Modelling queue stability of spatial queues with a moving server using discrete-event simulation

Jeremy Larsen supervised by Peter Taylor

The University of Melbourne

Motivation

Consider an office building with a single elevator that has $1 \leq n \leq N$ floors and a ground floor. People arrive to each floor randomly intending only to travel from their respective floor to the ground floor [1].

A natural question about this system is how to operate the elevator (design a service policy) in a way that reduces the expected waiting times of arrivals and ensures that the lengths of the queues on each floor remain stable.

This poster explores different circumstances for which four different service policies are stable and in what circumstances different policies are more efficient than others.

The model

To make matters more concrete, imagine the same office building as above but the building contains only 5 floors and a ground floor. Additionally, people arrive to each of the 5 floors **uniformly**.

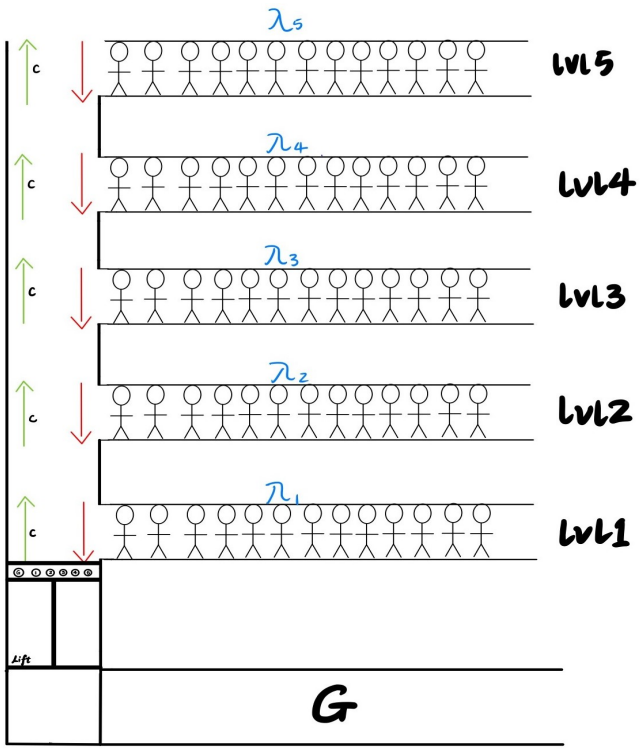

Figure 1. A representation of the elevator system.

- Customers arrive to each of the 5 floors **uniformly** according to a **Poisson arrival process** with rate λ_i, $i \in \{1, 2, 3, 4, 5\}$.
- Customers only intend to go from the floor they arrived at the **ground floor**.
- The elevator operates on a **first come first serve basis**, where it will always move towards the next arrival.
- The elevator takes time c to move between adjacent floors, takes time a to open its doors, and time b to close its doors with $a + b + c = D$
- The capacity of the elevator is determined by the service policy.

Service policies

Four service policies were considered and adapted to the model for simulation.

1. M/G/1: The lift moves from the ground floor to the next arrival, only takes that arrival and heads straight back to ground floor.
2. M/G/20*: The lift moves from the ground floor to the next arrival, takes the first 20 arrivals on that floor and heads straight back to the ground floor.
3. M/G/∞*: The lift moves in the same way as the previous two policies but takes everyone on the floor.
4. Checksfloors20 (CF20): The lift moves from the ground floor to the floor of the next arrival and takes up to 20 arrivals from that floor, if there are less than 20 arrivals on that floor the elevator checks the floor below it and fills the remaining spots, it repeats this until it reaches ground floor.

Method

The main method used for simulation was **discrete-event simulation**. Simulations were carried out using Python where various packages were used, most notably **SimPy**, to handle the discrete-event simulation. Additionally, each simulation was run with approx. $750,000$ arrivals, taking approx. 50 minutes .

M/G/1 as a baseline

An analytical bound for the queue stability of the M/G/1 policy is given by $\rho = \sum_{n=1}^{5} 2\lambda_n(a + b + nc) < 1$ (1) [1].

To ensure the accuracy of the simulation, the simulation was first tested by simulating the M/G/1 policy and checked to see that queue stability was in-fact bounded by (1).

As seen below in *Table 1*, as $\rho \to 1$ the expected waiting time of any arrival starts to increase exponentially. As $\rho \geq 1$ the expected waiting times become very large and indicates the queue lengths are becoming unbounded. Additionally, when $\rho \leq 1$ the expected waiting times remain quite small, indicating that the bound is at 1.

D	$\sum_{n=1}^{5} \lambda_n$	ρ	Expected waiting time
0.065	5.00	0.950	1.940
0.068	6.70	1.025	131.000
0.068	5.00	1.050	235.926
0.068	6.70	1.193	793.690
0.068	7.75	1.300	1136.764

Table 1. Simulated results of the M/G/1 policy adapted to the model using discrete-event simulation.

M/G/20* and M/G/∞

Initial results of simulations indicate that the M/G/20* and the M/G/∞* policy increase the efficiency of the system in a non-linear way.

M/G/20*

In the case $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5$ simulated results shown in *Table 2* indicate the queue lengths are stable when $\rho \approx 23$. Additionally, since the capacity has increased by 20 but the value of $\rho \approx 23$ this suggests that the increase in efficiency is non-linear.

D	$\sum_{n=1}^{5} \lambda_n$	ρ	Expected waiting time
0.510	12.50	21.000	8.4
0.545	12.50	22.720	84.6
0.500	15.00	25.000	425.550

Table 2. Simulated results of the M/G/20* policy with $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5$ adapted to the model using discrete-event simulation.

However, when $\lambda_1 = \lambda_2 \geq \lambda_3 = \lambda_4 = \lambda_5$ the policy becomes stable for larger values of ρ than previously. Therefore, the policy would be more optimised for a system that has faster arrivals on the earlier floors.

D	$\sum_{n=1}^{5} \lambda_n$	ρ	Expected waiting time
0.500	11.00	28.33	10.88
0.510	31.00	38.30	10.24
0.510	36.00	45.00	779.70

Table 3. Simulated results of the M/G/20* policy with $\lambda_1 = \lambda_2 \geq \lambda_3 = \lambda_4 = \lambda_5$ adapted to the model using discrete-event simulation.

M/G/∞*

A consequence of increasing the capacity of the lift to ∞ is that it seems to be stable for any values of parameter set $\{a, b, c, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5\}$.

D	$\sum_{n=1}^{5} \lambda_n$	ρ	Expected waiting time
15.000	35.00	1750.00	142.71
30.000	20.00	2000.00	248.63
30.000	25.00	25000.00	248.55
45.000	25.00	3750.000	374.000

Table 4. Simulated results of the M/G/∞* policy with $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5$

CF20

The next logical elevator policy to test is the CF20 policy, where the lift picks up passengers on its way to ground floor.

In the case $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5$ simulation results in *Table 5* indicate the queue lengths are stable when $\rho \approx 32$ and the average waiting time is bounded, suggesting a significant improvement to the M/G/20* policy.

D	$\sum_{n=1}^{5} \lambda_n$	ρ	Expected waiting time
1.018	9.00	30.300	21.200
0.905	10.000	31.400	17.179
1.018	10.000	33.710	146.000

Table 5. Simulation results of the CF20 policy when $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5$.

However, when $\lambda_1 = \lambda_2 \geq \lambda_3 = \lambda_4 = \lambda_5$ results from *Table 6* show the expected waiting time get significantly larger for a smaller value of $\rho \approx 22$, suggesting the queue lengths have become unstable and the policy is less efficient than the M/G/20* in this case.

D	$\sum_{n=1}^{5} \lambda_n$	ρ	Expected waiting time
0.800	11.00	22.800	185.685
0.900	11.00	26.300	663.818
1.018	12.000	33.120	1374.800

Table 6. Simulation results of the CF20 policy when $\lambda_1 = \lambda_2 \geq \lambda_3 = \lambda_4 = \lambda_5$.

If $\lambda_1 = \lambda_2 = \lambda_3 \leq \lambda_4 = \lambda_5$ the policy becomes much more efficient and the system is stable at $\rho \geq 193$.

D	$\sum_{n=1}^{5} \lambda_n$	ρ	Expected waiting time
1.018	12.000	47.800	6.551
1.018	24.000	102.220	6.450
1.018	44.000	193	6.510

Table 7. Simulation results of the CF20 policy when $\lambda_1 = \lambda_2 = \lambda_3 \leq \lambda_4 = \lambda_5$.

Conclusions and conjecture

Below *Table 8* provides simulated bounds for various situations. Results indicate that for different values of $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5\}$ different policies become more efficient and it is unlikely that any one policy strickly dominates another.

Service policy	constant λ_i	Lower weighted	Upper weighted
M/G/1	1	1	1
M/G/20*	23	40	20
M/G/∞*	∞	∞	∞
CF20	32	21	193

Table 8. Simulated bounds for each policy at given weights of $\lambda_i, i \in \{1, 2, 3, 4, 5\}$

Conjecture

Consider the M/G/∞* policy, where the elevator has ∞ capacity but instead of there being 5 floors there is only 1. If the elevator took at an arbitrary slow speed to move between adjacent floors, open its doors and close them, say any $\epsilon > 0$ would the queue lengths become unstable?

The state space of this system can be thought of as $\mathbb{Z}$ which is the number of people in the queue at any point in time. Since the number in the queue becomes 0 every 2ϵ time units the system always returns to state 0 within a finite amount of time, implying the policy is stable.

What if there were 5 floors? Results from my simulations of the M/G/∞*policy indicate that there is no bound and in-fact the queue lengths will be stable for any values of $\{a, b, c, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5\}$.

What about if there were 10 floors? or 20? or even 10000000 floors? Would the queue lengths be stable? Intuitively, it feels like there would be a certain number of floors for which the queue lengths would become unstable but from my simulations it seems otherwise. Therefore, my conjecture is that for any finite number of floors and any $\epsilon > 0$ the queue lengths will be stable.

Reference

[1] Peter Taylor.
Spatial queues with a moving server.
Queueing Systems: Theory and Applications, 100(3):261–263, April 2022.

Jllarsen@student.unimelb.edu.au