A FIRST LOOK AT THE DECISIONAL WEIGHTED BALLS-INTO-BINS PROBLEM

We introduce a new version of the classic Mathematical Balls-Into-Bins problem.

Problem Statement

Our variation:

"Weighted"
Each ball has a unique weight.

"Decisional"
There is a condition for success, where the bin can only hold a maximum weight.

Equations

n → no. of balls, n is a positive integer
m → no. of bins, m is a positive integer
W → maximum weight that each bin can hold
b → weights of n balls, where $\mathbf{b}=(b_1,\ldots,b_n)$ is a tuple where $1\leq b_i \leq W$ for all $i \in \{1,..,n\}$
We define $W_{p,\mathbf{b},m}$ to take on this minimum value for a particular p,b,m: $W_{p,\mathbf{b},m} = \min_{p\leq P_{\mathbf{b},m,w}} W.$

Why are we running simulations?

Finding a closed form is complex, because:

- large values of n and m
- different values of W
- varying ball weights

Previous research [1, 2] on both decisional and weighted versions could not find an easily computable closed form, and our problem is even more complex.

Simulations

Methodology

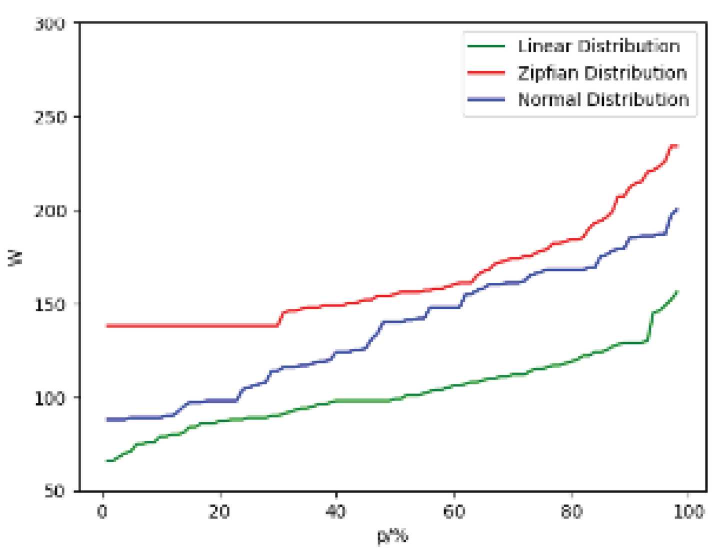
Distribution types: linear, normal, Zipfian

1. Assign each ball a weight based on a distribution type
2. Simulate throwing balls into bins uniformly at random
3. Retrieve W for a particular p.

Aim:

To observe the relationship between the success probability and the weight of the heaviest bin.

Results



Trends

1. **All graphs have similar shape of increasing linearly before having a steep incline as p approaches 100.**
   a. As p increases, the more extreme cases have to be accomodated for (ie the heaviest balls land in the same bin)
2. **Staircase structure**
3. **Zipfian requires a higher value of W to succeed, followed by normal, followed by linear.**
4. **a. The heaviest ball present in the zipfian distribution is heavier than that of the normal distribution, which is heavier than that of the linear distribution.**
5. **As the number of bins increases, the value of W for the same p decreases.**

Impact and Future Work

- Useful for users who categorise and encrypt sensitive data (eg: hospital/bank records, PDF files, etc.) in multimaps
- Using our graphs, users who encrypt sensitive data can accurately measure storage savings, making their storage more efficient
- We could delve into more robust formulae for finding p given specific parameters to increase accuracy in results.

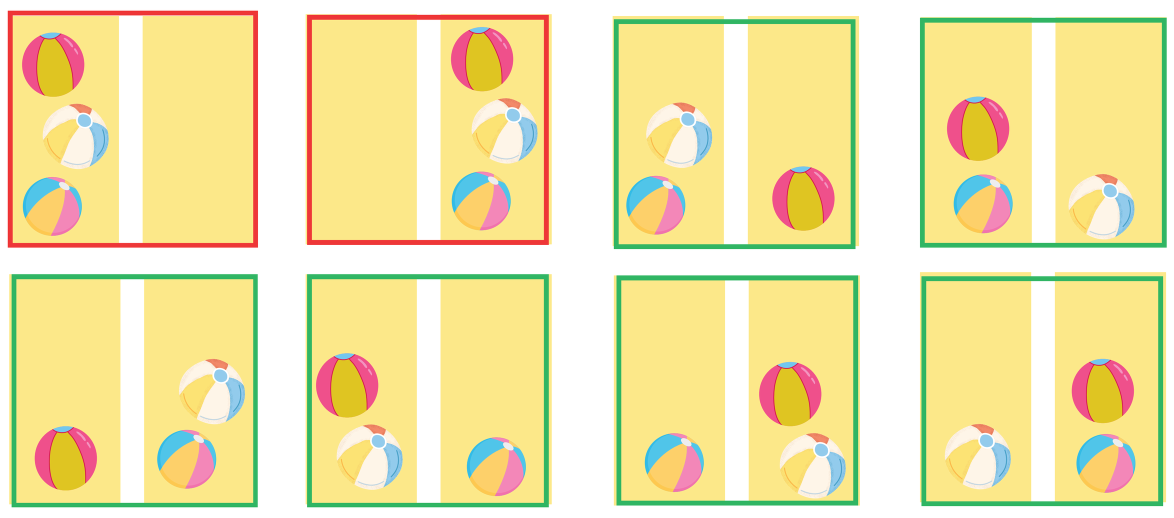
Example

We throw 3 balls into 2 bins uniformly at random

$$p = \left(1 - \frac{\text{failed ways}}{\text{total ways}}\right) \times 100\%$$
$$\text{total ways} = m^n$$

n = 3, m = 2, W = 2

8 ways for the balls to land



$$p = \left(1 - \frac{2}{8}\right) \times 100\% = 75\%$$

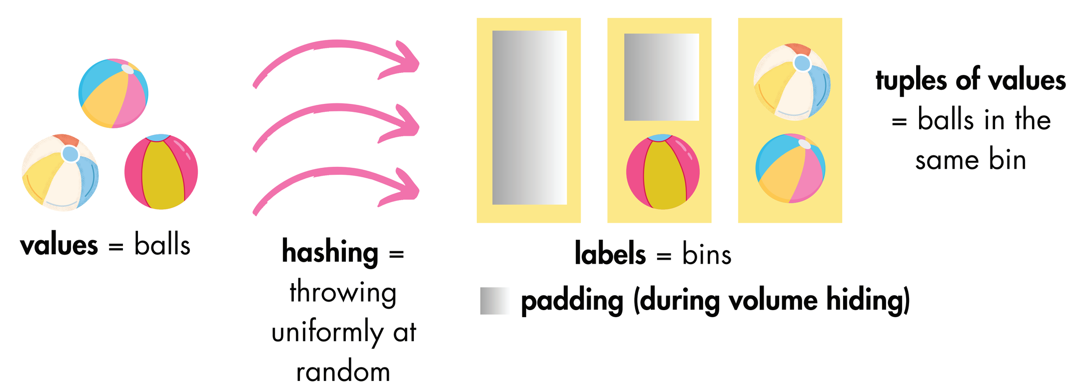
Real-world Application: Cryptography

Schemes:

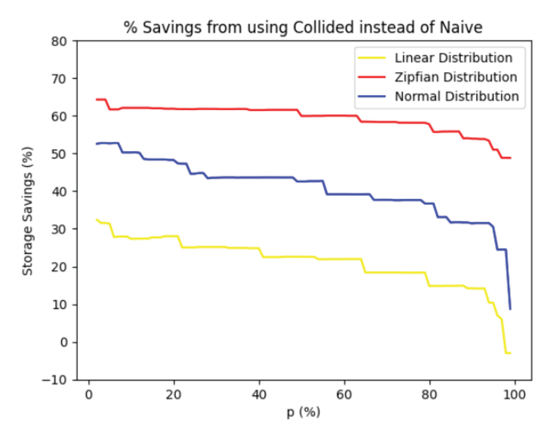
Naive Volume Hiding Scheme (NVHS)
Pad all tuples to length of longest tuple. Takes up too much storage space, and length of longest tuple is leaked.

Colliding Volume Hiding Scheme (CVHS)
Assign tuples to labels randomly, and then pad to a fixed maximum length. Failure occurs if a tuple exceeds this length.

We realise that our Balls-into-Bins problem directly parallels CVHS. From our earlier results, we can directly measure the **memory complexity** by modelling the labels as bins and the tuples as balls:



Results



Trends

1. Using CVHS is most efficient for Zipfian, normal, then linear.
2. CVHS saves storage compared to NVHS.

--> Our Balls-into-Bins problem allows us to measure exactly how much storage is saved

Members:
Arshia Mahajan, Raffles Girls' School
Chelsea Ling Xinyi, Raffles Girls' School
Tan Gim Qi Danielle Jeanne, Raffles Girls' School
Mentor:
Dr Ruth Ng Ii-Yung, DSO National Laboratories

YDSP Young Defence Scientists Programme
DSTA Defence Science & Technology Agency
DSO National Laboratories