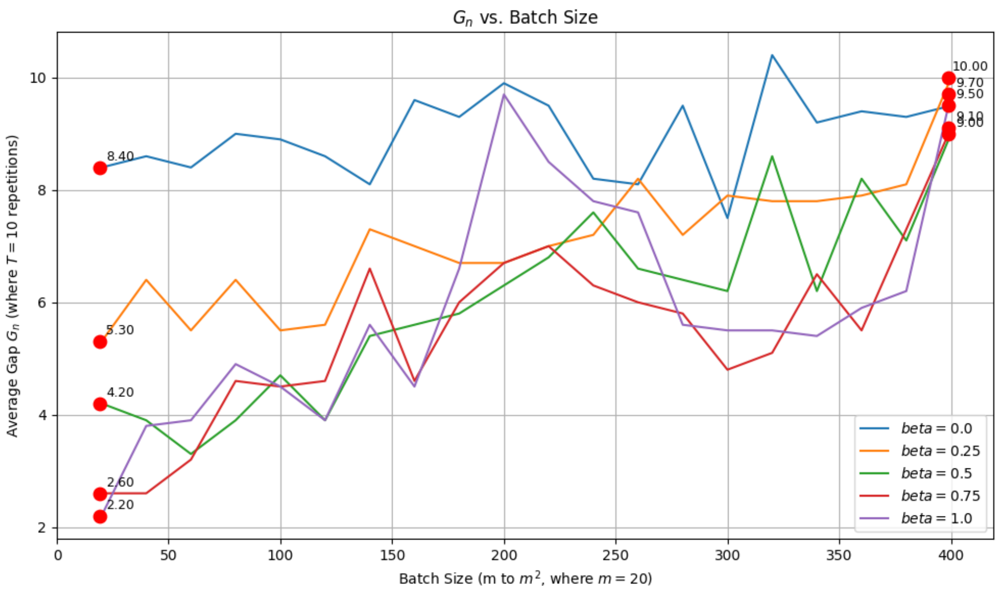

Figure 2: Evolution of the average gap G_n using the batch setup with $\beta \in \{0, 0.25, 0.5, 0.75, 1\}$ for one-choice and two-choice, $m = 20$ bins, $n = m^2$ balls and $T = 10$ repetitions.

In Figure 2 we can see the results of the experiments that we have carried out with a fixed number of balls ($n = m^2$) and a growing number of batches ($b = \lambda m, \lambda = 1, \ldots m$). The gap evolves from a clear separation between the one-choice and the two-choice strategies to a convergence towards the one-choice strategy. In particular, from left ($b = m$) to right ($b = m^2$), the two-choice strategy evolves as follows:

1. $b = m$: The size of the batches is small and the information on the loads updates fast enough for a balanced allocation of balls (similar to the last values in Figure 1)

2. $b = \lambda m$: As the batch grows, the delay of information on the loads causes the two-load strategy to allocate balls in the same bins (the ones that did not get balls in the first batches, will get a lot of balls in the next batches as the strategy will tend to choose them).

3. $b = \frac{m^2}{2}$: in this specific batch size, the gap presents a sudden spike. The balls are divided into two equally big batches and the first batch is allocated at random (as the one-choice strategy). The second batch, however, will use the updated information from the allocation of the balls from the first batch and focus the other half of the balls in the same low loaded bins, creating a sudden increase in their loads (and therefore an increase in the gap).

4. $b \geq \frac{m^2}{2}$: The situation is similar to the previous case, but the focused allocation of the balls is not as pronounced, since the second batch contains $m^2 - b \leq \frac{m^2}{2}$ balls. Therefore, the second batch cannot generate a gap as big as the one created when the two batches had the exact same size.

5. $b \to m^2$: The batch size is so large that the two-choice strategy allocates the balls randomly in the batch, converging to the one-choice strategy (lack of information turns into random choices).

4 Partial Information

The purpose of this section is to evaluate how the distribution gap behaves when allocation decisions are made under *partial information* rather than by inspecting the exact load of the bins. In this setting, the algorithm cannot query or compare numerical bin loads; instead, it may only issue a limited number of *binary questions* about each candidate bin. The parameter $k \in \{1, 2\}$ represents the number of such binary queries available per allocation step.
When $k = 1$, the algorithm may only determine whether a bin's load is *above or below the median* (the 50th percentile) of the current load distribution. When $k = 2$, the algorithm can ask a second binary question whenever both candidate bins fall on the same side of the median. If both are below the median, the algorithm checks whether the bin lies below the 25th percentile; if both are above the median, it checks whether the bin lies below the 75th percentile. In every case, the selected bin is the one classified as lighter according to the available binary information; if both bins return identical answers to all queries, the choice is made uniformly at random. These coarse categories replace the exact load comparisons used in the standard two-choice scheme.
The goal of the experiments is to empirically compare the resulting distribution gap G_n under these restricted-information models, and to quantify how reducing the available load information impacts the overall balance of the allocation process.

4.1 $k = 1$: Median Load

Figure 3 reports the evolution of the average gap G_n as the size of the batches grows with a fixed ball amount of $n = m^2$ under partial information with $k = 1$ using a batched allocation strategy. In this setting, allocation decisions rely exclusively on a single binary comparison against the median load, which replaces the exact load inspection available in the full-information two-choice scheme. The results exhibit several important differences relative to the classical model.

Effect of β Under Partial Information. Although higher values of β continue to produce smaller gaps, the differences between the β curves are significantly reduced compared to the online scenario. The coarse median-based signal often fails to distinguish the lighter bin among the two candidates, causing the decision to devolve into a random choice. Consequently, the $(1+\beta)$-choice mechanism provides only a limited improvement over the one-choice behavior: the curves follow the same general trend and stay relatively close throughout the entire range of batch sizes. Nonetheless, the ordering remains consistent, with larger β values achieving smaller gaps.

Variability Across Batch Sizes. The curves exhibit moderate fluctuations, especially for small and intermediate values of β. These fluctuations originate from the fact that, for many candidate pairs, the available information is insufficient to distinguish between bins, amplifying the stochastic nature of the allocation. When β is close to 1, these fluctuations diminish, as the algorithm benefits from favoring the two-choice operation even with partial information. However, variability remains noticeably higher than in strategies that rely on exact loads.

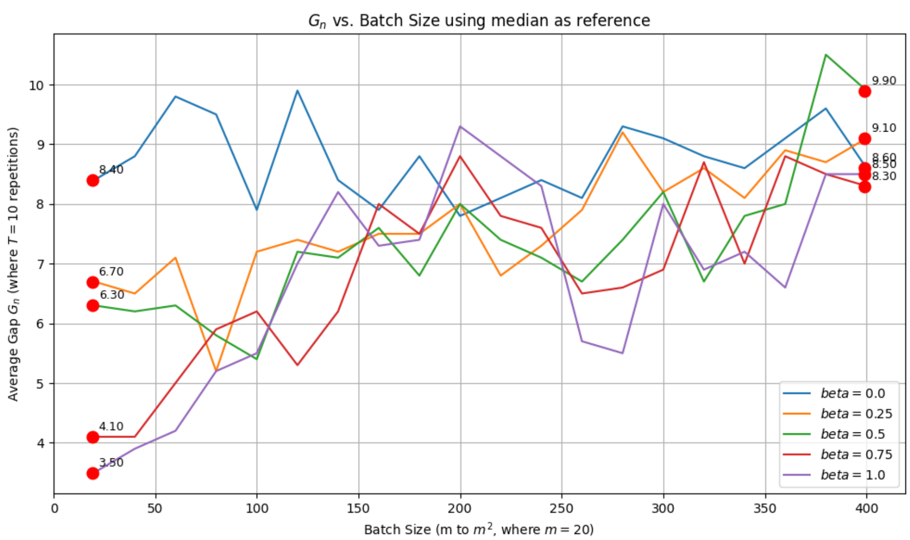

Figure 3: G_n vs. Batch Size using the median as reference.

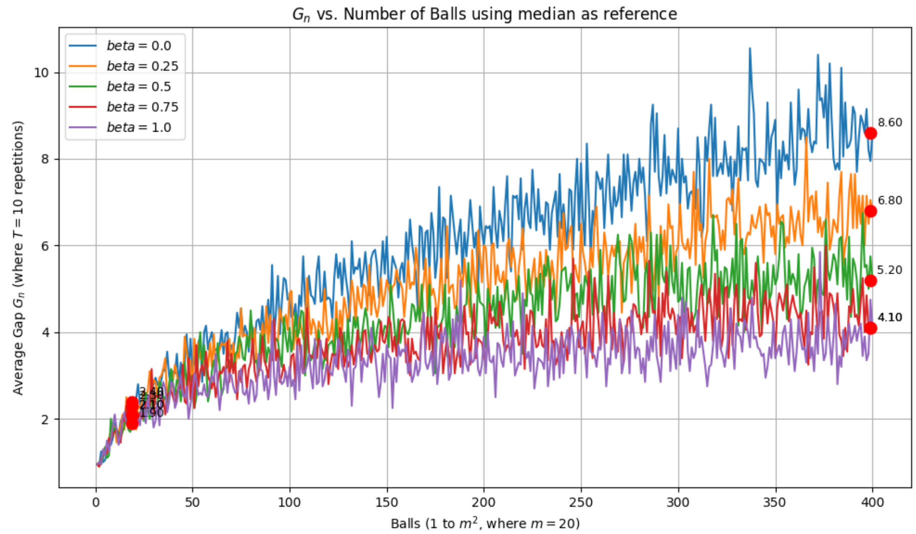

Figure 4: G_n vs. The number of balls using the single ball setup with partial information with the media as reference.

Figure 4 shows the evolution of the gap G_n when the allocation decisions rely only on a single binary query ($k = 1$), indicating whether each candidate bin is above or below the median load and using a single ball allocation strategy. Compared to the full-information curves (Figure 1), the effect of β is significantly reduced: although larger β values still perform better, the difference between strategies is much smaller.
The curves also exhibit higher variability, reflecting the greater randomness in the decisions. Overall, partial information weakens the benefits of the $(1 + \beta)$-choice scheme: the ordering with respect to β remains, but the gains are substantially smaller and the resulting allocations are noticeably less balanced.

4.2 $k = 2$: 25% and 75% Load

Figures 3 and 5 show the evolution of the average gap G_n as a function of the batch size b when using one or two binary questions per allocation. In the $k = 1$ case, the algorithm only checks whether each candidate bin is above or below the median load; for $k = 2$ it may refine this decision by determining whether the bin lies in the top or bottom quartile of its half of the distribution.

General Effect of the Second Question. Introducing the second binary question yields a consistent but moderate improvement across all values of β. The $k = 2$ curves tend to lie slightly below their $k = 1$ counterparts, especially for intermediate and large batch sizes. This indicates that the quartile-based refinement helps resolve some of the ambiguous cases where both candidate bins fall on the same side of the median.

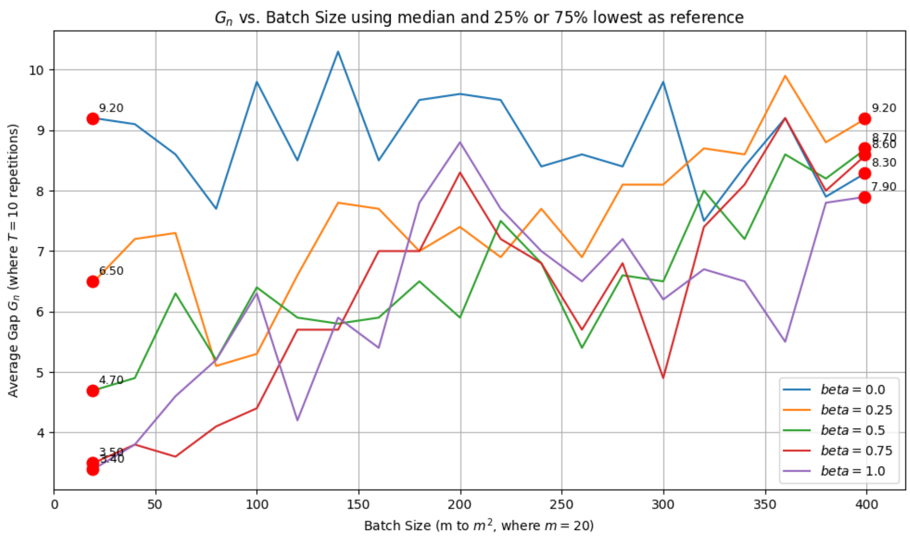

Figure 5: G_n vs. Batch Size using the median and the 25 or 75 percentile as reference.

In the single-ball allocation setup (Figures 4 and 6), the effect of introducing the second binary question is similar to what was observed in the batched case, but becomes slightly more evident as n increases. The curves for $k = 2$ remain consistently below those for $k = 1$ across all values of β, reflecting the benefit of the additional percentile-based refinement whenever the median alone does not distinguish the two candidates.
The improvement is modest, the trajectories for both k values stay close, and a large portion of the decisions remain effectively random due to the coarse information available. Even with two questions, many cases still can't be resolved reliably, and this limits how much the allocation can be improved.
Overall, increasing from $k = 1$ to $k = 2$ yields a small but consistent reduction in the gap throughout the process, especially near the heavy-load regime, but the behavior continues to resemble a noisy variant of the one-choice strategy.

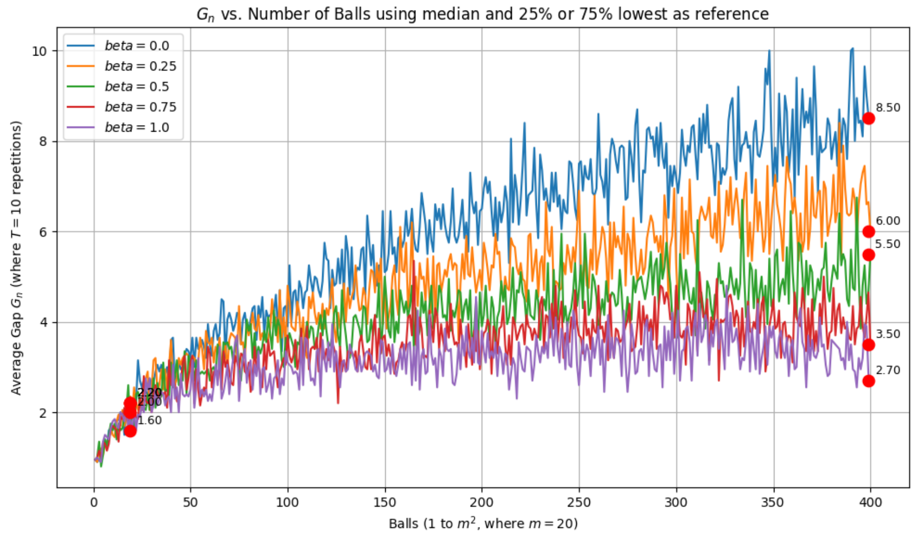

Figure 6: G_n vs. Batch Size using the median and the 25 or 75 percentile as reference.

5 Conclusions

These set of experiments focus on the interaction between delayed (batching) and partial ($k = 1, 2$) information in the context of randomization. They show how, in the presence of updated and complete information, allocating balls following the "power of two random choices" paradigm creates a balanced system of balls and bins. When the information is updated but only partially complete (single ball setup with $k = 1, 2$), the system presents higher variance and average gap, but the two-choice strategy still outperforms pure randomization. However, in the presence of delayed information, the "power of two random choices" paradigm breaks down and presents no differences with the purely randomized allocation of balls.
You can access the repository containing the files related to the assignment in the following link: GitHub repository.