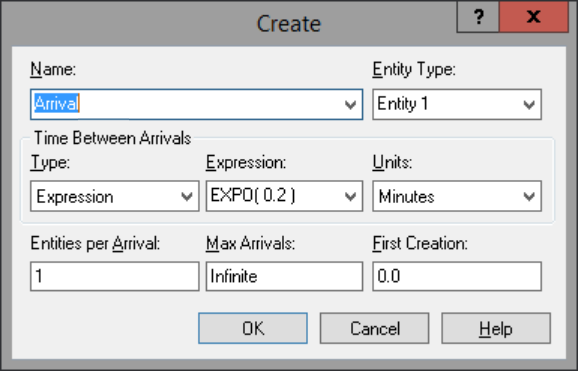
**Question 13.2**

In this problem you, can simulate a simplified airport security system at a busy airport. Passengers arrive according to a Poisson distribution with λ1 = 5 per minute (i.e., mean interarrival rate μ1 = 0.2 minutes) to the ID/boarding-pass check queue, where there are several servers who each have exponential service time with mean rate μ2 = 0.75 minutes. [Hint: model them as one block that has more than one resource.] After that, the passengers are assigned to the shortest of the several personal-check queues, where they go through the personal scanner (time is uniformly distributed between 0.5 minutes and 1 minute).

Use the Arena software (PC users) or Python with SimPy (PC or Mac users)to build a simulation of the system, and then vary the number of ID/boarding-pss checkers and personal-check queues to determine how many are needed to keep average wait times below 15 minutes. [If you’re using SimPy, or if you have access to a non-student version of Arena, you can use λ1 = 50 to simulate a busier airport.]

We use Arena to construct our model.

First , we set up arrival:



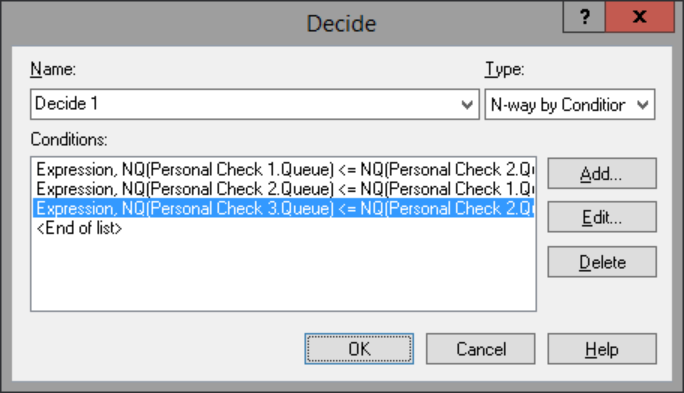
Since passengers arrive according to a Poisson distribution with λ1 = 5 per minute, the time between arrivals shall follow the exponential distribution with mean=0.2

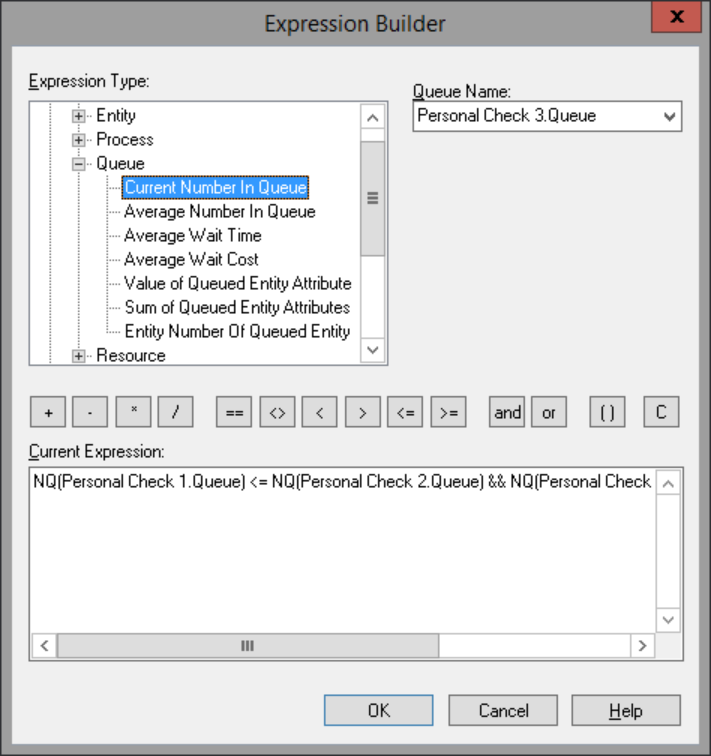
# 

Next is the ID check process. We have several servers who each have exponential service time with mean rate = 0.75 minutes. We can change the number of servers in the capacity of the resource:

# 

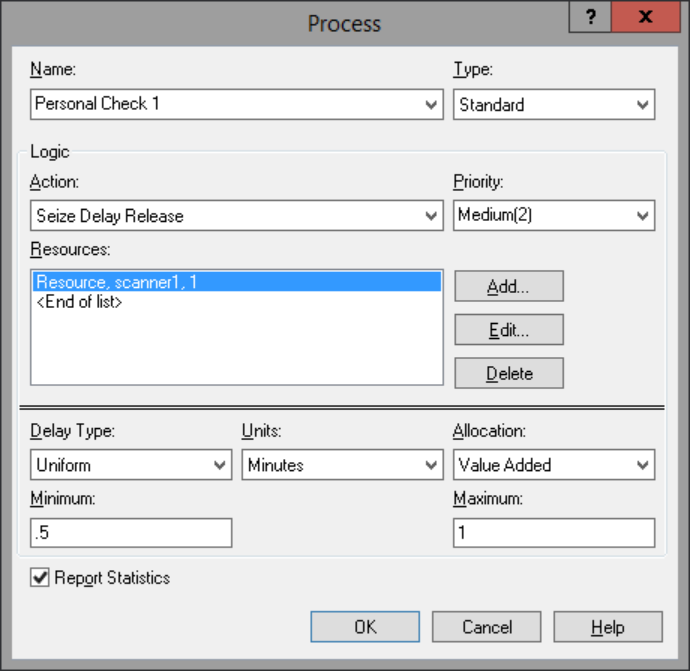
After that, the passengers are assigned to the shortest of the several personal-check queues. In this case, we should add a decide process, to find out which queue is the shortest. We set up 3 scanners first, and the decide and Personal check blocks looks like these:



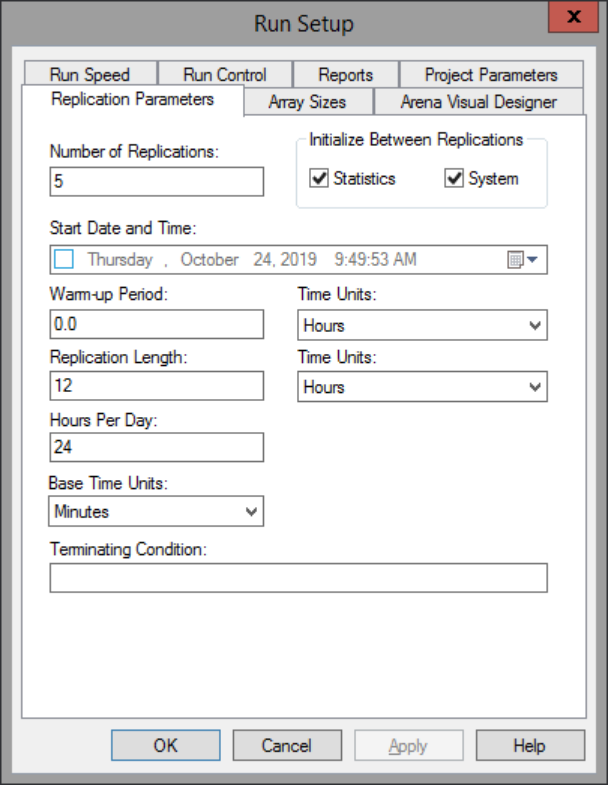


Our model looks like this:

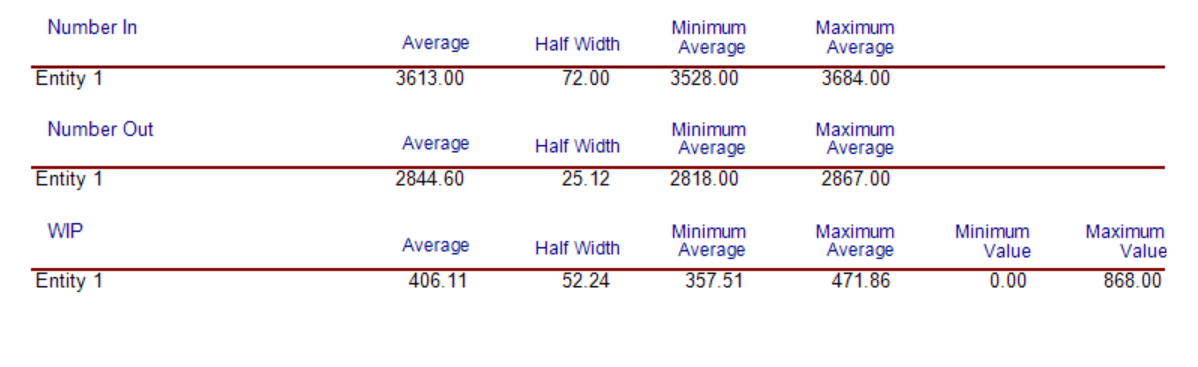
# 



At first, passengers arrive according to a Poisson distribution with λ1 = 5 per minute. We try 3 servers at the ID check and 3 scanners at the personal check process, we first set up some variables to detect the average waiting time in each queue. We set up the running hours to 12 hours, and number of replications 5. Then we run the model:

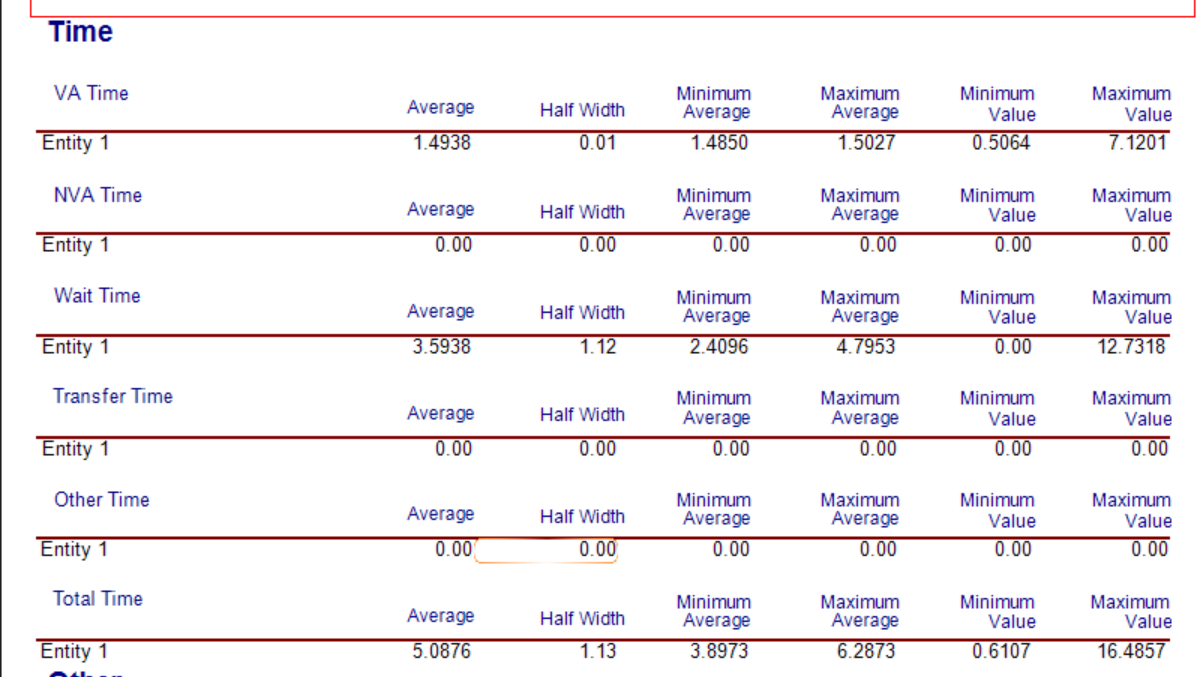


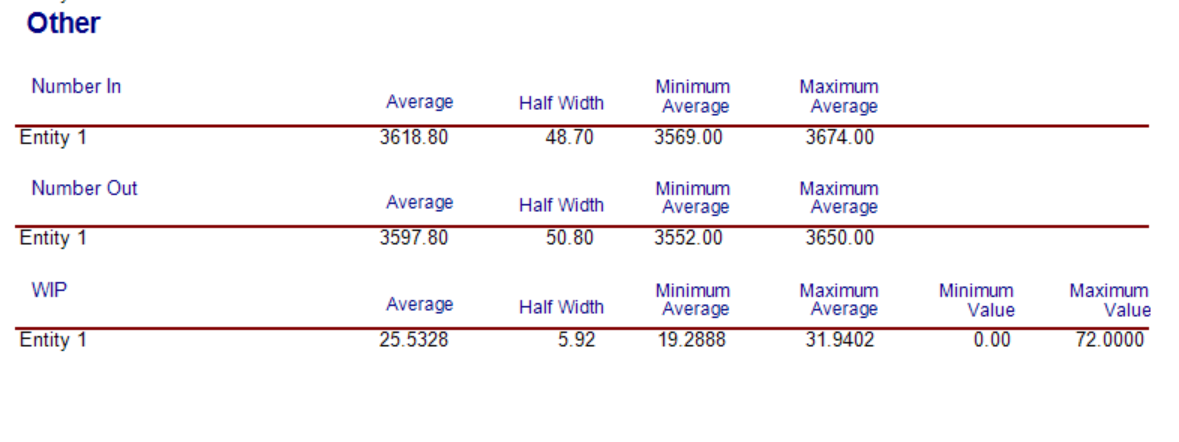




# 

The waiting time and number of waiting at the ID checking process looks incredibly long, which is 81.75 minutes. So we add the capacity of resources to 4, and the number of scanners to 4:



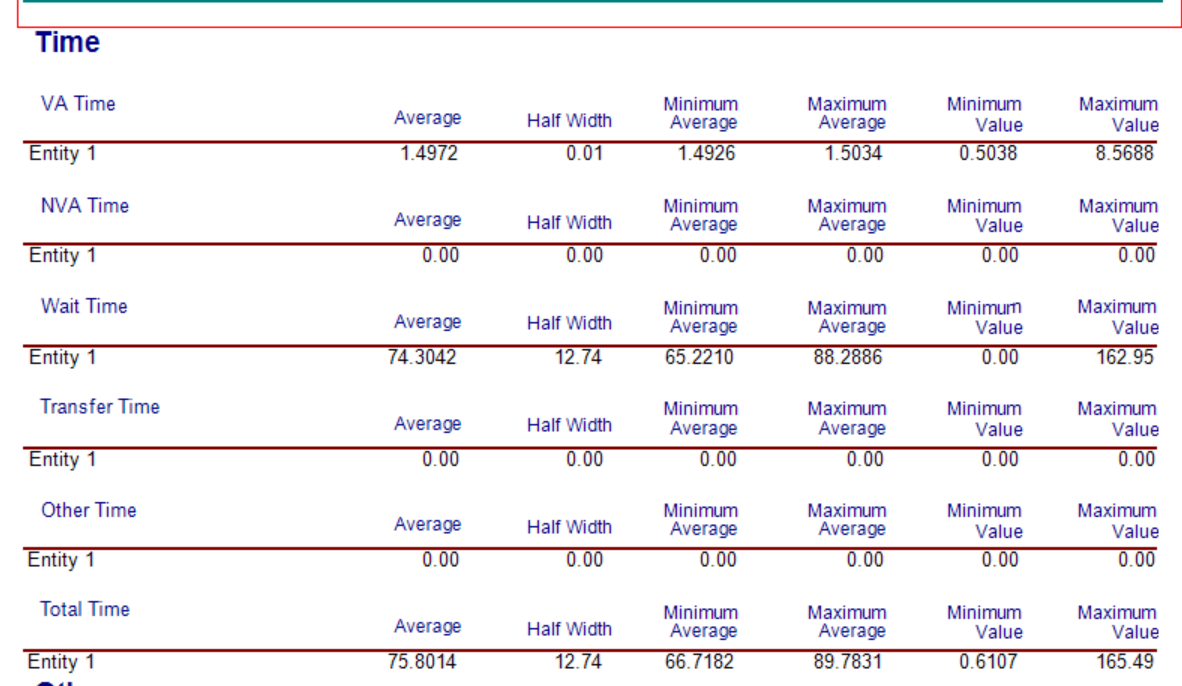


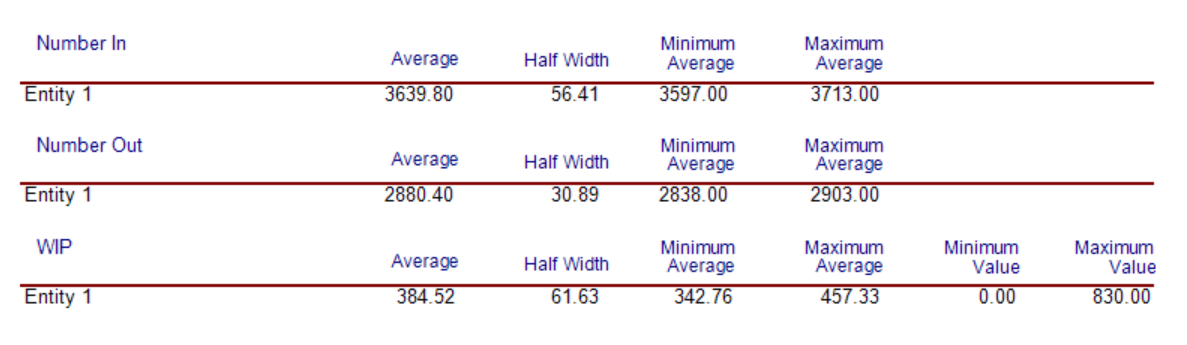
# 

Now the average waiting time looks much more acceptable(<=15 minutes), which is 5.09 minutes. At the ID check process, the average waiting time is about 0.3 minutes and at each scanner, the average waiting time is about 2.5 minutes. Meanwhile, we have 4 ID check servers and 4 scanners.

If we change ether servers to 3 or scanners to 3, we can see the result below:

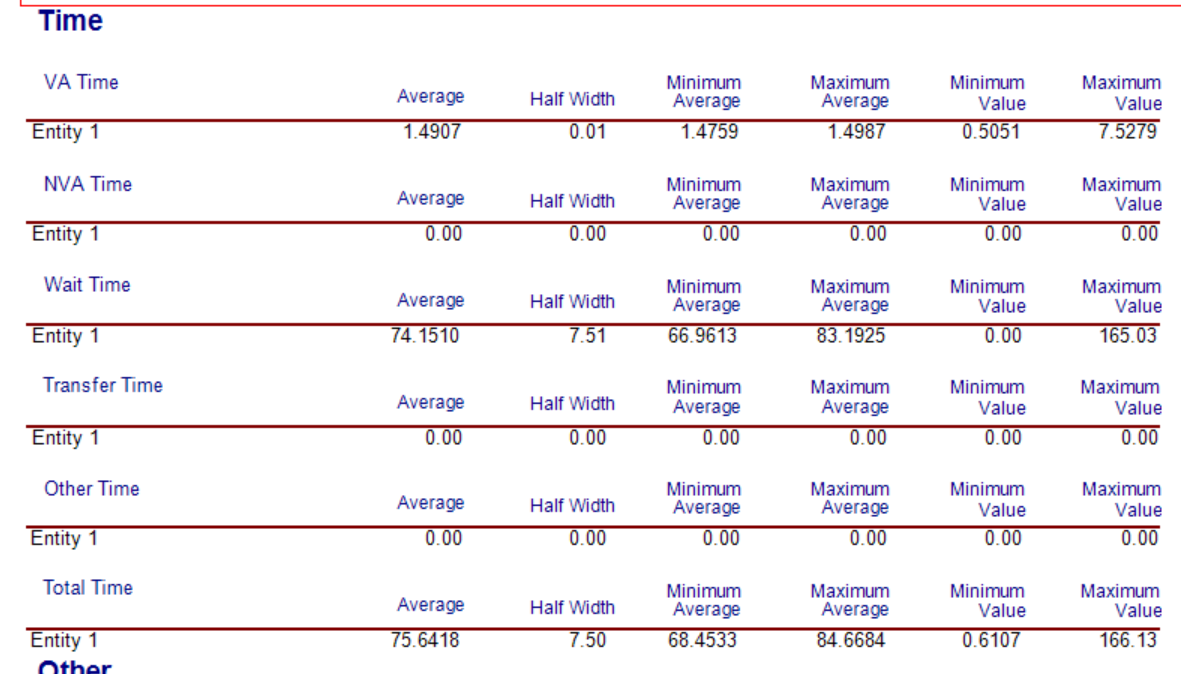
3+4:

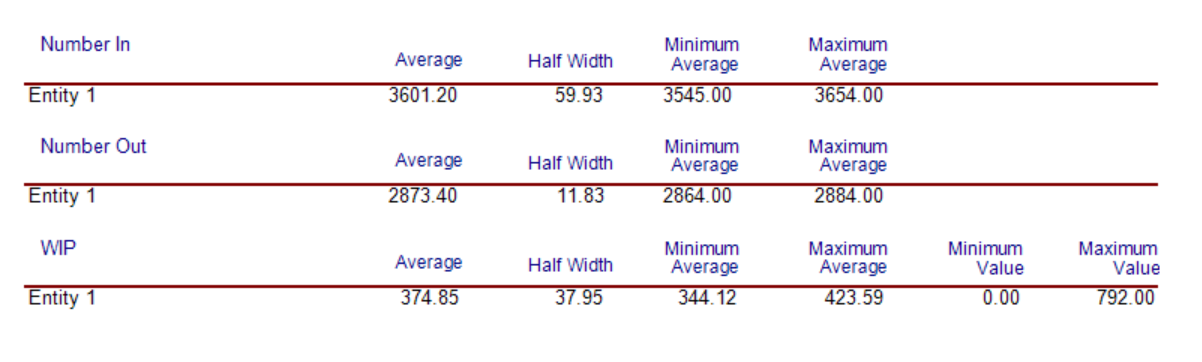


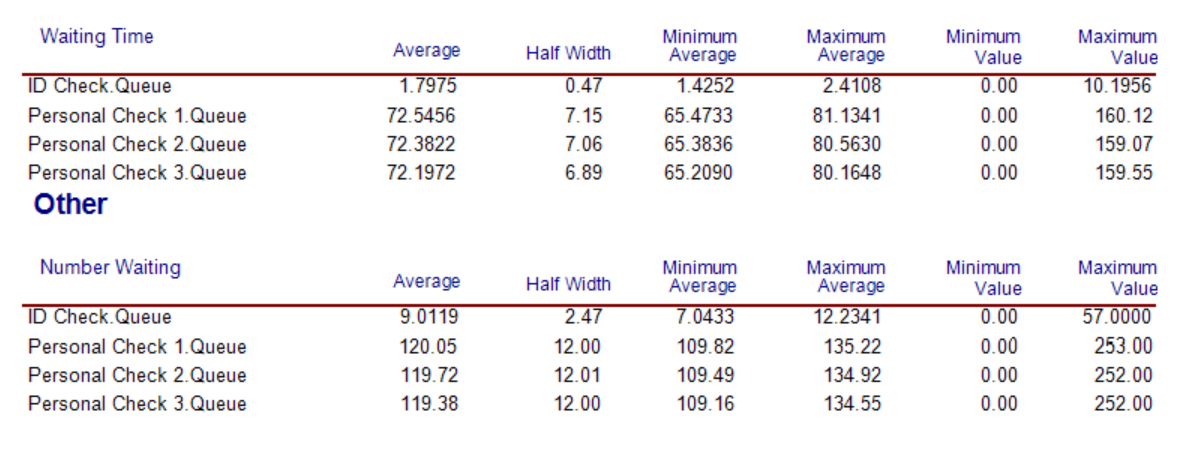


# 

4+3







3+4: the total time is 75.80 minutes.

4+3: the total time is 75.64 minutes.

They all look infeasible.

Now we can draw a conclusion that we need at least 4 ID check servers and 4 scanners to make the average waiting time less than 15 minutes.