Problem 1:

1. Description

Simulate a series of N M/M/1 queues. The output of queue i becomes the input of queue (i+1). Plot a graph of N versus run-time to show that simulation time can go up with N.

2. Event handlers

There are still two types of events: arrival event and departure event. But since there are N queues, every time we try to pull an event out of the calendar, we need to choose the one with smallest time stamp among N*2 events, therefore the simulation time can go up with N.

The arrival event handler and departure event handler are basically the same as for the single M/M/1 queue.

(1) Arrival event handler for queue i

```
If (server is IDLE)
Then
    Change the server status into BUSY;
    Schedule the next type1 customer departure event for queue i
Else
Then
    Get in queue i
/* Schedule next type1 customer arrival event */
If ( i == 0)
Then
    Schedule the next arrival event following the Poisson distribution
Else
Then
    Next arrival event for queue i happens immediately after the next departure event for queue (i – 1)
```

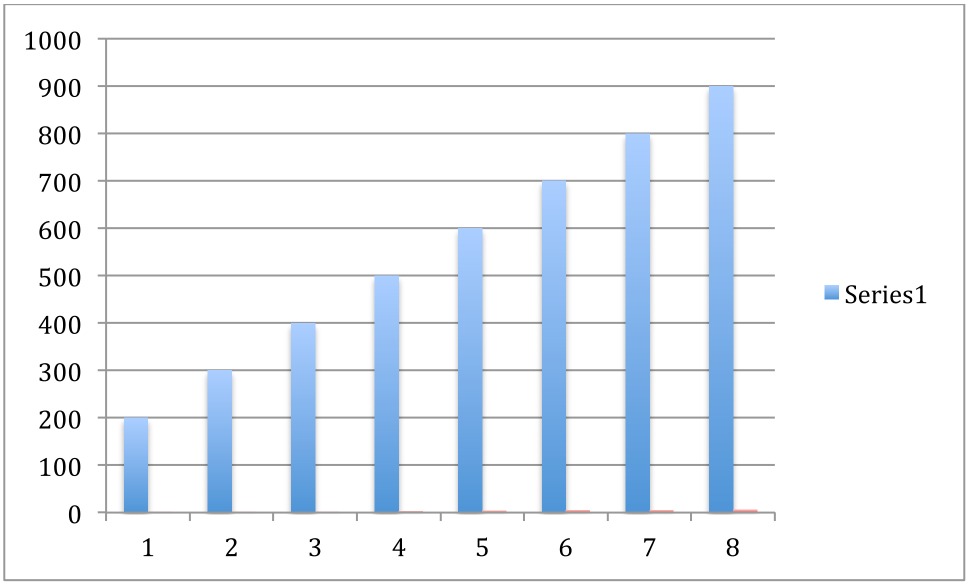
(2) Departure event handler for queue i

```
If (queue i is empty)
Then
    Change the server status into IDLE;
    Set the next departure event for queue i to be INFINITE
Else
```

```
Then
    Extract a customer from the queue i;
    Change the server status into BUSY;
    Schedule the next customer departure event
If ( i + 1 < N)
Then
    Set the next arrival time for queue (i+1) as immediately after the clock time
```

3. Simulation Results

Runtime vs N (I unit of runtime = 100usec)



Problem 2 – Part 1

1. Description

Make some variations of the previously built M/M/1 system. There are two types of customers – high priority (type 1) and low priority (type 2). Type 1 jobs preempt type 2 jobs.

2. Parameters and Values

Parameters	**Meanings**	**Values**
NTYPE1	The number of type1 customer per run	1000
NTYPE2	The number of type2 customer per run	1000

EXPARRIV1	The expected interarrival time of type1	10~20 (fix EXPARRIV2=10 while vary EXPARRIV1 from 11~20)
EXPARRIV2	The expected interarrival time of type2	10~20 (fix EXPARRIV1=10 while vary EXPARRIV2 from 11~20)
EXPSERVICE	The expected service time	5

3. Event handlers

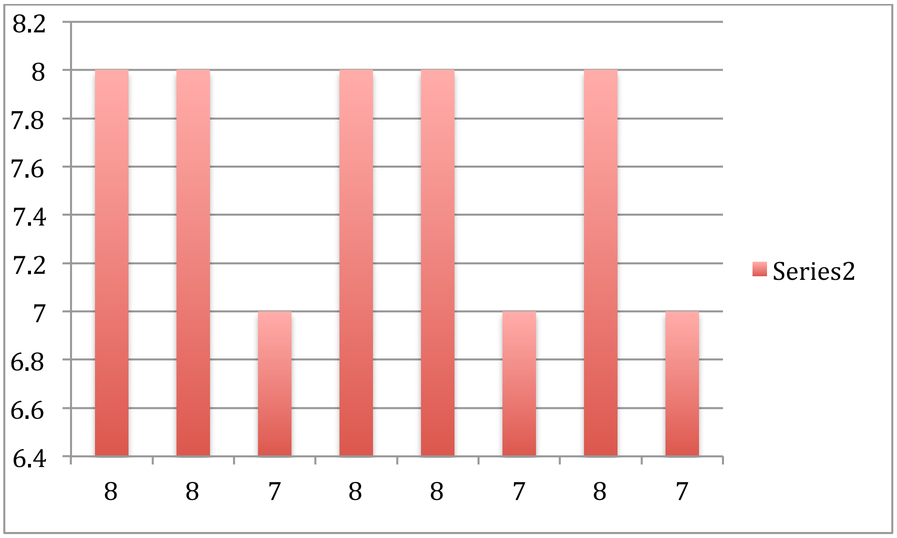
The simulation process is similar to the previous system: while the simulation is not terminated, system clock move to the earliest event in the calendar and invoke the event handler.

The difference is that in this problem there are four events: arrival event of type1 customer, departure event (end-of-service event) of type1 customer, arrival event of type2 customer, departure event of type2 customer.

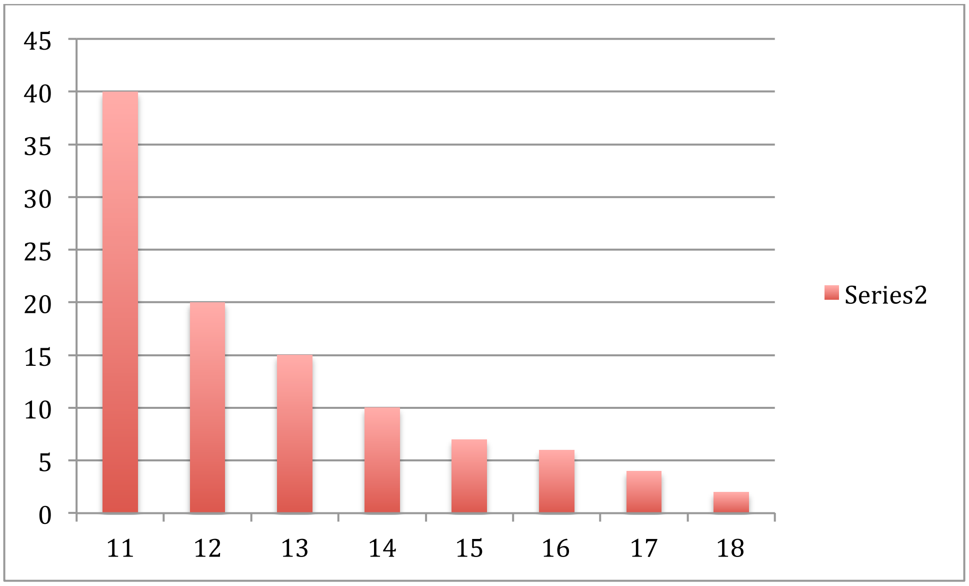
4. Simulation Results

1) Fix ExpArrival1 =10 and vary ExpArrival2

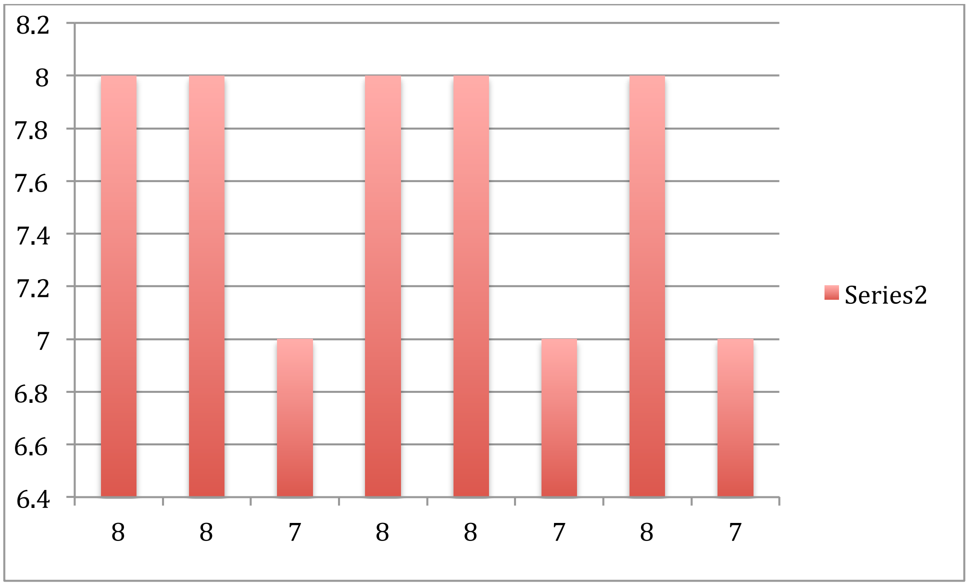
Expected Arrival vs Average number of Type1 Job (one unit = 0.1)



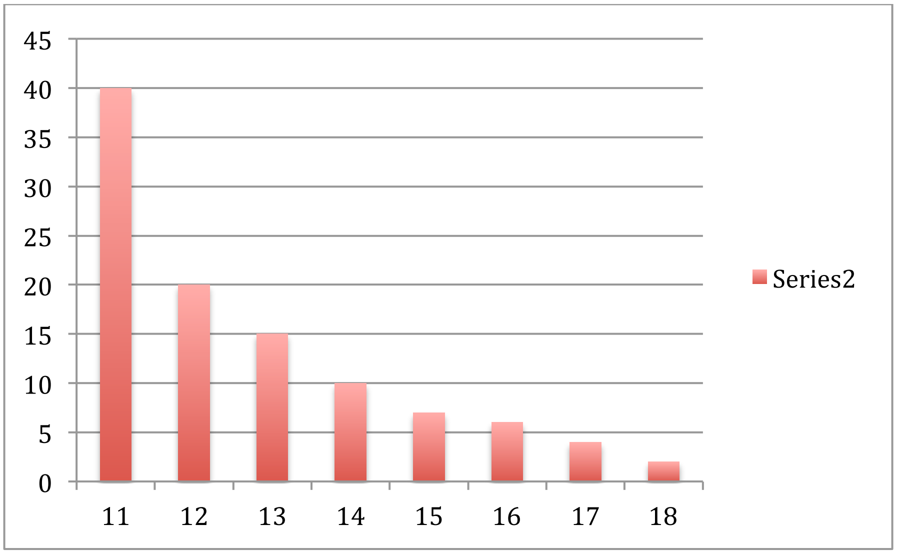
Expected Arrival vs Average number of Type 2 Jobs (One unit =1)



Expected arrival vs avg time spent by type 1 job (one unit =1)

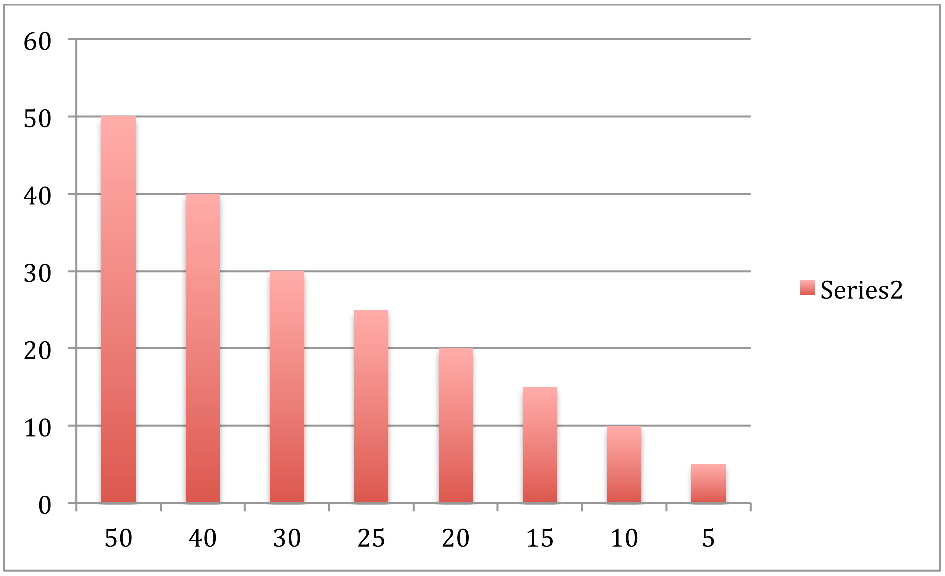


Expected arrival vs Average time spent by type 2 (one unit =5)

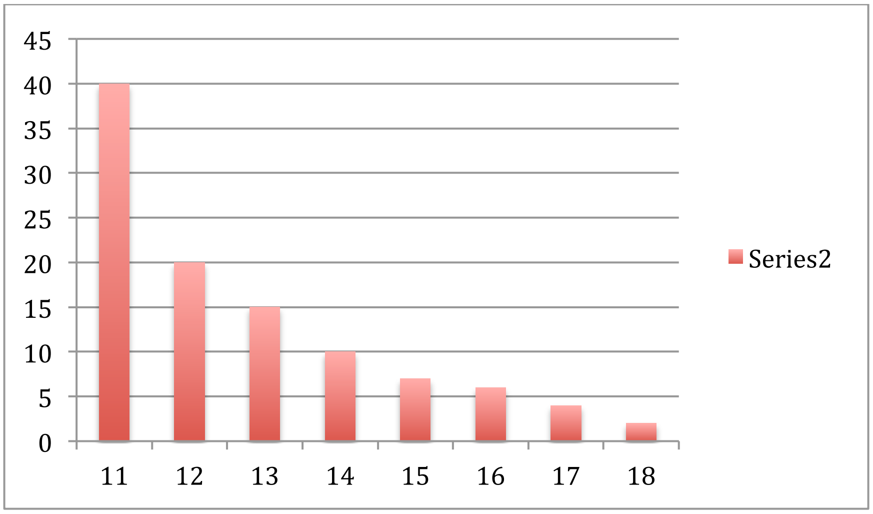


2) Fix ExpArrival2 =10 and vary ExpArrival1

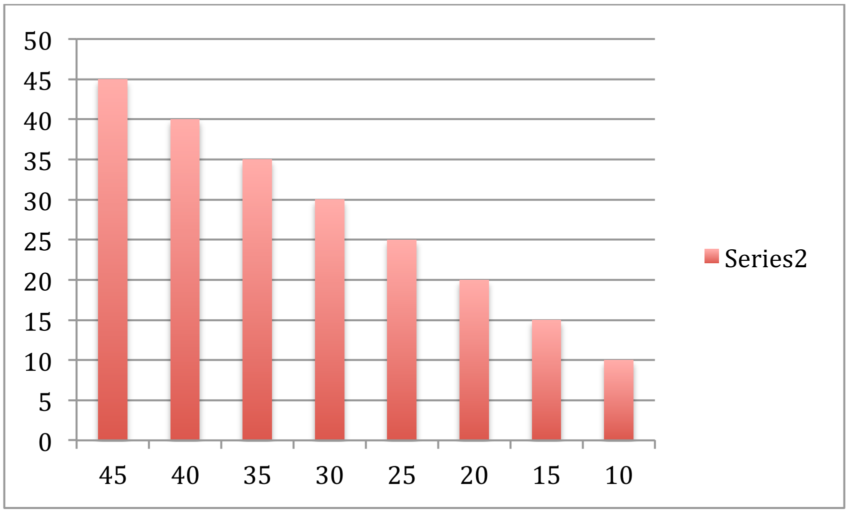
Exparriv1 vs Avg number of Jobs(1 unit = 0.1)



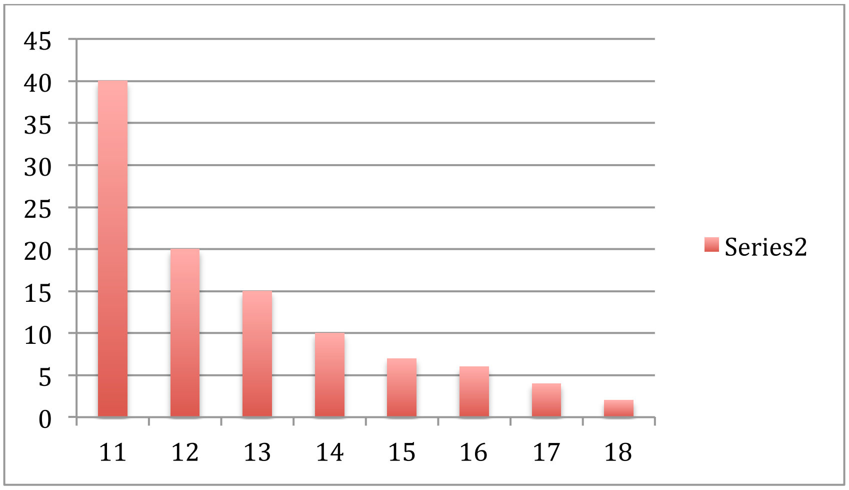
ExpArrival1 vs avg number of Type 2 job



Exparrival1 vs avg time spent by type1



Exparrival1 vs avg time spent by type2



Problem 2 – Part 2

1. Description

Make some variations of the previously built M/M/1 system. There are two types of customers – high priority (type 1) and low priority (type 2). Type 1 jobs do not preempt type 2 jobs.

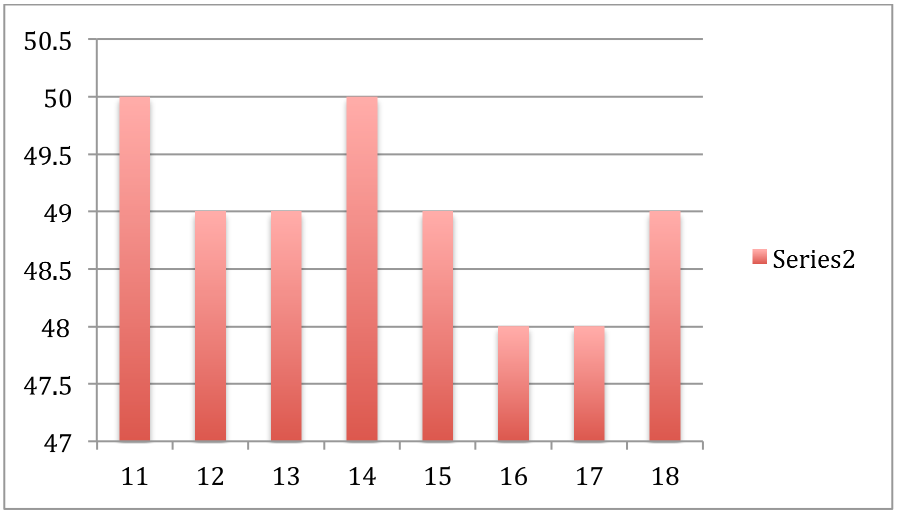
2. Parameters and Values

Same as Part 1.

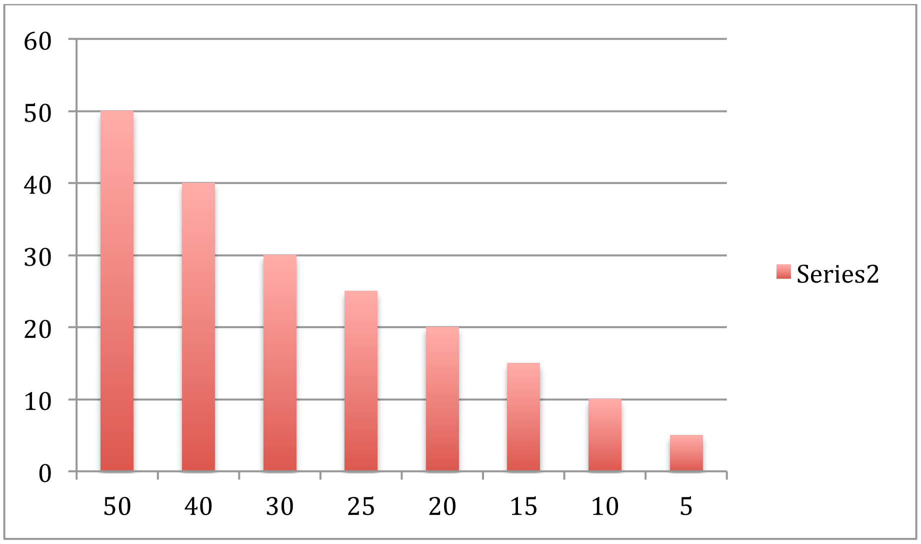
3. Simulation Results

1) Fix ExpArrival1 =10 and vary ExpArrival2

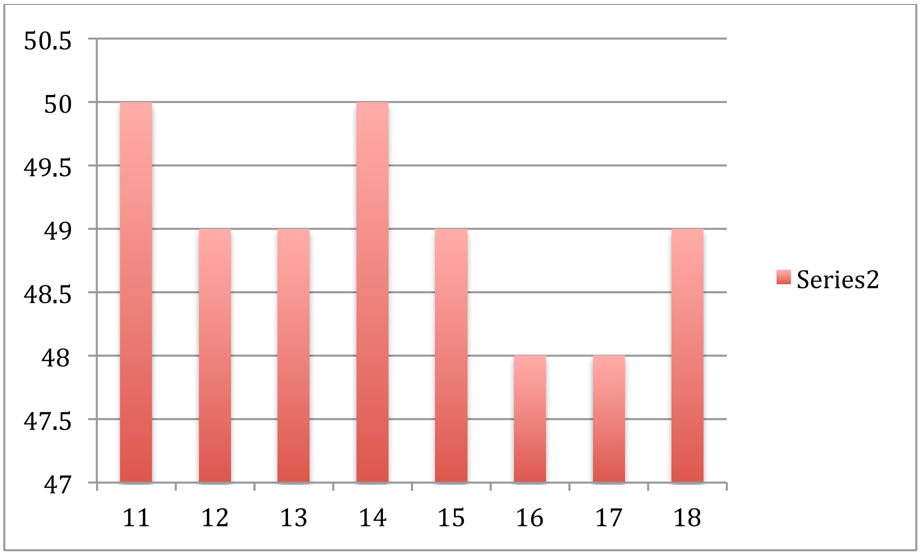
Exparrival2 vs avg number of type1 jobs



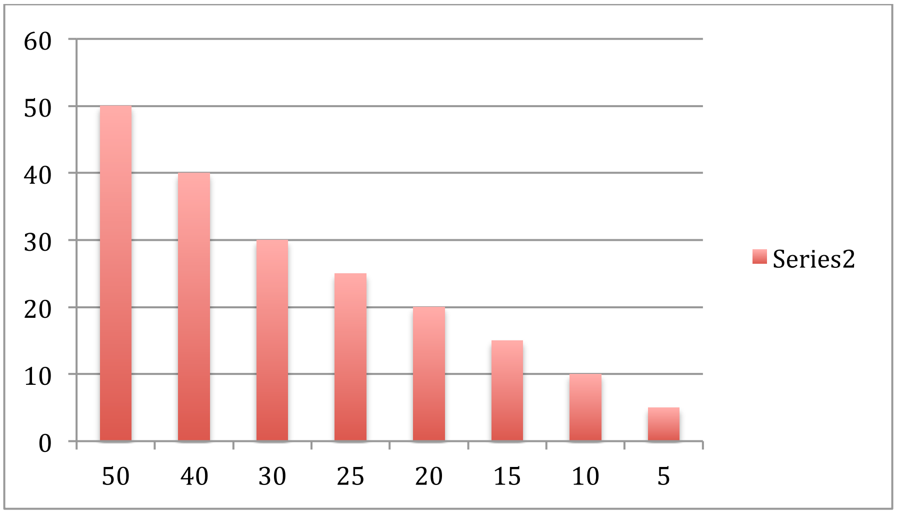
Exparrival2 vs Avg number of jobs in type2



Exparrival2 vs Avg time spent by type1

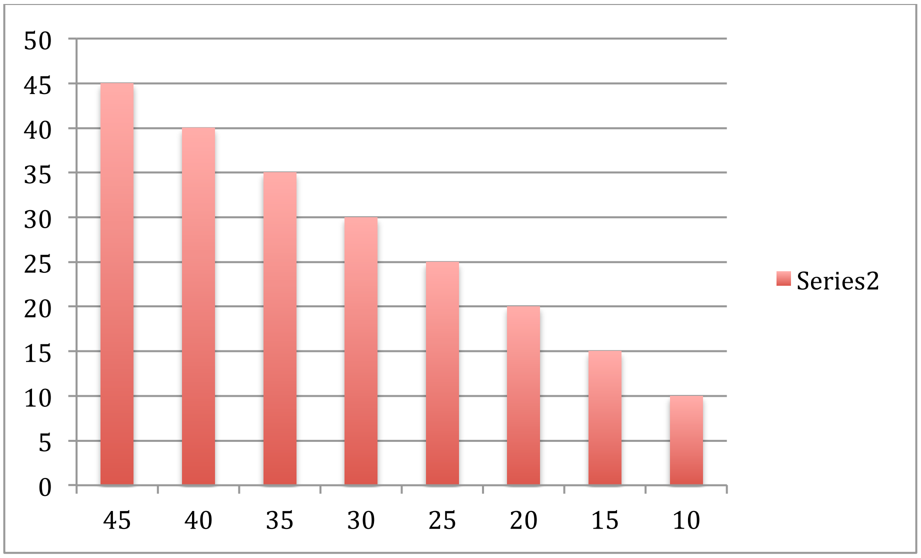


Exparrival2 vs Avg time spent by type2

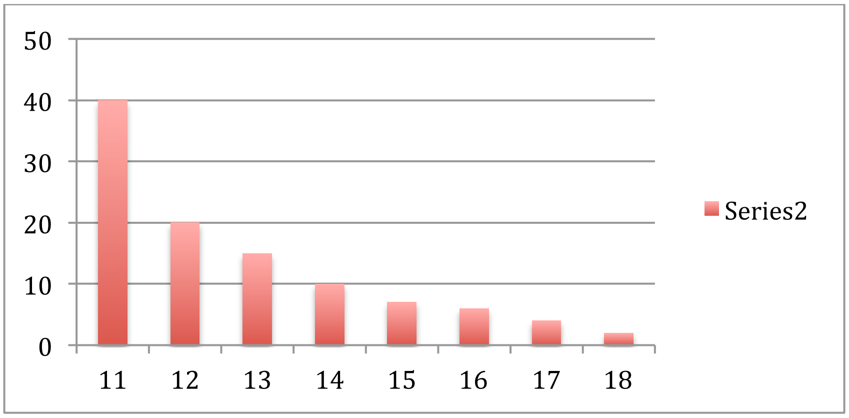


2) Fix exparrival2=10 and vary Exparrival1

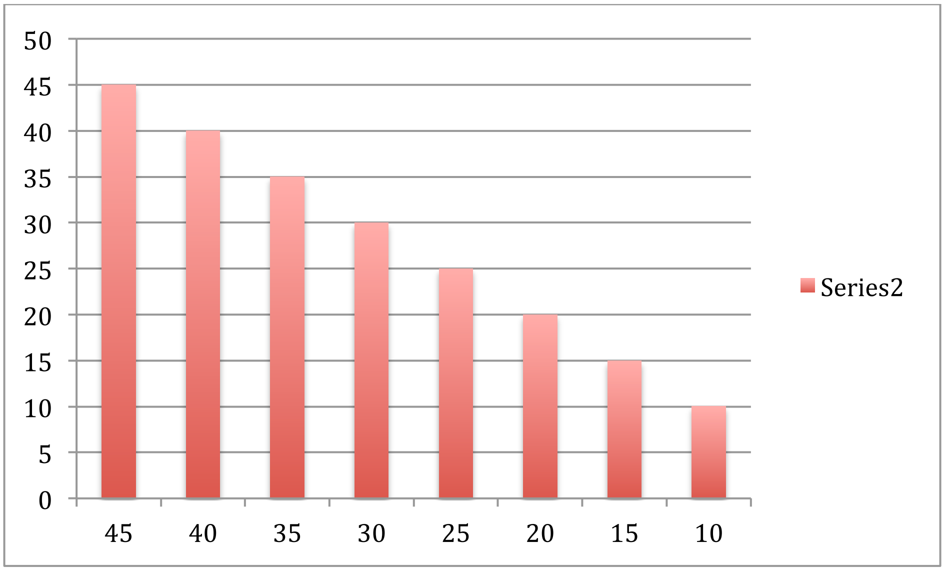
Exparrival1 vs avg number of jobs in type1



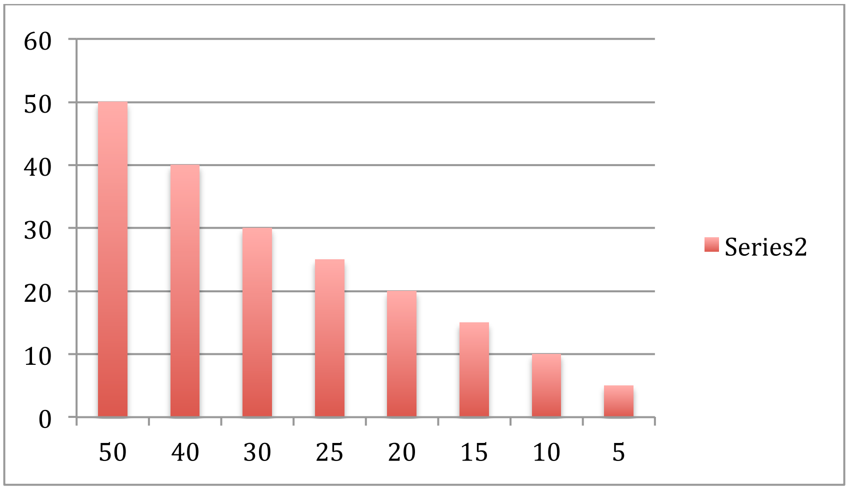
Exparrival1 vs avg number of jobs in Type2



Exparrival1 vs avg time spent by type1



Exparrival1 vs avg time spent by type2



Conclusion

In Part 1, type1 jobs can preempt type2 jobs. The average number of type1 jobs in the system does not change with the arrival rate of type2 jobs, but decreases as the arrival rate of type1 decreases. The average number of type2 jobs decreases either when the arrival rate of type1 or type2 jobs decreases. Similarly, the time spent by the type1 jobs in the system does not change with the arrival rate of type2 jobs, but decreases as the arrival rate of type1 decreases. The average time type2 jobs spent in the system decreases either when the arrival rate of type1 or type2 jobs decreases.

In Part2, type1 jobs cannot preempt type2 jobs. The differences between Part 2 and Part 1 are that the average number and the time spent in the system of type1 jobs decreases a little as the arrival rate of type2 decreases; also the number and time are larger than in Part 1. For type2 jobs, the average number and time are smaller than in Part 1.

Problem 3

Using GA to solve the following question:

You are at the airport waiting for your international flight, and suddenly find out that you have a $100 local store gift card that will expire at midnight tonight. Fortunately, they own one of their stores at the airport, so you decide to put the card to good use. At the airport store, the gift card is restricted to use only on the following items:

Item No.	Cost ($)	Weight (kg)
1	43	22
2	11	10
3	18	23
4	15	16
5	9	8
6	13	6
7	73	12
8	46	22
9	42	34
10	32	12
11	27	21
12	24	34

Since you are about to get on the plane, and the airline has a strict limitation on baggage weight which is 25 kilograms per person, you total purchase must not exceed 25 kg. Which items should you buy to make the best of the gift card while not exceeding the weight limit?

1. The Geneotype

Define the genes array, population size is 30, each has 12 items to choose from:

2. Initialize the genes array

Each gene has equal chance to be 0 or 1. If the gene has a value of 1, the corresponding item is chosen, so put it in the shopping cart; if the current gene being looked at has a value of 0, leave the item on the shelf, we are not buying it. overprice() and overweight() check if the overall price exceeds the gift card amount or overall weight exceeds the limit; if either condition happens, we initialize the gene array again.

3. Main process

We randomly pick 2 gene arrays from the total population, and compare the total price of them. The winner is the one with higher price since we are trying to make best use of the gift card. With given recombination rate, copy from Winner to Loser (overwrite); with given mutation rate, mutate that locus of the Loser. So only the loser gets to change. One thing worth noticing is that the loser might change into some gene array exceeds the price and weight limit; if, so we recombine and mutate the loser again.

4. Output

After the algorithm is done, find the member with highest price among the total population and output the result which is as following:

Item Number	Cost	Weight
6	13	6
7	73	12

```
Your Sum Purchase costs 86 Dollars, Weights 18Kg.
```