# 9_ses_sim.py

## 1. For all bots which are currently at a node & idle

Update:

- true idleness of all nodes in graph (true) = +Δt
- Store all the true idleness values at this time stamp
- Expected idleness of nodes at which bots are currently present is calculated by performing simple exponential smoothing (ses) of true idleness for a particular edge (now, expected idleness is function of edge not node)
- We have chosen last 5 values at any instant and $\alpha = 0.4$ for SES.

**Calculate:**

- Value function all edges where bots are present $(Q)$ - = $\frac{(expect - true)}{expect}$

- Softmax of Value function = value_exp = $\frac{e^{Q_i}}{\sum_{j=i}^{k} e^{Q_j}}$ (summation over all edges)

Set:

- True idleness of nodes where bots are present = 0

**OBSERVATION model**: bot will calculate the expected idleness as an average of all the past true idleness it has seen when it last visited the node while travelling **along that particular edge**.

## 2. For a bot deciding the next node to visit

Set:

- True idleness of the node where the bot is present = 0

**Decision Making:** here, we chose ε=0.1

- With $(1 - \varepsilon)$probability, check all neighbours and visit the one with highest value of = $[expected\ idleness]\ x\ [value\_exp]$

- With ε probability, go to a random node

----------------------------------------END-------------------------------------------------

DOUBTS:

- Here, there is no need of expected idleness to depend on edge right?
  It is just a function of time, bot and current node.
- Is it correct to initialize the value function to zero?

MAJOR DRAWBACK:

- The decision making of next node is$[expected\ idleness]\ x\ [value_exp]$
  So when expect=true; implies no other bot has visited this node yet, this bot also has no incentive to visit this node and hence, true idleness can go to very high values.