Exercise 4

Write a discrete event simulation program for a blocking system, i.e. a system with m service units and no waiting room. The offered traffic A is the product of the mean arrival rate and the mean service time

1

The arrival process is modelled as a Poisson process. Report the fraction of blocked customers, and a confidence interval for this fraction. Choose the service time distribution as exponential. Parameters: m = 10, mean service time = 8 time units, mean time between customers = 1 time unit (corresponding to an offered traffic of 8 Erlang), 10 x 10.000 customers.

In [ ]:
```
import numpy as np
#import poission
import math
from scipy.stats import poisson
#import exponential
from scipy.stats import expon
import bisect
from discrete_event import Customer, main_loop, confidence_intervals, erlang_b
```

In [ ]:
```
m = 10 #number of servers
s = 8 #mean service time
lam = 1#arrival_intensity
total_customers =10000 #10*10000
A = lam*s
```

In [ ]:
```
#arrival time differences are exponentially distributed
np.random.seed(0)
arrival_interval = lambda : np.random.exponential(1/lam, size = total_customers)
service_time =lambda : expon.rvs(scale = s, size = total_customers)
```

In [ ]:
```
#Amount of people blocked in the system
blocked_1 = main_loop(arrival_interval, service_time, m)
```

In [ ]:
```
print("Blocking probability: ", blocked_1/total_customers)
print("Mean blocking probability: ", np.mean(blocked_1/total_customers))
```

```
Blocking probability:  [0.1214 0.1155 0.1193 0.1204 0.1229 0.1217 0.1233 0.1193 0.1163 0.1163]
Mean blocking probability:  0.11964000000000001
```

In [ ]:
```
#Theoretical blocking probability
print("Theoretical blocking probability",erlang_b(m, A))
```

```
Theoretical blocking probability 0.12166106425295149
```

Answer

According to the discrete event simulation, the fraction of blocked customers is 0.119 which corresponds well with the theoretical value of 0.1216.

2

The arrival process is modelled as a renewal process using the same parameters as in Part 1 when possible. Report the fraction of blocked customers, and a confidence interval for this fraction for at least the following two cases

In [ ]:
```
# (a) Experiment with Erlang distributed inter arrival times The
#Erlang distribution should have a mean of 1
inter_arrival = lambda : np.random.gamma(2, 0.5, size = total_customers)
service_time = lambda : expon.rvs(scale = s, size = total_customers)
blocked_erlang = main_loop(arrival_interval, service_time, m)
print("Blocking probability: ", blocked_erlang/total_customers)
print("Mean blocking probability: ", np.mean(blocked_erlang/total_customers))
```

```
Blocking probability:  [0.1186 0.1174 0.1172 0.1283 0.1275 0.1256 0.1194 0.1238 0.1253 0.1228]
Mean blocking probability:  0.12259
```

Answer

When the inter arrival time is Erlang distributed with mean 1 time unit, the fraction of blocked customers is 0.06712 which does not correspond with the theoretical value of 0.1216.

In [ ]:
```
# hyper exponential inter arrival times. The parameters for
#the hyper exponential distribution should be
p1 = 0.8
λ1 = 0.8333
p2 = 0.2
λ2 = 5.0
s = 8
arrival_interval = lambda : np.random.choice([expon.rvs(scale = 1/λ1), expon.rvs(scale = 1/λ2)], total_c

service_time = lambda : expon.rvs(scale = s, size = total_customers)

blocked_hyperexp = main_loop(arrival_interval,service_time, m)
print("Blocking probability: ", blocked_hyperexp/total_customers)
print("Mean blocking probability: ", np.mean(blocked_hyperexp/total_customers))
```

```
Blocking probability:  [0.9604 0.8635 0.331  0.0562 0.9571 0.5923 0.1861 0.0194 0.8323 0.4622]
Mean blocking probability:  0.52605
```

Answer For hyperexponential inter arrival time with mean 1 time unit, the fraction of blocked customers is 0.416 which does not correspond with the theoretical value of 0.1216.

3

The arrival process is again a Poisson process like in Part 1. Experiment with different service time distributions with the same mean service time and m as in Part 1 and Part 2

a)

Constant service time

In [ ]:
```
# a) Constant service time
arrival_interval = lambda : np.random.exponential(1/lam, size = total_customers)
service_time = lambda : s*np.ones(total_customers)

blocked_constant = main_loop(arrival_interval,service_time, m)
print("Blocking probability: ", blocked_constant/total_customers)
print("Mean blocking probability: ", np.mean(blocked_constant/total_customers))
```

```
Blocking probability:  [0.1208 0.1199 0.1089 0.1157 0.1213 0.127  0.1194 0.1165 0.1095 0.1274]
Mean blocking probability:  0.11864
```

Answer

When the service time is constant, the fraction of blocked customers is 0.12164 which corresponds well with the theoretical value of 0.1216.

In [ ]:
```
# Pareto distributed service times with at least k = 1.05 and
#k = 2.05.

k = 1.05
service_time = lambda : np.random.pareto(k, total_customers)
blocked_pareto_1 = main_loop(arrival_interval, service_time, m)
print("Blocking probability for k= 1.05: ", blocked_pareto_1/total_customers)
print("Mean blocking probability: ", np.mean(blocked_pareto_1/total_customers))
k = 2.05
service_time = lambda : np.random.pareto(k, total_customers)
blocked_pareto_2 = main_loop(arrival_interval, service_time, m)
print("Blocking probability for k= 2.05: ", blocked_pareto_2/total_customers)
print("Mean blocking probability: ", np.mean(blocked_pareto_2/total_customers))
```

```
Blocking probability for k= 1.05:  [0.0431 0.0626 0.0317 0.0615 0.0538 0.0954 0.0977 0.0519 0.0996 0.087
]
Mean blocking probability:  0.06843
Blocking probability for k= 2.05:  [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
Mean blocking probability:  0.0
```

Answer

When the service time is pareto distributed with k=1.05 the mean blocking fraction is 0.089 which is somewhat close to the theoretical value of 0.1216. For k=2.05 no customers are blocked.

In [ ]:
```
#absolute gaussian distributed service times with mean s and standard deviation s/4 #99%+ of the values
service_time = lambda : np.random.normal(s, s/4, size = total_customers)
blocked_gauss = main_loop(arrival_interval, service_time, m)
print("Blocking probability: ", blocked_gauss/total_customers)
print("Mean blocking probability: ", np.mean(blocked_gauss/total_customers))
```

```
Blocking probability:  [0.1251 0.1237 0.1308 0.1255 0.1286 0.1228 0.1243 0.1219 0.1203 0.1313]
Mean blocking probability:  0.12542999999999999
```

Answer

When the service time is normally distributed with mean 8 time units and standard deviation 0.5 time units, the fraction of blocked customers is 0.12198 which corresponds well with the theoretical value of 0.1216.

4

Compare confidence intervals for Parts 1, 2, and 3 then interpret and explain differences if any.

In [ ]:
```
#show confidence intervals for all the experiments
print("Confidence intervals for blocking probability")
print("Part 1: ", confidence_intervals(blocked_1/total_customers))
print("Part 2 (Erlang distribution): ", confidence_intervals(blocked_erlang/total_customers))
print("part 3 (Hyper exponential distribution): ", confidence_intervals(blocked_hyperexp/total_customers
print("Part 4 (Constant service time): ", confidence_intervals(blocked_constant/total_customers))
print("Part 5 (Pareto distribution k=1.05): ", confidence_intervals(blocked_pareto_1/total_customers))
print("Part 5 (Pareto distribution k=2.05): ", confidence_intervals(blocked_pareto_2/total_customers))
print("Part 6 (Gaussian distribution): ", confidence_intervals(blocked_gauss/total_customers))
```

```
Confidence intervals for blocking probability
Part 1:  (0.11798123311342432, 0.1212987668865757)
Part 2 (Erlang distribution):  (0.12013640407890787, 0.12504359592109215)
part 3 (Hyper exponential distribution):  (0.3094141139524663, 0.7426858860475337)
Part 4 (Constant service time):  (0.11496185750357601, 0.12231814249642398)
Part 5 (Pareto distribution k=1.05):  (0.053946513723001646, 0.08291348627699836)
Part 5 (Pareto distribution k=2.05):  (0.0, 0.0)
Part 6 (Gaussian distribution):  (0.12325253791399252, 0.12760746208600746)
```

For part 1-3 (Varying arrival intervals) the only confidence interval which includes the Erlang's B-value is the one where the inter-arrival times are exponential, since it's the only arrival process which is Poisson.