Patrick Emedom-Nnamdi

Pseudocode

Simulation of a Single Server Queue with Redispersion

**Initialization**

(Note: m is number of customers who enter queue

* n is number of customers who are served (same as number in system))

During this period set the following variable to their starting values:

* no.in.system, virtual.wait, avg.no.in.system, n, m, time, and avgBeta SET TO 0

Create vectors to store the following:

* departure times, redispersion at each event, arrival times of the remaining customers, number of customers at each event, the load at each event, and the time at each event

Set the parameters of the system:

* alpha prime, mu, target number of customers, warmup number of customers, and delta

**Thinning Algorithm**

**Warm Up**

After the initialization, start the warm up period

* create the first arrival
  + at this moment the current rate is equal to alpha prime
* generate the first interarrival time (next.arr)
  + using an exponential random number with the current rate as the parameter

While the number of customers served is less than the warm up number of customers:

* separate the simulation into three events: arrival when system is empty, arrival when system is not empty, and a departure
  + Arrival when system is empty (if the length of the scheduled departures is 0)
    1. Update time (new.time = next.arr) and number of
    2. Update m (an index for people who enter queue) by 1
    3. Calculate the average number of customers in system
    4. Generate next service time for new arrival
    5. Update the virtual wait (virtual wait = next service time)
    6. Place future departure time for this customer in vector of departures
    7. Generate new arrival rate based on alpha prime and redispersion
    8. Generate arrival time of the next customer using thinning algorithm
    9. Store statistics: customer arrival time, arrival time of customers remaining in queue, virtual wait of customer, redispersion rate of customer, next arrival rate, number of customers in system, and current load
  + Arrival when system is not empty (if next.arr < first scheduled departure time)
    1. Update time (new.time = next.arr) and number of
    2. Update m (an index for people who enter queue) by 1
    3. Calculate the average number of customers in system
    4. Generate next service time for new arrival
    5. Update the virtual wait (virtual wait = next service time)
    6. Place future departure time for this customer in vector of departures
    7. Generate new arrival rate based on alpha prime and redispersion
    8. Generate arrival time of the next customer using thinning algorithm
    9. Store statistics: customer arrival time, arrival time of customers remaining in queue, virtual wait of customer, redispersion rate of customer, next arrival rate, number of customers in system, and current load
  + A departure (when first 2 events don’t occur)
    1. Update time (new time becomes time of first departure)
    2. Update virtual wait
    3. Update number of people in system (subtract by 1)
    4. Calculate the average number of customers in system
    5. Update number of people served (add by 1)
    6. Remove first entry from customer’s remaining in queue
    7. Remove first entry from departure vector
    8. Store statistics: number of customers in system and load
* At the end of the conditional statement, UPDATE time (time = new.time)

**Simulation Start**

At the end of the warm up period:

* Set average number in system to 0

Begin simulation – REPEAT STEPS IN WARMUP