

## Elevator Problem

Amro Asali

**Back story :** i live at the 2nd floor of a building and i was wondering if i should always be the last one to enter the elevator because it's almost surely i will be the first one to exit the elevator

**Problem :** at the 0th floor of a building with  $M$  floors ,  $K > 1$  tenants each one wants to know where to stand in the elevator which has  $N$  spaces ( 1 being closest to the door ,  $N$  being the furthest ) without knowing which floor the rest of the tenants live , such that when they want to exit the elevator at the  $i$ 'th floor minimal number of tenants will block the door .

**Algorithm 1** ( $K = N$ ) :

- Input (  $M$  ,  $K$  ,  $i$  ,  $\{j\}$  )
- If  $M=N$  stand in the  $i$ 'th place
- If  $M>N$  :
  - Let  $1 \leq j \leq N$  Calculate :
  - // unnormalized probability for "j" being the optimal position
  - //  $\rightarrow (j-1)$  tenants live below or at  $i$  and  $(K-1-(j-1))=K-j$  live
  - // above or at  $i$
  - $$P(Q = j) = \left[ \left(\frac{i}{M}\right)^{j-1} \left(\frac{M-i+1}{M}\right)^{K-j} (K-1 \text{ choose } j-1) \right]$$
  - Choose  $1 \leq J \leq N$  where  $P(Q = J)$  is the largest .

**Algorithm 0** ( $K=N$ ) :

- If  $i \leq N$  :
  - If  $i$ 'th place is available stand there else :
    - Stand at the closest available place from either side
- Else:
  - Use *algorithm\_1* to find where to stand

### Algorithm approx sort (K=N) :

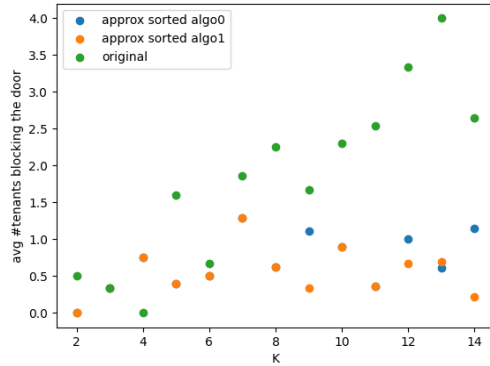
// let all K tenants run algorithm 1 now the problem is equivalent to sorting an array of  
// size N where each tenant is represented by their floor number , the solution would be

// for  $T_k$  in  $[T_1, \dots, T_K]$  find “J”

- For  $T_k$  in  $[T_1, \dots, T_K]$  :
  - $J_k = \text{Algorithm\_1}(M, K - (k - 1), \{j\} \setminus \{J_1, \dots, J_{k-1}\})$
- Return  $[J_1, \dots, J_K]$

### Results

average number of tenants blocking the door comparison as a function of k



average number of tenants blocking the door comparison as a function of k

