Technical University of Crete
School of Electrical and Computer Engineering
Course: **Randomized Algorithms**
Exercise 3
Angelopoulos Dimitris - 2020030038

Max-Cut Algorithms in Random Graphs

For the first exercise we will experiment with max cut algorithms in random graphs $G_{n,p} = \langle V, E \rangle$. We use two algorithms, one that is deterministic and another that is not.

1. **Randomized Max-Cut Algorithm** The large cut algorithm guarantees that there exists a cut that is $\geq \frac{|E|}{2}$. Initially we partition V into two sets A and B by assigning each vertex independently and uniformly at random. Thus, for some vertex $v \in V$, we have $P(v \in A) = P(v \in B) = \frac{1}{2}$. We then define a random variable X_i that takes the value 1 if some edge e_i connects A to B.

   That means that the sum of all X_i gives us the cut $C(A, B)$. Additionally, we know that

   $$\mathbb{E}[C(A,B)] = \mathbb{E}\left[\sum_{i=1}^{|E|} X_i\right] = \frac{|E|}{2}$$

   Using the expectation argument we get that $C(A, B) \geq \frac{|E|}{2}$.

2. **Deterministic Max-Cut Algorithm** This is the deterministic counterpart of the former algorithm and also guarantees a cut with the property $\mathbb{E}[C(A, B)] = \frac{|E|}{2}$ and thus $C(A, B) \geq \frac{|E|}{2}$. This is a greedy algorithm that partitions the initial vertex set V by comparing the number of neighbors each partitioned set A, B has. By doing this we essentially try to maximize the amount of edges connecting the partitioned sets in each step. Even though this method is suboptimal, it is enough to achieve a large cut.

Now we will generate a $G_{n,p}$ with probabilities $p = 10^{-2}, 10^{-1}, 0.8$ and simulate the problem. After running both algorithms mentioned above we observe that the randomized version finds a large cut for every number of vertices n and for any probability p. On the other

hand the deterministic one finds a large cut for any n and $p = 0.8$, but sometimes fails when we lower the probability and finds a cut that is close to $\frac{|E|}{2}$ but not larger.

Concentration of Empty Bins

In this experiment we consider a function f, that takes as inputs m random variables X_i with $i = 1, \ldots, m$. Where X_i represents the bin into which the i-th ball falls. The function returns a scalar value that represents the number of empty bins after m balls are thrown to n bins. For some fixed inputs $\mathbf{x} = (x_1, \ldots, x_i, x_{i+1}, \ldots, x_m)$ and $\mathbf{y} = (x_1, \ldots, y_i, x_{i+1}, \ldots, x_m)$, we want f to be Lipschitz.

$$\|f(\mathbf{x}) - f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$$

Since f returns scalar values, we choose the norm of the left hand side to be the L_1 norm. Likewise we can quantify the value represented in the right hand side using the Hamming norm, since $\mathbf{x}$, $\mathbf{y}$ differ in only one element, meaning that $\|\mathbf{x} - \mathbf{y}\|_H = 1$. Knowing that $\mathbf{x}$, $\mathbf{y} \in \mathbf{X}$, where $\mathbf{X}$ is some finite dimension space, it holds that all norms are equivalent; meaning that f needs to satisfy the following condition.

$$|f(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_m) - f(x_1, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_m)| \leq L$$

Our function f is Lipschitz continuous if there exists $L \geq 0$ such that the former condition hold. Assume that $f(\mathbf{x}) = k$ empty bins, if we move some ball to some other empty bin and the initial bin has only one ball then $f(\mathbf{y}) = k$ (the same holds if we move some ball to a non-empty bin and the initial bin has more than one ball). For the case where we move a ball to an empty bin, given that the initial bin has more than one balls, we get $f(\mathbf{y}) = k - 1$. Likewise, if we move a ball to a non-empty bin, given that the initial bin has only one ball, we get $f(\mathbf{y}) = k + 1$. That means that the Lipschitz criteria holds with $L = 1$ since $|f(\mathbf{x}) - f(\mathbf{y})| \leq 1$.

Since $X_1, \ldots, X_m$ are independent random variables, using McDiarmid's inequality we get the following.

$$P(|f - \mathbb{E}[f]| \geq \epsilon) \leq 2e^{-2\epsilon^2/m}$$

The theoretical mean of the number of empty bins is given by $\mathbb{E}[f] = n\left(1 - \frac{1}{n}\right)^m$. Before simulating the problem we will assume that $m = n$ and we set $\epsilon = \lambda\sqrt{n}$. By doing this we can expect that the experimental values will be concentrated around the mean with deviation of the order $\sqrt{n}$.

First of all we will be running the algorithm for $n = 100, 1000$ and $K = 10^3$ times to get the following plots.

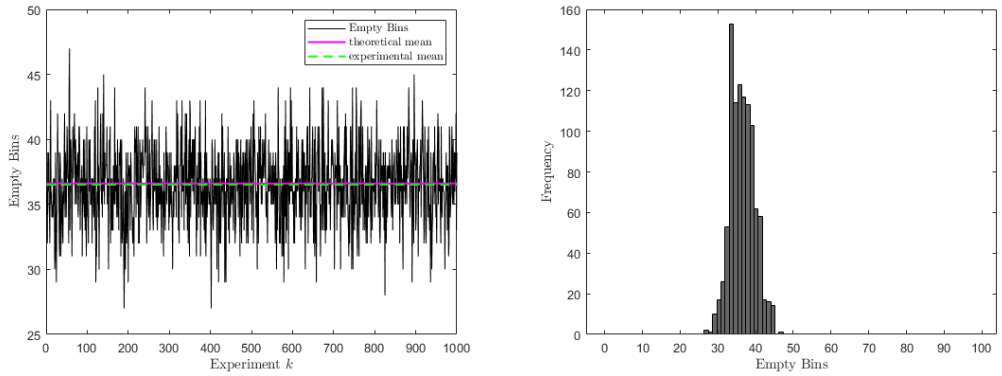

Figure 1: Results of # empty bins after throwing $n = 100$ balls.

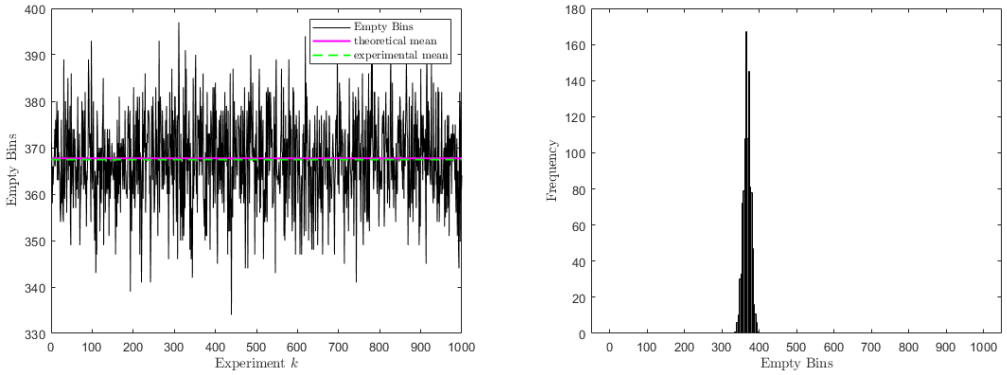

Figure 2: Results of # empty bins after throwing $n = 1000$ balls.

For both cases we observe that the experimental mean is almost the same to the theoretical one. Additionally, the histogram of each case is concentrated around its experimental mean and in fact we do not observe any empty bin outside the interval $[\mathbb{E}[f] - \sqrt{n}, \mathbb{E}[f] + \sqrt{n}]$. Now we will simulate the experiment once more (providing only the histograms) for $K = 10^4$ to get the plots in Fig. 3.

We also observe that as we increase n the histogram starts to get even more concentrated around the mean, reminding us the delta function as shown in Fig. 4.

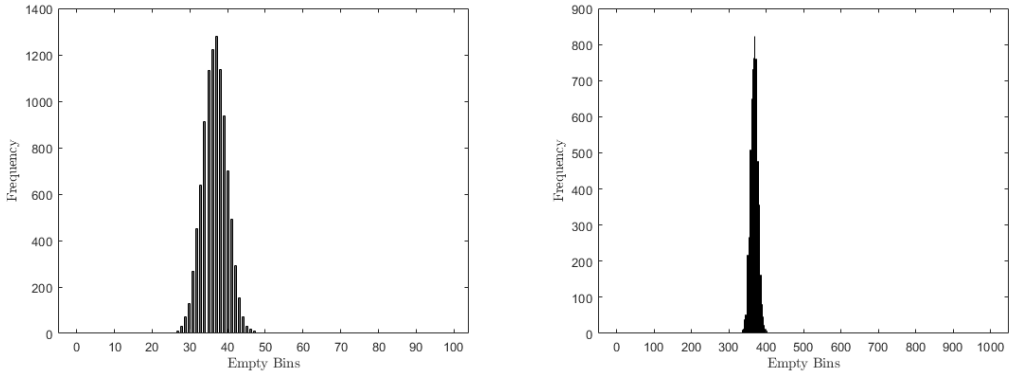

Figure 3: Left : $n = 100$. Right : $n = 1000$.

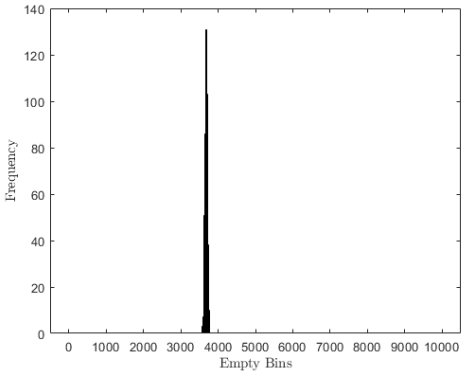

Figure 4: Histogram for $n = 10000$.