CSC 446 Simulation Analysis Report

1. Project description

   The project aims to study the common queueing models with real-life problems mapping to each in detail by computing queueing models' theoretical value of performance metrics. Using Java Modeling Tools with the help of Python to collect and visualize simulation results, and compare them with the computed expected value. Summarize the findings according to the comparison between the theoretical and simulated values.

2. The problem mapping to each of the queuing models

   For the M/M/1 queuing model, it is assumed as an automated teller machine (ATM) of a bank with infinite queuing capacity. Both service time of the ATM and the inter-arrival time of banking customers (poisson arrival) are exponentially distributed.

   M/G/1 queuing model can be thought of as an airport automatic check-in station with infinite queuing capacity. Service time is in uniform distribution and inter-arrival time of customers is exponentially distributed (poisson arrival).

   An auto repair shop can be modeled as a M/M/C/N system. Assuming C=3 and N=15, there are 3 available spots inside the shop for repairing and 12 available parking spots for customers to wait to be served. Both service time of the mechanics and the inter-arrival time of customers (poisson arrival) are exponentially distributed.

   Airport security check-in system is modeled as a network of queues. After the initial check-in with the custom, passengers are sent to different security check-in sections based on their flight types. Domestic flight passengers (70% of all overall passengers) are sent to in-person inspection line A, and international flight passengers (30% of all overall passengers) are sent to in-person inspection B. We assume all stations have infinite capacity. Both service time of the security check-in team (customs and in-person inspection teams) and the inter-arrival time of passengers (poisson arrival) are exponentially distributed.

3. Simulation goals and simulation parameters

   The purpose of the simulations is to imitate the real-life queuing model through software modeling tools and collect the simulated results when a large number of events are generated. For each model that is created in JMT, a maximum of 100,0000 samples can be collected for each simulation unless a stop event is generated. Performance indices ρ (utilization), L (number of customers in the system), L_Q (average number of customers in the queue, not an available performance index in JMT), W (system response time), W_Q (queueing time), inter-arrival time (through logger), and other relevant information in JMT. For performance indices such as W_Q cannot be collected through JMT, they will be calculated using available data (arrival rate, utilization, and etc...).

4. Basic M/M/1 system/queue model

4.1. Methodology

JMT is used to model M/M/1 queue. A source (banking customers), a queue (ATM) with infinite capacity which applies non-preemptive station scheduling and first come first serve (FCFS) policies, and a sink are used. By enabling the What-if analysis parameters, changing the arrival rate of customers from 0.1 to 0.8 in 8 steps. The service time of the ATM is in exponential distributions with a constant mean of 1. Arrival rate or mean inter-arrival rate is changed to achieve 0.1~0.8 (with 0.1 increment) system utilization. Confidence level is set to 95% and the maximum error is 0.03. Python matplotlib.pyplot is used when images cannot be generated through JMT

4.2. Theoretical values

The following formula is used to calculate theoretical values of desired performance metrics (for M/M/1 queue)

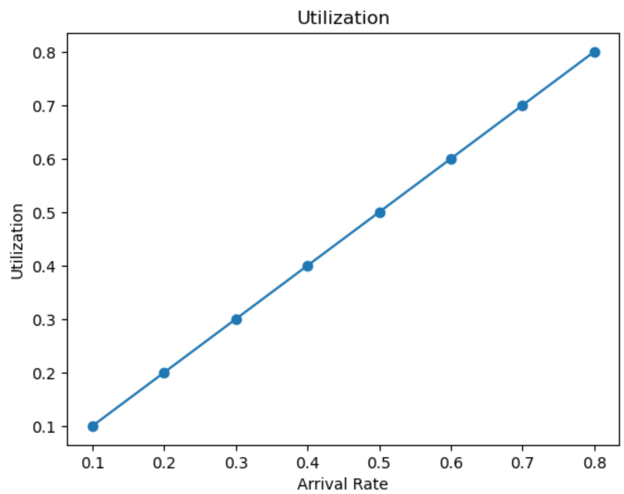
$\rho = \lambda / \mu$, utilization
$L = \rho / (1 - \rho)$, (long-run) time-average number of customers in system
$L_Q = \rho^2 / (1 - \rho)$ (long-run) time-average number of customers in queue
$W = 1 / (\mu(1 - \rho))$ (long-run) average time spent in system per customer
$W_Q = \rho / (\mu(1 - \rho))$ (long-run) average time spent in queue per customer
$P_n = (1 - \rho)\rho^n$, (steady-state) probability of having n customers in system

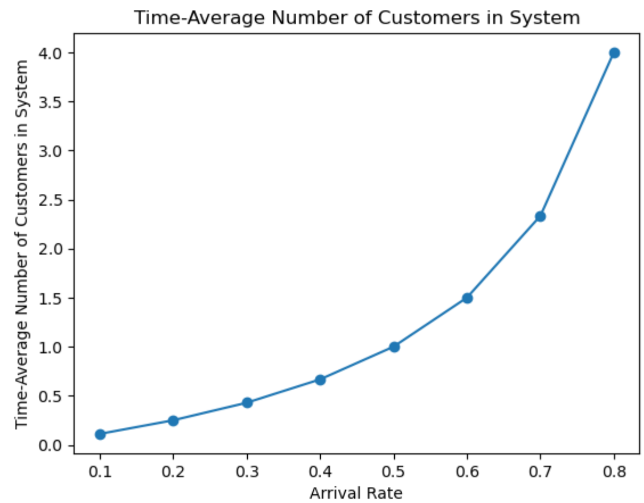
Mean service time is 1.0, so service rate $\lambda = 1 / 1.0 = 1$
Arrival rate $\mu = \lambda / \rho$ = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]
(Mean of inter-arrival time $= 1 / \mu$)

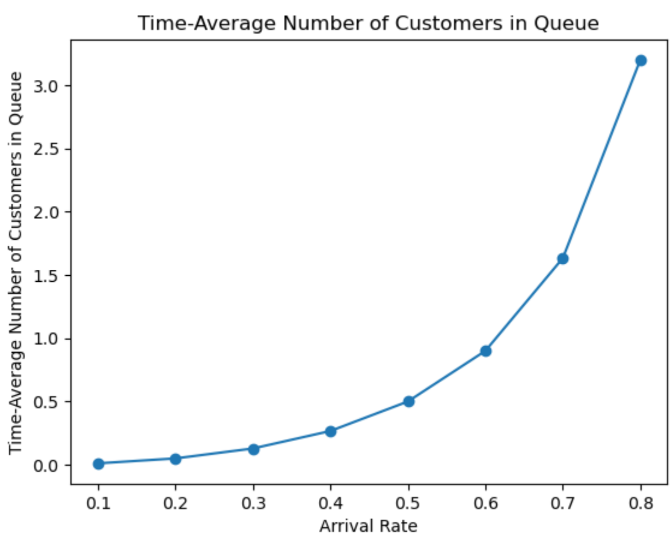
a. Performance metrics

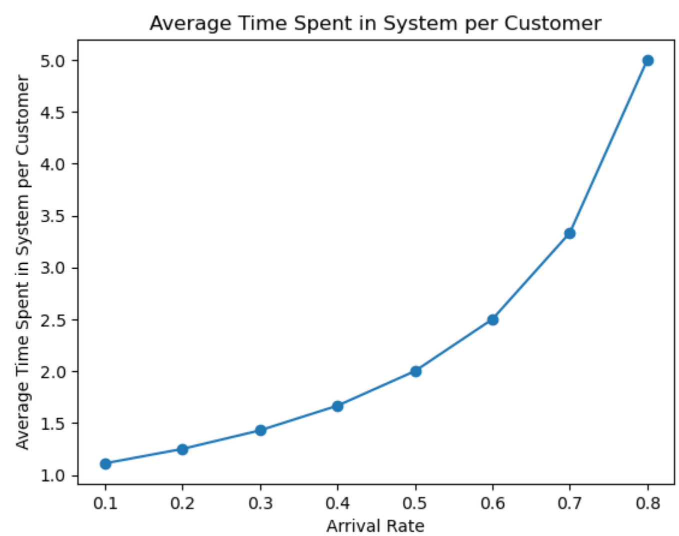
ρ	Mean of L	Mean of LQ	Mean of W	Mean of WQ
0.10	0.1111	0.0111	1.1111	0.1111
0.20	0.2500	0.0500	1.2500	0.2500
0.30	0.4286	0.1286	1.4286	0.4286
0.40	0.6667	0.2667	1.6667	0.6667

0.50	1.0000	0.5000	2.0000	1.0000
0.60	1.5000	0.9000	2.5000	1.5000
0.70	2.3333	1.6333	3.3333	2.3333
0.80	4.0000	3.2000	5.0000	4.0000



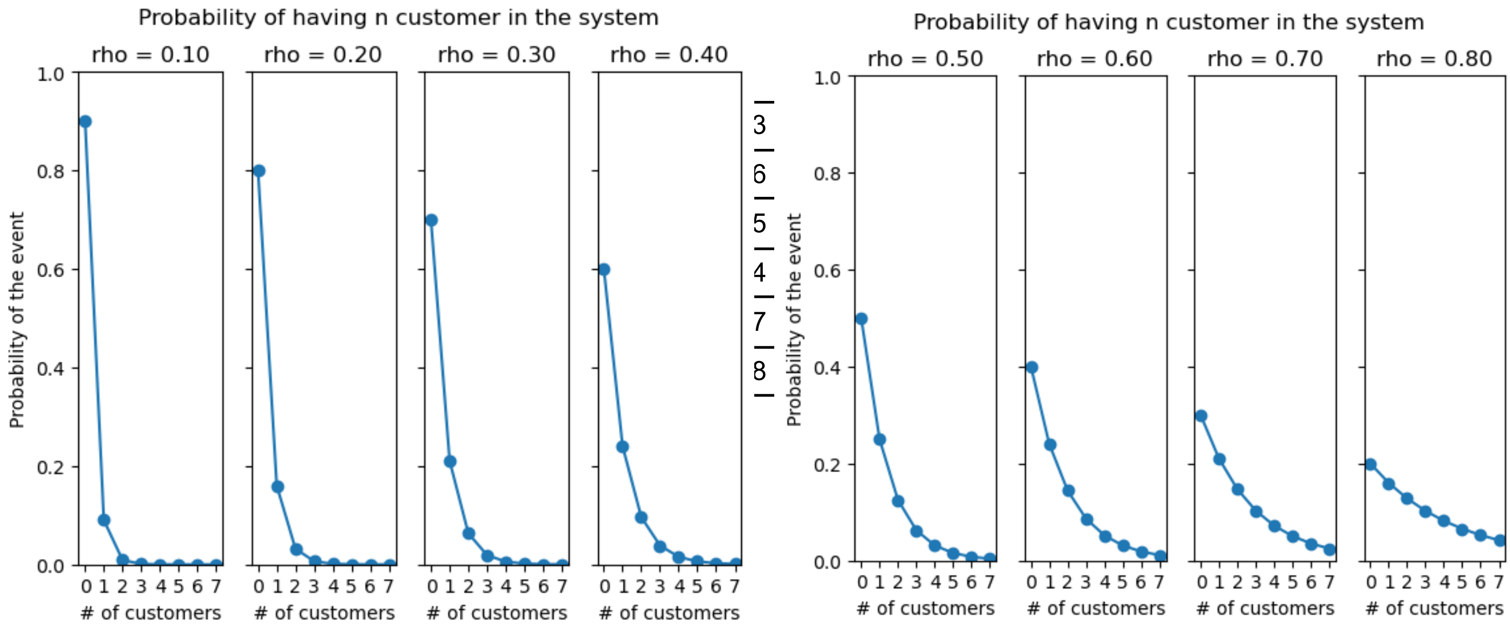






b. Probability of having n customers in the system (n=[0, 1, 2, 3, 4, 5])

ρ	P0	P1	P2	P3	P4	P5
0.10	0.9	0.09	0.009	0.0009	0.00009	0.000009
0.20	0.8	0.16	0.032	0.0064	0.00128	0.000256



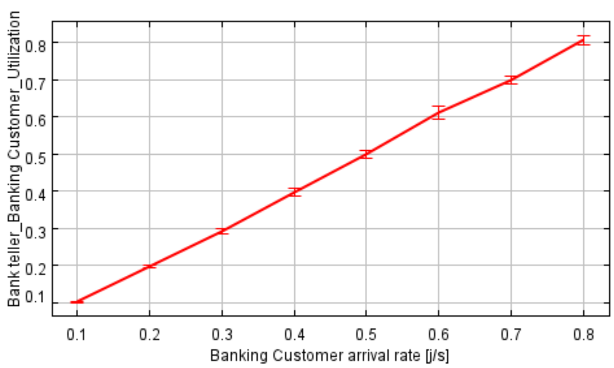
4.3. Collected simulation results

The mean of L_Q is calculated using little's law ($L_Q = \lambda W_Q$), and the variance is calculated using $\rho^2(1+\rho-\rho^2) / (1-\rho)^2$

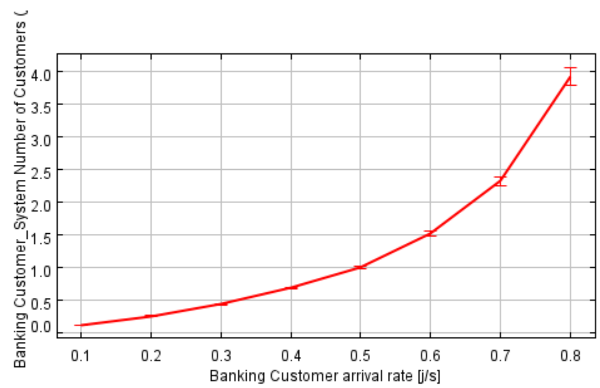
a. Performance metrics

Mean of ρ	CI of ρ	Mean of L	CI of L	Mean of LQ	CI of LQ	Mean of W	CI of W	Mean of WQ	CI of WQ
0.1005	[0.0980, 0.1031]	0.1099	[0.1072, 0.1125]	0.0115	[0.0113, 0.0117]	1.1056	[1.0820, 1.1291]	0.1154	[0.1124, 0.1184]
0.1975	[0.1927, 0.2023]	0.2497	[0.2433, 0.2562]	0.0509	[0.0504, 0.0514]	1.2384	[1.2026, 1.2742]	0.2566	[0.2495, 0.2638]
0.2911	[0.2846, 0.2976]	0.4299	[0.4183, 0.4415]	0.1277	[0.1268, 0.1286]	1.4391	[1.4039, 1.4742]	0.4320	[0.4193, 0.4446]
0.3979	[0.3869, 0.4089]	0.6806	[0.6631, 0.6981]	0.2670	[0.2656, 0.2684]	1.6837	[1.6430, 1.7245]	0.6671	[0.6496, 0.6846]
0.4988	[0.4880, 0.5097]	1.0005	[0.9775, 1.0235]	0.4978	[0.4956, 0.5000]	1.9617	[1.9035, 2.0199]	0.9931	[0.9665, 1.0197]
0.6155	[0.5936, 0.6294]	1.5209	[1.4756, 1.5662]	0.9080	[0.9045, 0.9115]	2.5090	[2.4367, 2.5814]	1.4949	[1.4576, 1.5322]
0.6993	[0.6877, 0.7109]	2.3200	[2.2573, 2.3827]	1.6442	[1.6392, 1.6492]	3.3175	[3.2281, 3.4069]	2.3305	[2.2741, 2.3868]
0.8069	[0.7941, 0.8197]	3.9261	[3.7845, 4.0676]	3.1695	[3.1607, 3.1783]	4.9463	[4.8381, 5.0545]	3.9638	[3.8637, 4.0639]

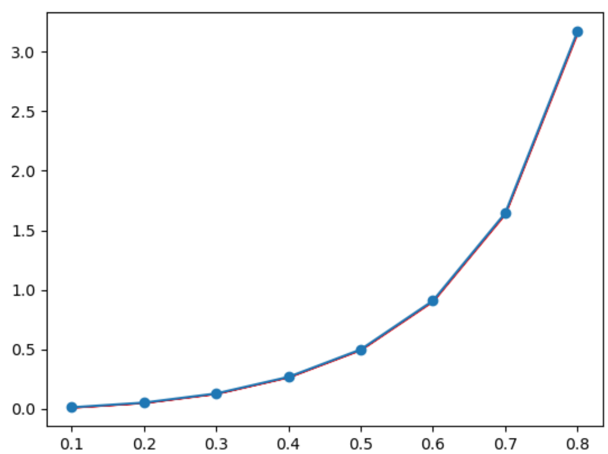
Utilization



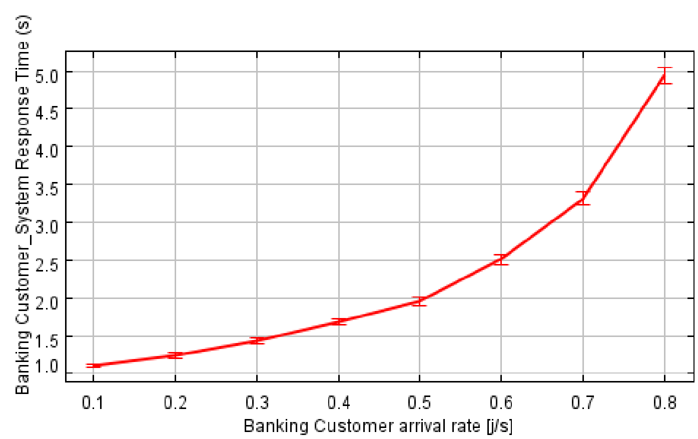
time-average number of customers in system



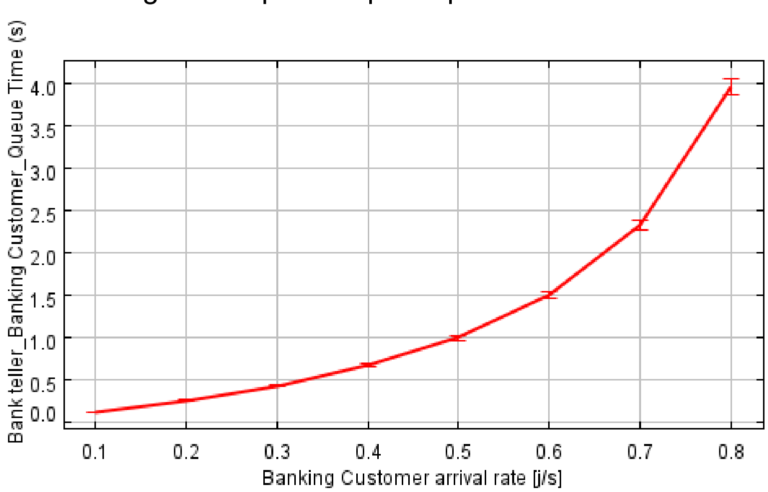
time-average number of customers in queue



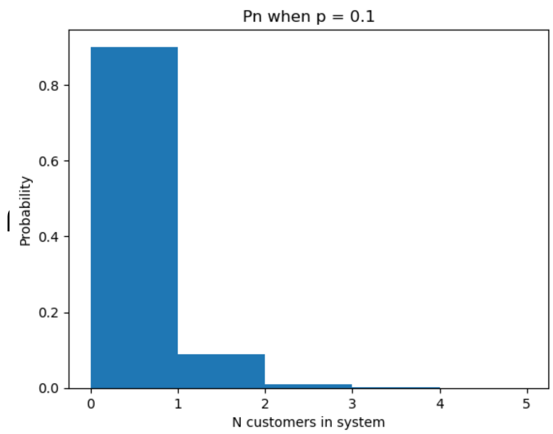
average time spent in system per customer

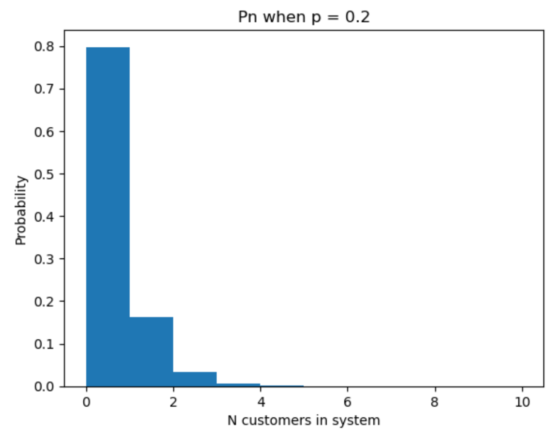


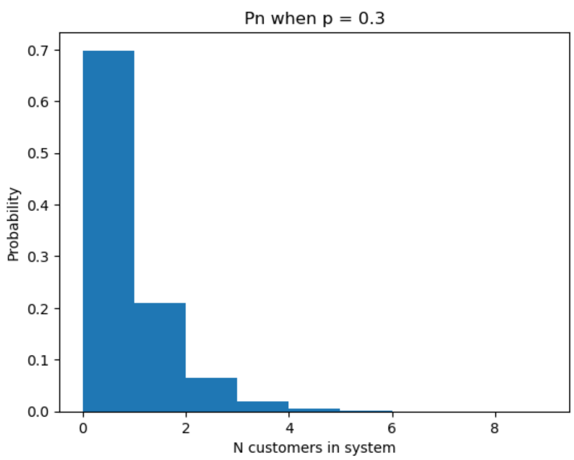
average time spent in queue per customer

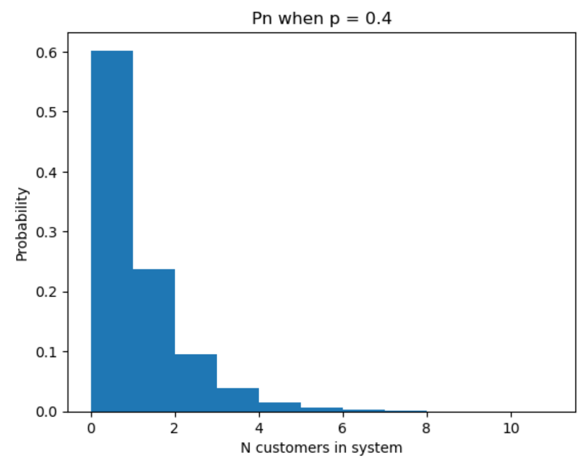


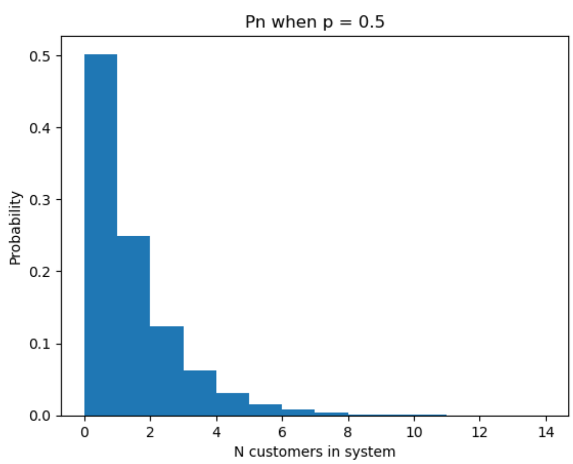
b. System states from JMT stats tab (re-do it in Python for better clarity)

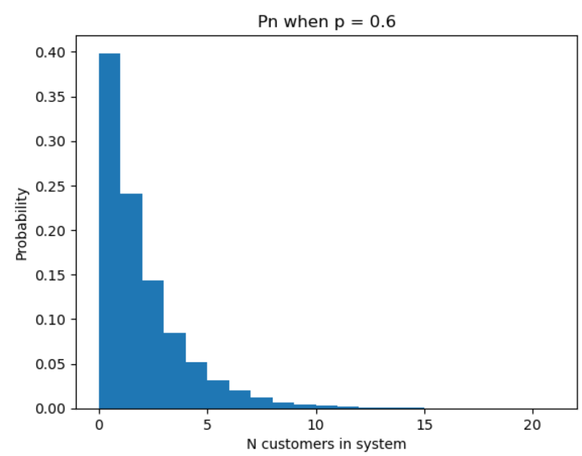


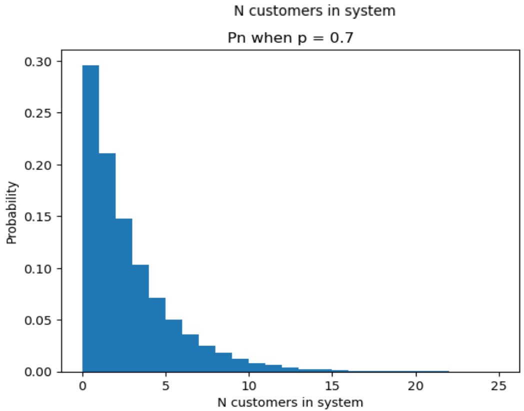


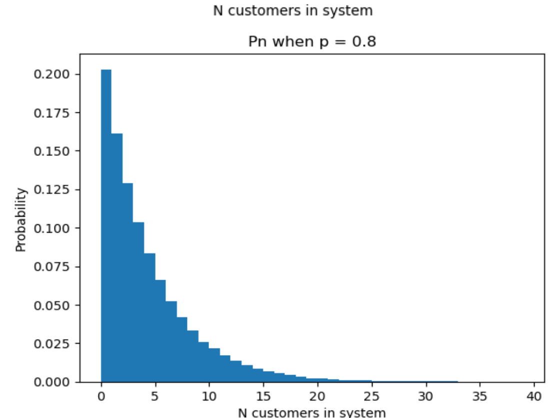












Best case of ρ	Worse case of ρ	Best case of L	Worse case of L	Best case of LQ	Worse case of LQ	Best case of W	Worse case of W	Best case of WQ	Worse case of WQ
0.002	0.0031	0.0014	0.0026	0.0002	0.0006	0.018	0.0291	0.0013	0.0073
0.0023	0.0073	0.0062	0.0067	0.0004	0.0014	0.0242	0.0474	0.0005	0.0138
0.0024	0.0154	0.0103	0.0129	0	0.0018	0.0247	0.0456	0.0093	0.016
0.0089	0.0131	0.0036	0.0314	0.0011	0.0017	0.0237	0.0578	0.0171	0.0179
0.0097	0.012	0.0225	0.0235	0	0.0044	0.0199	0.0965	0.0197	0.0335
0.0064	0.0294	0.0244	0.0662	0.0045	0.0115	0.0633	0.0814	0.0322	0.0424
0.0109	0.0123	0.0046	0.076	0.0059	0.0159	0.0736	0.1052	0.0535	0.0592
0.0059	0.0197	0.2155	0.0676	0.0217	0.0393	0.0545	0.1619	0.0639	0.1363

c. Comparison between theoretical values and simulated results
From calculating the best-case and worst-case errors for each performance metrics, we can see several worse-case and best-case errors of W and W_Q are greater than maximum error (0.05). Additional replications are necessary. It is possible that the value of maximum error needs adjustments considering multiple errors from W and W_Q exceeds the value of maximum error; or the queuing model needs improvement when arrival rate increases. The value P_n, the probability of having n customers in the system are very similar in theory and simulations. When utilization increases, the probability of having 0 customer in the system decreases gradually; however, there is also a high chance of having a longer queue length. In addition, the value of L_Q is calculated through little's law which may not accurately reflect the performance of the queuing model. Alternative method of data collection is needed.

d. Statistics collected for exponential inter-arrival time distribution
Kolmogorov-Smirnov test is used to evaluate the inter-arrival time distribution (with the help of scipy.stats module and .cdf() function). Since service time is set to exponential distribution with a constant mean of 1, validation of inter-arrival time distribution is sufficient.

20 samples are randomly selected (λ = 0.5): 0.443278724, 1.707226407, 1.305955981, 0.645164104, 4.217947887, 2.491242071, 4.924627621, 6.593462374, 1.776549345, 10.59783812, 15.98806097, 1.009894575,

1.50406247, 0.450606048, 0.092749436, 4.951197757, 3.184693559, 0.55837947, 9.238345837, 8.97604417

Sort the data in ascending order and calculate the empirical cumulative distribution data. Calculate $D_0 = \max|F(x) - S_N(x)|$ (using Python); $D_0 = 0.279 < D_{0.05,20}$ (0.294). We accept that the inter-arrival times are sample from a exponential distribution when $\lambda = 1$ with 0.05 significance level

e. Testing of the random generator used

Chi-Square test is selected for frequency test. 100 samples are randomly selected from the $\lambda = 1$ simulation result, set $k = 100^{1/2} = 10$. Compare it with uniform distribution (mean of 1, U(0, 2))

$E_i = n * (1/k)$

xi	Observed Freq. Oi	Expected Freq. Ei	(Oi-Ei)^2 / Ei
1	16	10	3.6
2	16	10	3.6
3	15	10	2.5
4	8	10	0.4
5	10	10	0
6	9	10	0.1
7	9	10	0.1
8	6	10	1.6
9	6	10	1.6
10	5	10	2.5

$X_0^2 = 3.6 < X_{0.05,\,9} = 16.919$, null hypothesis is not rejected. Test of uniformity has passed.

Test of correlation is performed to check for independence. 20 samples are selected (same sample used in the previous K-S test) at 0.05 significance level. Starting with the 1st sample, the correlation between every 3rd number is examined. Set M=5 (determined by sample size n=20 and the lag m=3)

$\rho_{13} = -0.1002$, $\sigma_{\rho 13} = 0.1179$, $-Z_{0.025} <= Z_0 = -0.8505 <= Z_{0.025} = 1.96$.

Hypothesis is not rejected. Independence is validated.

f. Conclusion

Most of the simulated performance indices match with the theoretical values under different utilization. However, the best-case error of system response time exceeds the maximum error for most simulations. It could be influenced by the process of data collection. Exponential inter-arrival time distribution has been validated using Kolmogorov-Smirnov test. Frequency and autocorrelation tests have been performed and passed.

5. Basic M/G/1 queuing model

5.1 Methodology

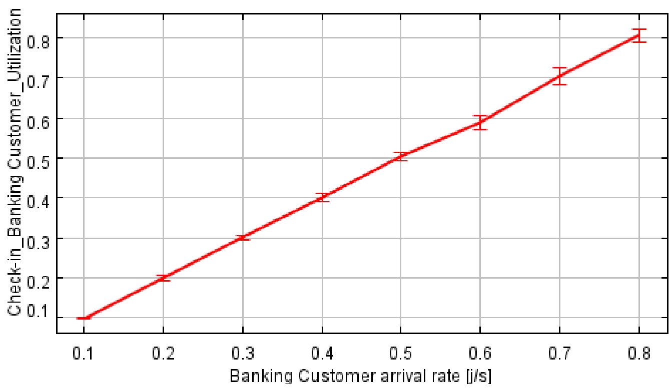
The structure of the M/G/1 queuing model is very similar to the M/M/1 model used above except that service time is uniformly distributed with a mean of 1 (U(0, 2)). Relevant images are generated and tests are run using Python.

5.2 Collected simulation results

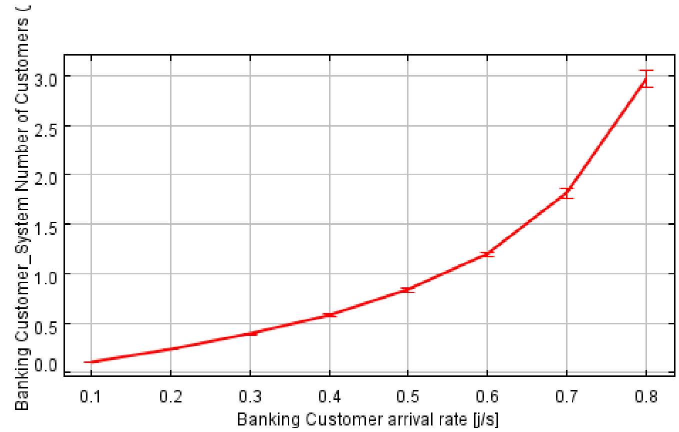
a. Performance metrics

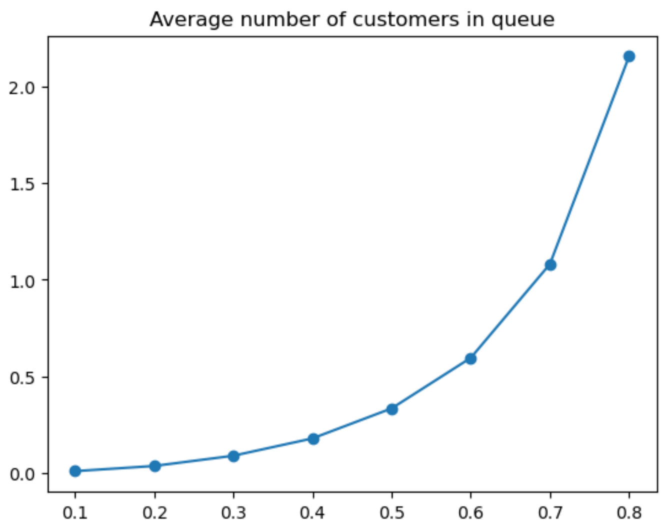
Mean of ρ	CI of ρ	Mean of L	CI of L	Mean of LQ	CI of LQ	Mean of W	CI of W	Mean of WQ	CI of WQ
0.0999	[0.0973, 0.1024]	0.1064	[0.1039, 0.1089]	0.0073	[0.0069, 0.0077]	1.0766	[1.0536, 1.0996]	0.0730	[0.0710, 0.0750]
0.2008	[0.1951, 0.2065]	0.2352	[0.2306, 0.2399]	0.0341	[0.0325, 0.0358]	1.1756	[1.1549, 1.1962]	0.1691	[0.1653, 0.1729]
0.3011	[0.2943, 0.3080]	0.3886	[0.3776, 0.3995]	0.0872	[0.0832, 0.0913]	1.2808	[1.2467, 1.3150]	0.2881	[0.2820, 0.2943]
0.4016	[0.3897, 0.4135]	0.5758	[0.5610, 0.5907]	0.1771	[0.1694, 0.1852]	1.4480	[1.4176, 1.4784]	0.4453	[0.4323, 0.4583]
0.5039	[0.4946, 0.5132]	0.8344	[0.8104, 0.8584]	0.3327	[0.3147, 0.3517]	1.6638	[1.6284, 1.6993]	0.6636	[0.6458, 0.6814]
0.5883	[0.5717, 0.6049]	1.1974	[1.1753, 1.2195]	0.5948	[0.5636, 0.6276]	1.9671	[1.9213, 2.0129]	0.9877	[0.9610, 1.0143]
0.7053	[0.6848, 0.7257]	1.8122	[1.7629, 1.8616]	1.0810	[1.0240, 1.1411]	2.5530	[2.4886, 2.6173]	1.5527	[1.5115, 1.5940]
0.8057	[0.7893, 0.8221]	2.9700	[2.8830, 3.0569]	2.1581	[2.0630, 2.2581]	3.6633	[3.5796, 3.7470]	2.6906	[2.6358, 2.7454]

Utilization

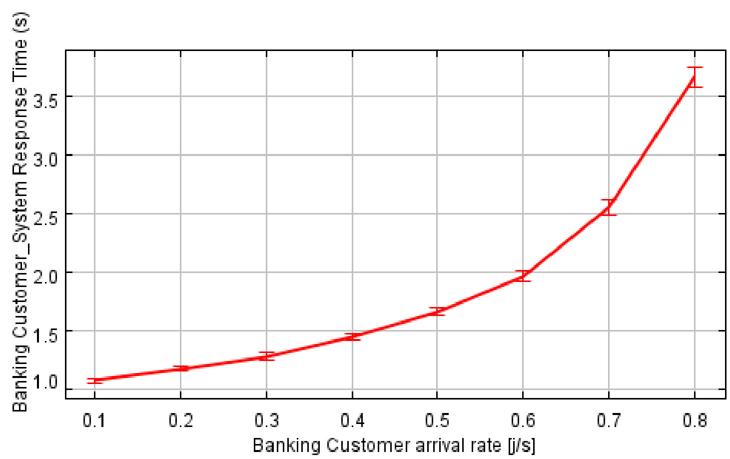


Average number of customers in system

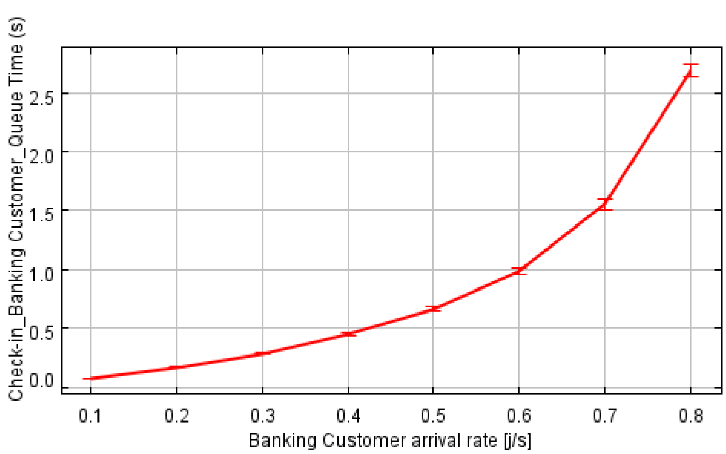




Average time spent in system per customer

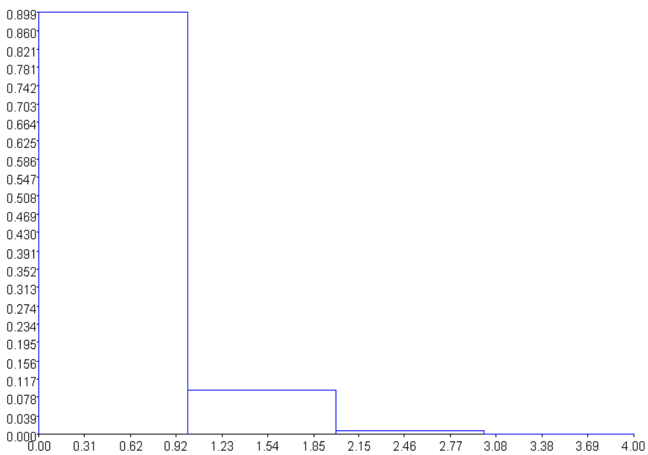


Average time spent in queue per customer

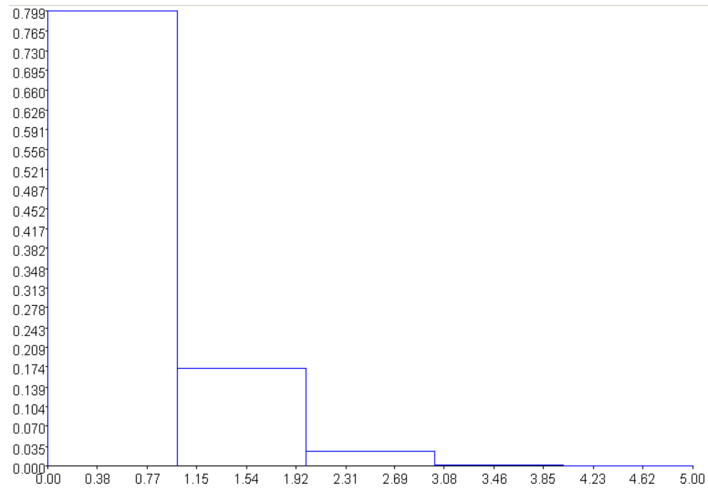


b. System state

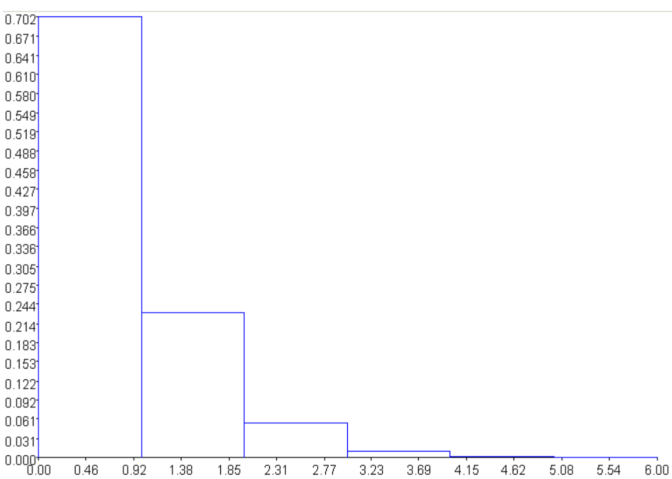
System state when ρ = 0.1



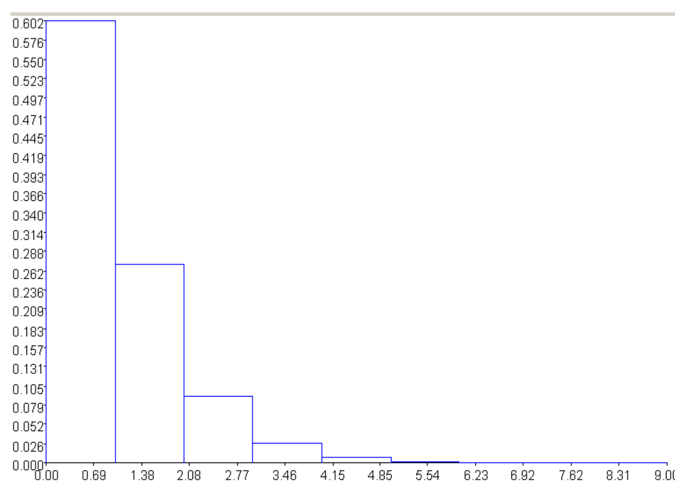
System state when ρ = 0.2



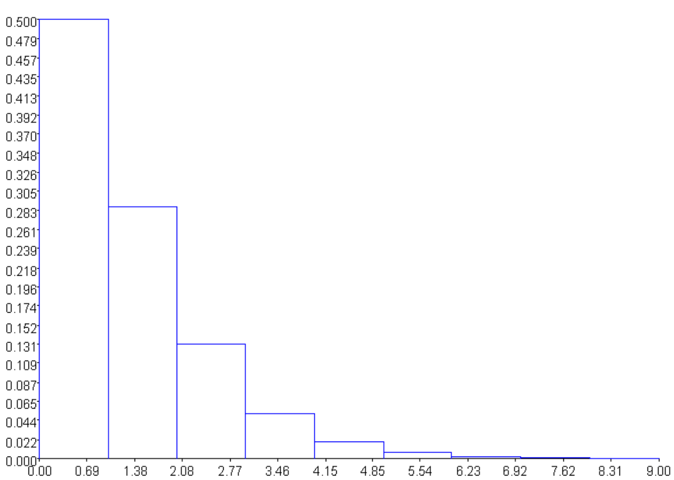
System state when ρ = 0.3



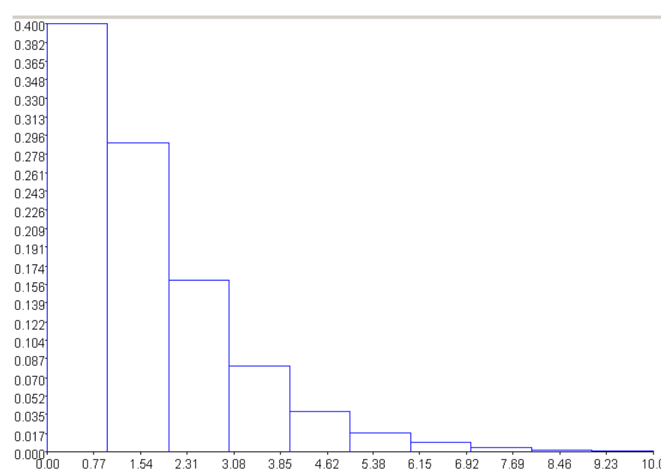
System state when ρ = 0.4



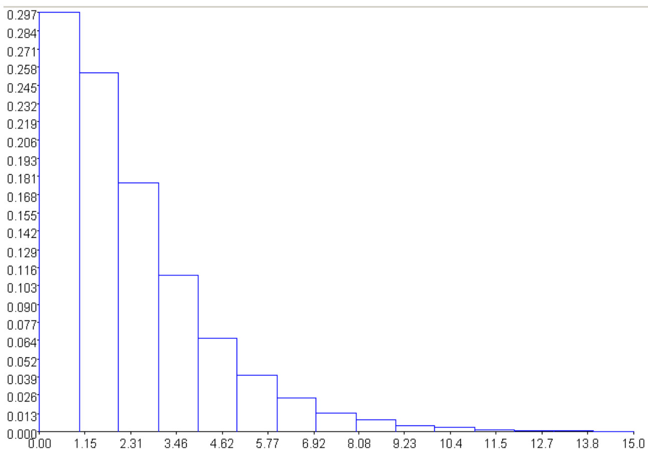
System state when ρ = 0.5



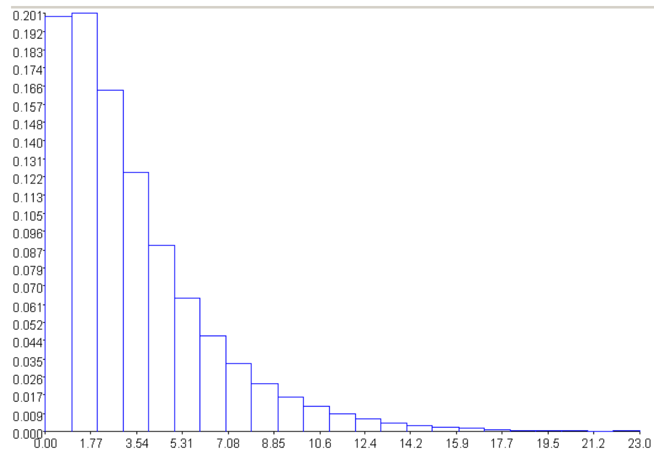
System state when ρ = 0.6



System state when ρ = 0.7



System state when ρ = 0.8



c. Simulation analysis and comparison between M/M/1 and M/G/1 performance

Best case of ρ	Worst case of ρ	Best case of L	Worse case of L	Best case of LQ	Worse case of LQ	Best case of W	Worse case of W	Best case of WQ	Worse case of WQ
0.0025	0.0027	0.0025	0.0025	0.0004	0.0004	0.023	0.023	0.002	0.002
0.0057	0.0057	0.0046	0.0047	0.0016	0.0017	0.0206	0.0207	0.0038	0.0038
0.0068	0.0069	0.0109	0.011	0.004	0.0041	0.0341	0.0342	0.0061	0.0062
0.0119	0.0119	0.0148	0.0149	0.0077	0.0081	0.0304	0.0304	0.013	0.013
0.0093	0.0093	0.024	0.024	0.018	0.019	0.0354	0.0355	0.0178	0.0178
0.0166	0.0166	0.0221	0.0221	0.0312	0.0328	0.0458	0.0458	0.0266	0.0267
0.0204	0.0205	0.0493	0.0494	0.057	0.0601	0.0644	0.0643	0.0412	0.0413
0.0164	0.0164	0.0869	0.087	0.0951	0.1	0.0837	0.0837	0.0548	0.0548

The model would require additional replications especially when the utilization is between 0.7 and 0.8. Very few worst-case and best-case errors exceed the maximum relative error (0.05) and there's less deviation between best-case and worst-case errors. Larger errors tend to appear when the utilization gets close to 0.8 and more errors are observed for the performance metric, W_Q. We could say the M/G/1 queuing model is more stable than the M/M/1 queuing model when the mean of inter-arrival time and service time are identical. Comparing the system states between two models, both models have very similar system state histograms when utilization is [0.1, 0.2, 0.3, 0.4, 0.5]. When utilization is [0.6, 0.7, 0.8], P_i, i=1, 2, 3, 4, 5, is slightly increased.

d. Statistics collected for uniform service time distribution

Kolmogorov-Smirnov test is used to evaluate the service time distribution. The simulated service time value is computed through collected W_Q and W (W - W_Q).

20 samples are randomly selected (λ = 0.5): 1.801153634, 1.876646251, 0.83877102, 0.02558439, 1.854616124, 0.955981964, 1.163366245, 0.828400851, 1.877480127, 1.302293764, 0.659099115, 0.895598944, 0.04969456, 0.949896481, 1.195423341, 1.874920803, 0.691548965, 1.289870438, 0.94821716.

Sort the data in ascending order and calculate the empirical cumulative distribution data. Calculate D0 = max|F(x) − SN (x)| (using Python); $D_0 = 0.230 < D_{0.05,20}$ (0.294). We accept that the service times are sample from a uniform distribution U(0, 2) with 0.05 significance level

e. Testing of the random generator used

Chi-Square test is selected for frequency test. 100 samples are randomly selected from the U(0,2) simulation result, set k = 1001/2 = 10. Compare it with uniform distribution (mean of 1, U(0, 2))

Ei = n * (1/k)

xi	Observed Freq. Oi	Expected Freq. Ei	(Oi-Ei)^2 / Ei
1	13	10	0.9
2	5	10	2.5
3	14	10	1.6
4	11	10	0.1
5	13	10	0.9
6	7	10	0.9
7	6	10	1.6
8	9	10	0.1
9	10	10	0
10	12	10	0.4

$X_0^2 = 9.0 < X_{0.05,\,9} = 16.919$, null hypothesis is not rejected. Test of uniformity has passed.

Test of correlation is performed to check for independence. 20 samples are selected (same sample used in the previous K-S test) at 0.05 significance level. Starting with the 1st sample, the correlation between every 3rd number is examined. Set M=5 (determined by sample size n=20 and the lag m=3)

ρ_{13} = -0.3378, $\sigma_{\rho 13}$ = 0.1179, $-Z_{0.025} <= Z_0 = 2.8651 <= Z_{0.025} = 1.96$.

Hypothesis is rejected. Validation of data independence has failed.

f. Conclusion

Comparing to the M/M/1 performance, the mean values of M/G/1 queuing model (with the same mean service and inter-arrival time)'s performance metrics are greater than the M/M/1 queuing model. The performance becomes greater when the utilization increases. However, M/G/1 is more stable than M/M/1 since M/G/1's worst-case and best-case errors have smaller deviations and its maximum worst-case error is smaller than M/M/1 maximum worst-case error. Uniform service time distribution has been validated; however, it fails the autocorrelation test. One possible explanation for it is that the service time is obtained by subtracting W_Q (average time spent in system per customer) from W (average time spent in queue per customer)while W is correlated to W_Q in queueing models. Unfortunately, JMT cannot collect the service time directly. The exploration of JMT or other software tools could be done.

6. M/M/C=3/N=15 queuing model

6.1 Methodology

JMT is used to model the M/M/3/15 system. A source (car owners), 1 queue of 3 servers (mechanics) with 15 capacity in total (queue+service) which applies non-preemptive station scheduling and first come first serve (FCFS) policies, and a sink are used. By enabling the What-if analysis parameters, changing the arrival rate of customers from 0.3 to 2.4 in 8 steps. The service time of the mechanics is in exponential distributions with a constant mean of 1. Confidence level is set to 95% and the maximum error is 0.03. Python matplotlib.pyplot is used when images cannot be generated through JMT

6.2 Theoretical value

Mean of p	Mean of L	Mean of Lq	Mean of W	Mean of Wq
0.1	0.3004	0.0004	1.0014	0.0014
0.2	0.6061	0.0062	1.0103	0.0103
0.3	0.9300	0.0300	1.0333	0.0333
0.4	1.2941	0.0941	1.0784	0.0784
0.5	1.7364	0.2364	1.1576	0.1576
0.6	2.3251	0.5256	1.2921	0.2921
0.7	3.1765	1.0808	1.5157	0.5157
0.8	4.4350	2.0571	1.8651	0.8651

Use $\lambda * P_N$ to get expected drop rate

$$P_0 = \left[1 + \sum_{n=1}^{c} \frac{a^n}{n!} + \frac{a^c}{c!} \sum_{n=c+1}^{N} \rho^{n-c}\right]^{-1}$$

$$\rho = \lambda / c\mu; \quad a = \lambda / \mu$$

$$P_N = \frac{a^N P_0}{c!c^{N-c}}$$

$$L_Q = \frac{\rho a^c P_0}{c!(1-\rho)^2}\left[1-\rho^{N-c}-(N-c)\rho^{N-c}(1-\rho)\right]$$

$$\lambda_e = \lambda(1-P_N)$$

$$w_Q = \frac{L_Q}{\lambda_e}$$

$$w = w_Q + \frac{1}{\mu}$$

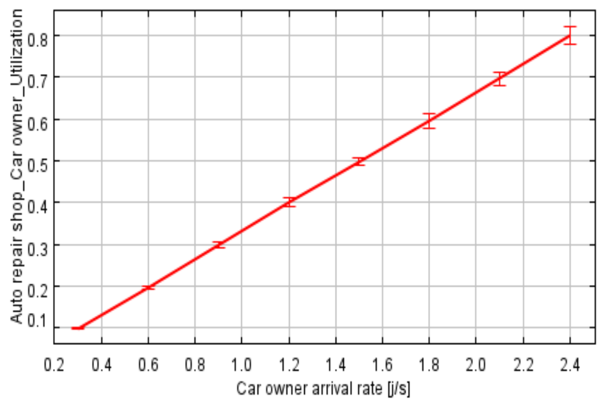
$$L = \lambda_e . w$$

Arrival Rate	Drop Rate
0.3	9.99E-16
0.6	4.84E-15
0.9	2.34E-08
1.2	1.71E-06
1.5	4.34E-05
1.8	0.0006
2.1	0.0043
2.4	0.0221

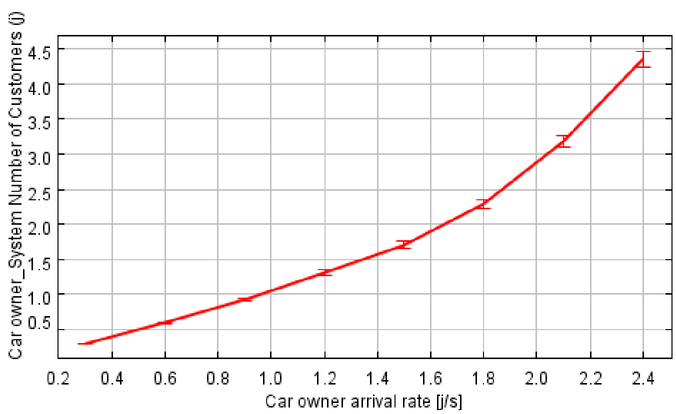
6.3 Collected performance metrics

Mean of p	CI of p	Mean of L	CI of L	Mean of Lq	CI of Lq	Mean of W	CI of W	Mean of Wq	CI of Wq	Mean of Drop Rate	CI of Drop Rate
0.1006	[0.0977, 0.1034]	0.3022	[0.2934, 0.3109]	4E-04	[3.66E-04, 4.35E-04]	1.009	[0.983, 1.0349]	1E-03	[0.00125, 1.42E-3]	0	[0, 0]
0.1997	[0.1951, 0.2042]	0.6057	[0.5948, 0.6166]	0.0063	[0.0060, 0.0066]	1.0066	[0.9764, 1.0367]	0.0105	[0.0102, 0.0108]	0	[0, 0]
0.3026	[0.2959, 0.3094]	0.9356	[0.9091, 0.9622]	0.0302	[0.0289, 0.0315]	1.0234	[1.0008, 1.0459]	0.0333	[0.0325, 0.0341]	0	[0, 0]
0.3979	[0.3892, 0.4065]	1.29	[1.2589, 1.3210]	0.0941	[0.0900, 0.0984]	1.0657	[1.0465, 1.0849]	0.0785	[0.0763, 0.0807]	3.31E-05	[0, 0]
0.5026	[0.488, 0.5171]	1.7382	[1.7103, 1.7662]	0.2356	[0.2242, 0.2478]	1.1571	[1.1368, 1.1773]	0.1567	[0.1536, 0.1599]	0	[0, 0]
0.5989	[0.5892, 0.6086]	2.3398	[2.2735, 2.4061]	0.5397	[0.5113, 0.5700]	1.2827	[1.2442, 1.3211]	0.2967	[0.2895, 0.3040]	5.65E-04	[5.52E-04, 5.78E-04]
0.7059	[0.6861, 0.7256]	3.1639	[3.0899, 3.2379]	1.0751	[1.0335, 1.1181]	1.5047	[1.4742, 1.5352]	0.5107	[0.4992, 0.5222]	4.25E-03	[4.16E-03, 4.34E-03]
0.7852	[0.7664, 0.8039]	4.4495	[4.3468, 4.5523]	2.0318	[1.9430, 2.1240]	1.9057	[1.8505, 1.9609]	0.851	[0.8284, 0.8737]	0.0218	[0.0212, 0.0224]

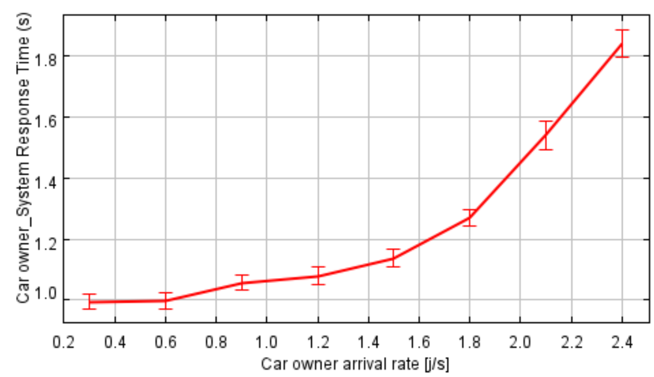
ρ


Auto repair shop_Car owner_Utilization
Car owner arrival rate [j/s]

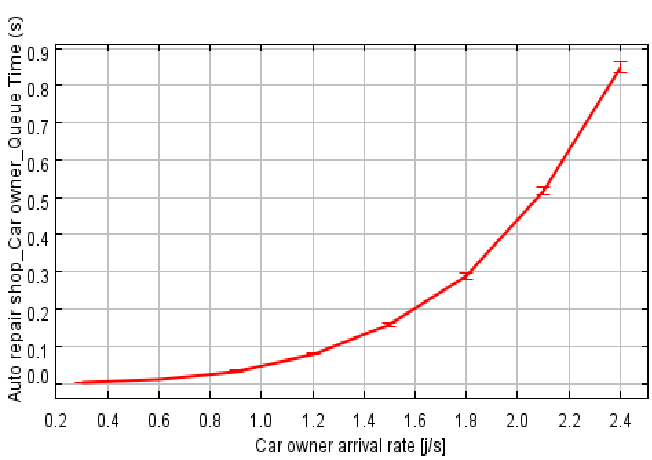
L

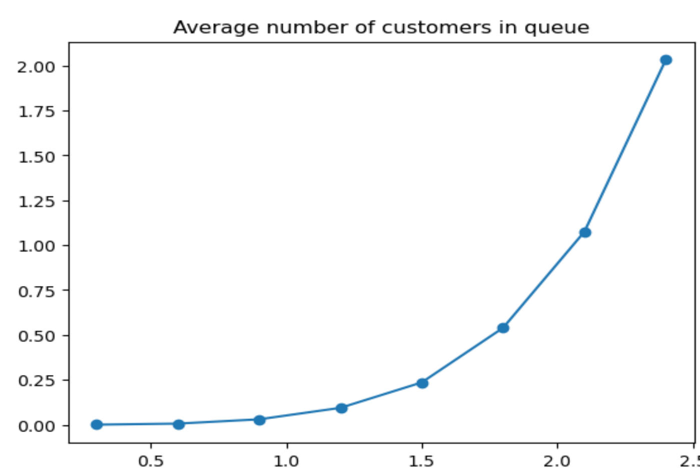

Car owner_System Number of Customers (j)
Car owner arrival rate [j/s]

W


Car owner_System Response Time (s)
Car owner arrival rate [j/s]

W_q


Auto repair shop_Car owner_Queue Time (s)
Car owner arrival rate [j/s]


Average number of customers in queue

6.3 Comparison between theoretical value and simulation results

Best case of ρ	Worse case of ρ	Best case of L	Worse case of L	Best case of LQ	Worse case of LQ	Best case of W	Worse case of W	Best case of WQ	Worse case of WQ	Best-case of Drop Rate	Worst-case of Drop Rate
0.0023	0.0034	0.007	0.0105	3.4E-05	3.5E-05	0.0184	0.0335	0.00002	0.00015	9.99E-16	9.99E-16
0.0042	0.0049	0.0105	0.0113	0.0002	0.0004	0.0264	0.0339	0.0001	0.0005	4.84E-15	4.84E-15
0.0041	0.0094	0.0209	0.0322	0.0011	0.0015	0.0126	0.0325	0.0008	0.0008	2.34E-08	2.34E-08
0.0065	0.0108	0.0269	0.0352	0.0041	0.0043	0.0065	0.0319	0.0021	0.0023	1.71E-06	1.71E-06
0.012	0.0171	0.0261	0.0298	0.0114	0.0122	0.0197	0.0208	0.0023	0.004	4.34E-05	4.34E-05
0.0086	0.0108	0.0516	0.081	0.0143	0.0444	0.029	0.0479	0.0026	0.0119	2.20E-05	4.80E-05
0.0139	0.0256	0.0614	0.0866	0.0373	0.0473	0.0195	0.0415	0.0065	0.0165	4.00E-05	1.40E-04
0.0039	0.0336	0.0882	0.1173	0.0669	0.1141	0.0958	0.0146	0.0086	0.0367	6.00E-04	6.00E-04

From the simulation results, we can see that both L, L_Q and W's worst-case errors exceed maximum error (0.05). However, these appear more often for performance metric L when utilization is between 0.6 and 0.8. For all performance metrics, worst-case errors are likely to exceed the maximum error when utilization is greater than or equal to 0.6. Additional replications would be required.

Collected worst-case and best-case errors of drop rate are always less than the maximum error. When utilization is between 0.1 and 0.4, the drop rate is 0 (or very small to be evaluated). JMT has not computed the drop rate when utilization=[0.1, 0.2, 0.3, 0.4] with the requested precision which explains the occurrence of value 0. When the utilization has increased from 0.7 to 0.8, a significant increase in the drop rate has been observed although it is still considered as relatively small (2%). Balking or reneging behaviour could be added to have a more comprehensive analysis.

7. Network of Queues

7.1 Methodology

The network of queues consists of 1 source (customer class of exponential service distribution with mean of 1), 3 queues (custom queue, inspection A queue, and inspection B queue) and 1 sink. Source is connected to the custom queue; all customers will arrive at the custom queue first. When a passenger have been served at the custom queue, they will be routed to inspection A queue with a probability of 0.7 and inspection B with a probability of 0.3 (use the probability routing strategy of JMT queues). When the service has been fulfilled at either inspection A or B, all passengers are routed to the sink (or exit). Arrival rate is set to [0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40] and the mean of service time of each service station remains 1.0

7.2 Theoretical value (utilization and total response time only)

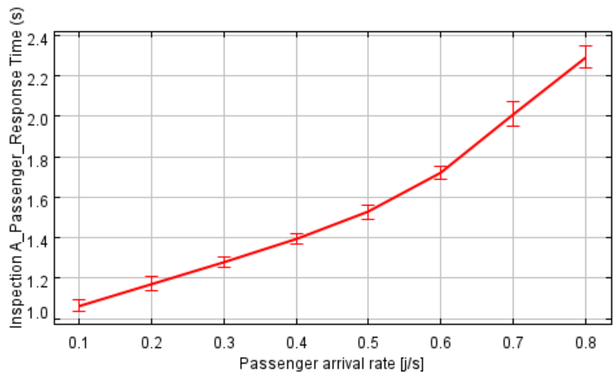
Mean ρ of Inspection A	Mean ρ of Inspection B	Overall Mean of ρ	Mean response time of stream inspection A	Mean response time of stream inspection B
0.07	0.03	0.10	1.0753	1.0309
0.14	0.06	0.20	1.1628	1.0638
0.21	0.09	0.30	1.2658	1.0989
0.28	0.12	0.40	1.3889	1.1364
0.35	0.15	0.50	1.5385	1.1765
0.42	0.18	0.60	1.7241	1.2195
0.49	0.21	0.70	1.9608	1.2658
0.56	0.24	0.80	2.2727	1.3158

7.3 Collected performance metrics

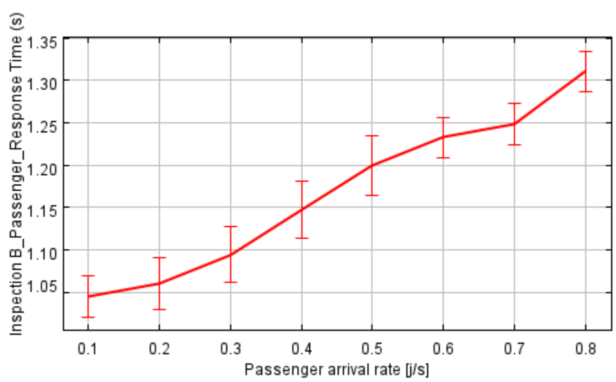
Mean ρ of Inspection A	CI of ρ of Inspection A	Mean ρ of Inspection B	CI of ρ of Inspection B	Mean response time of stream inspection A	CI of response time of stream inspection A	Mean response time of stream inspection B	CI of response time of stream inspection B
0.0695	[0.0683 ,0.0708]	0.0305	[0.0298, 0.0312]	1.0651	[1.0349, 1.0952]	1.0443	[1.0199, 1.0686]
0.1451	[0.1418,0 .1483]	0.0586	[0.0569, 0.0602]	1.1721	[1.1370, 1.2072]	1.0603	[1.0293, 1.0913]
0.2084	[0.2029, 0.2140]	0.0904	[0.0883, 0.0925]	1.2786	[1.2506, 1.3067]	1.0945	[1.0620, 1.1271]
0.2826	[0.2771, 0.2881]	0.1182	[0.1152, 0.1211]	1.3936	[1.3704, 1.4169]	1.1471	[1.1136, 1.1806]
0.3527	[0.3459, 0.3595]	0.1507	[0.1475, 0.1539]	1.5255	[1.4923, 1.5587]	1.1996	[1.1642, 1.2350]
0.4225	[0.4140, 0.4311]	0.1790	[0.1747, 0.1833]	1.7192	[1.6854, 1.7531]	1.233	[1.2093, 1.2567]

0.5033	[0.4920, 0.5146]	0.2066	[0.2012, 0.2119]	2.0089	[1.9490, 2.0689]	1.2484	[1.2234, 1.2734]
0.5572	[0.5496, 0.5647]	0.2430	[0.2381, 0.2479]	2.2919	[2.2358, 2.3480]	1.3109	[1.2872, 1.3346]

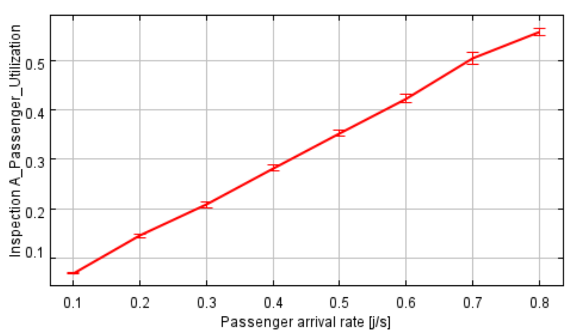
Mean response time per customer of stream A



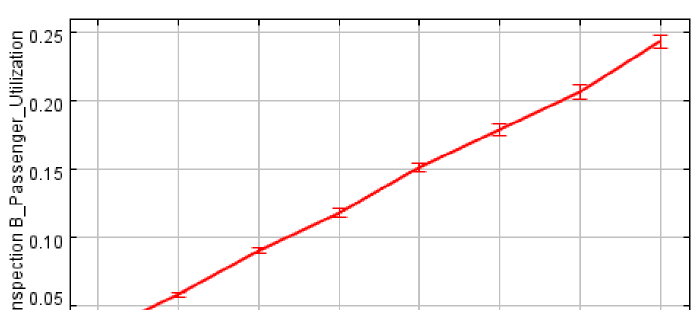
Mean response time per customer of stream B



Utilization of stream A



Utilization of stream B



Comparison between theoretical and simulated values

Best case of Inspection A ρ	Worst case of Inspection A ρ	Best case of Inspection B ρ	Worst case of Inspection B ρ	Best-case response time of inspection A	Worst-case response time of inspection A	Best-case response time of inspection B	Worst-case response time of inspection B
0.0008	0.0017	0.0002	0.0012	0.0199	0.0404	0.011	0.0377
0.0018	0.0083	0.0002	0.0031	0.0258	0.0444	0.0345	0.0275
0.004	0.0071	0.0017	0.0025	0.0152	0.0409	0.0369	0.0282
0.0029	0.0081	0.0011	0.0048	0.0185	0.028	0.0228	0.0442
0.0041	0.0095	0.0025	0.0039	0.0202	0.0462	0.0123	0.0585
0.006	0.0111	0.0033	0.0053	0.029	0.0387	0.0102	0.0372
0.002	0.0246	0.0019	0.0088	0.0118	0.1081	0.0076	0.0424
0.0047	0.0104	0.0019	0.0079	0.0753	0.0189	0.0286	0.0188

From the simulation results, we can see that most simulation results do not exceed the maximum relative errors. Both mean value and confidence interval of collected performance metrics are very close to the theoretical values. When the utilization rate increases from 0.1 to 0.8, the mean response time for both streams increases linearly with a relatively slow slope. Comparing it to other queuing models, the network of queues has shown better stability in response time.