A Heuristic for the Situational Application of a Destination Dispatch Elevator System

Ryan Do

April 2020

Contents

Introduction

In the realm of traditional elevator control algorithms, Highest Unanswered Floor First (HUFF), is a widely implemented one that has been shown to demonstrate a competitive performance even when compared to more sophisticated alternatives [3]. In HUFF, elevators start at either extreme of the chute and proceed in a single direction until all requests are answered.

A more recently developed elevator control technique known as destination dispatch (DD) moves the destination buttons from within the car to the waiting area, making buttons absent inside the cars. In this set-up, information on a patron's intended destination is available to the system as soon as they register on a panel on each floor (left). DD has the advantage of knowing each patron's intended route so that cars can effectively prioritize pickup and drop-off routes according to demand. This also eliminates the scenario of cars being at capacity upon stopping to pick up passengers. Additionally, since each DD elevator only takes a single destination at a time, there are less stops en route. However, it is unclear in what scenarios a destination dispatch system is advantageous over the traditional HUFF algorithm. The objective of this study is to

determine a general heuristic for selecting between destination dispatch or HUFF based on the properties of some building in question. This report will firstly dive deep into the simulation modelling approach of the elevator systems. The experimental design will then be covered, ending with the results of the study and conclusions.

Problem Description

In specific, the problem to be solved is to determine for what set of building parameters values (characterized by number of floors and patrons per floor) destination dispatch is to be recommended over traditional HUFF, given a typical office building arrival process. DD is recommended for some set of building dimensions if less elevator cars are required for the system performance to satisfy some criteria in simulation. The experimental design section elaborates on more of the details.

Model Description

Simulation Logic

When modelling HUFF and destination dispatch, certain aspects were controlled to not obscure the root source of potential performance differences. Consistent behavior between HUFF and DD include idling on the spot, idle cars being prioritized, and to prioritize floors closest to terminal ends first. In addition, the NSPP arrival process and physical properties are also made to be consistent.

Traditional Highest Unanswered Floor First (HUFF) Elevator

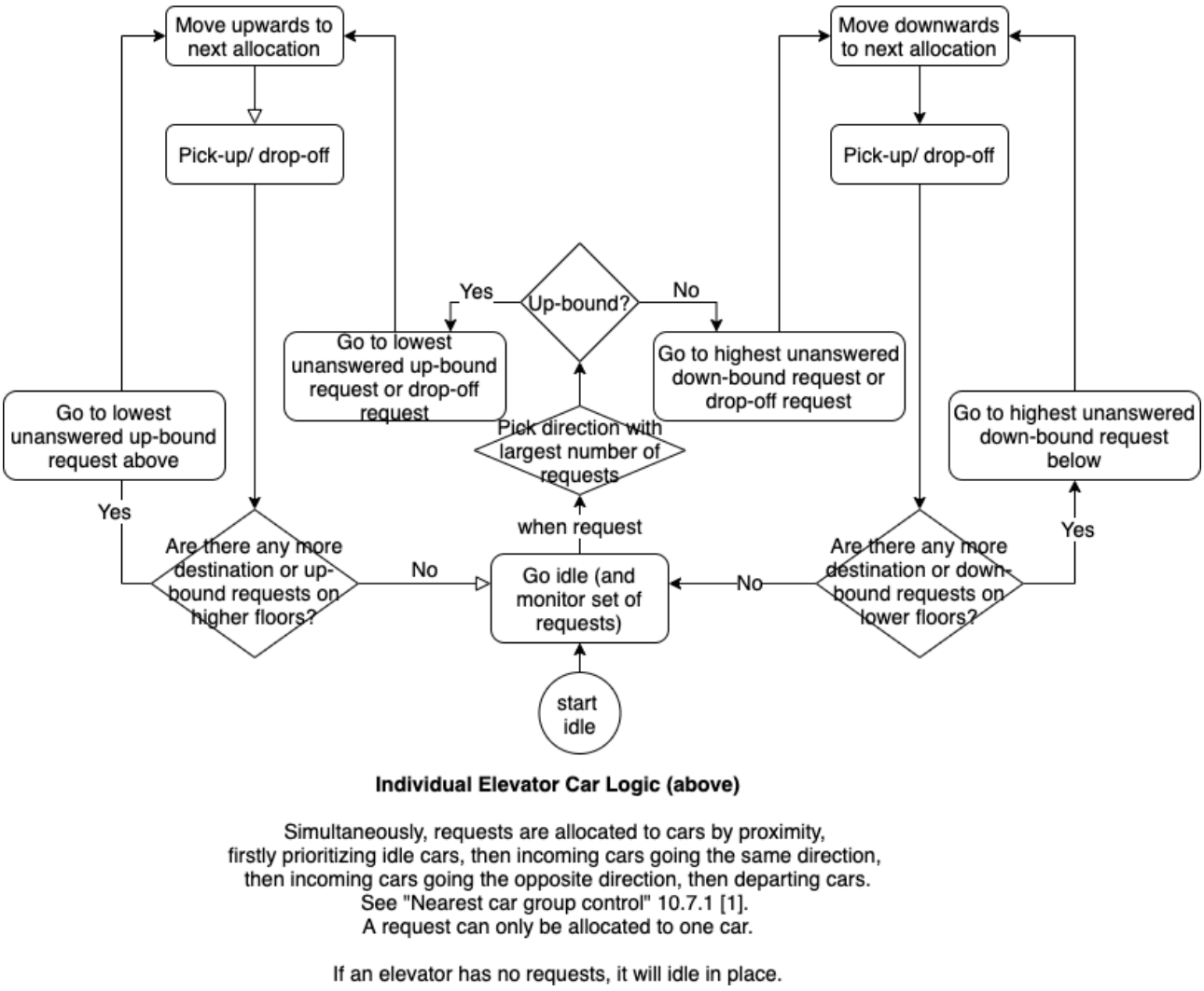


Figure 1. A complete description of the implemented HUFF logic as a flowchart.

In a nutshell, the HUFF system is implemented in simulation as two decision-making agent classes: (1) a controller dispatches pick-up requests to elevator cars based on prioritization rules, and (2) elevator cars complete the pool of requests allocated to each one in an intelligent order (Figure 1 for more detail). Some additional details and implications of the above logic include the following:

1. While a car is in transit to the next destination, if any new requests to pick-up from floors along the way are allocated to it, it will *not* be interrupted.
2. If a car is or becomes full when attempting to pick up from its current floor, the left-over waiting passengers re-submit their request(s).
3. Cars tend to complete all requests in one direction before switching directions.

Destination Dispatch (DD)

Note that there is little publicly available information on the exact detailed algorithms implemented in real destination dispatch systems. The algorithm detailed in Figure 2 is an informed assumption based on observation and a conceptual knowledge on its expected operation. Similarly to HUFF, two intelligent agents are implemented: elevator car and controller. The elevator car logic for DD is the same as HUFF. Cars look into their pool of requests, pick the direction with the largest number and start from the highest or lowest floor. The controller maintains a pool of unallocated tasks and cars allocate themselves tasks when they can take them. In addition, a couple custom data structures were designed to implemented DD efficiently (Figure 3). When an elevator is idle and ready to take a new task, it scans a central *source-destination matrix* and picks the column (destination) with the greatest sum.

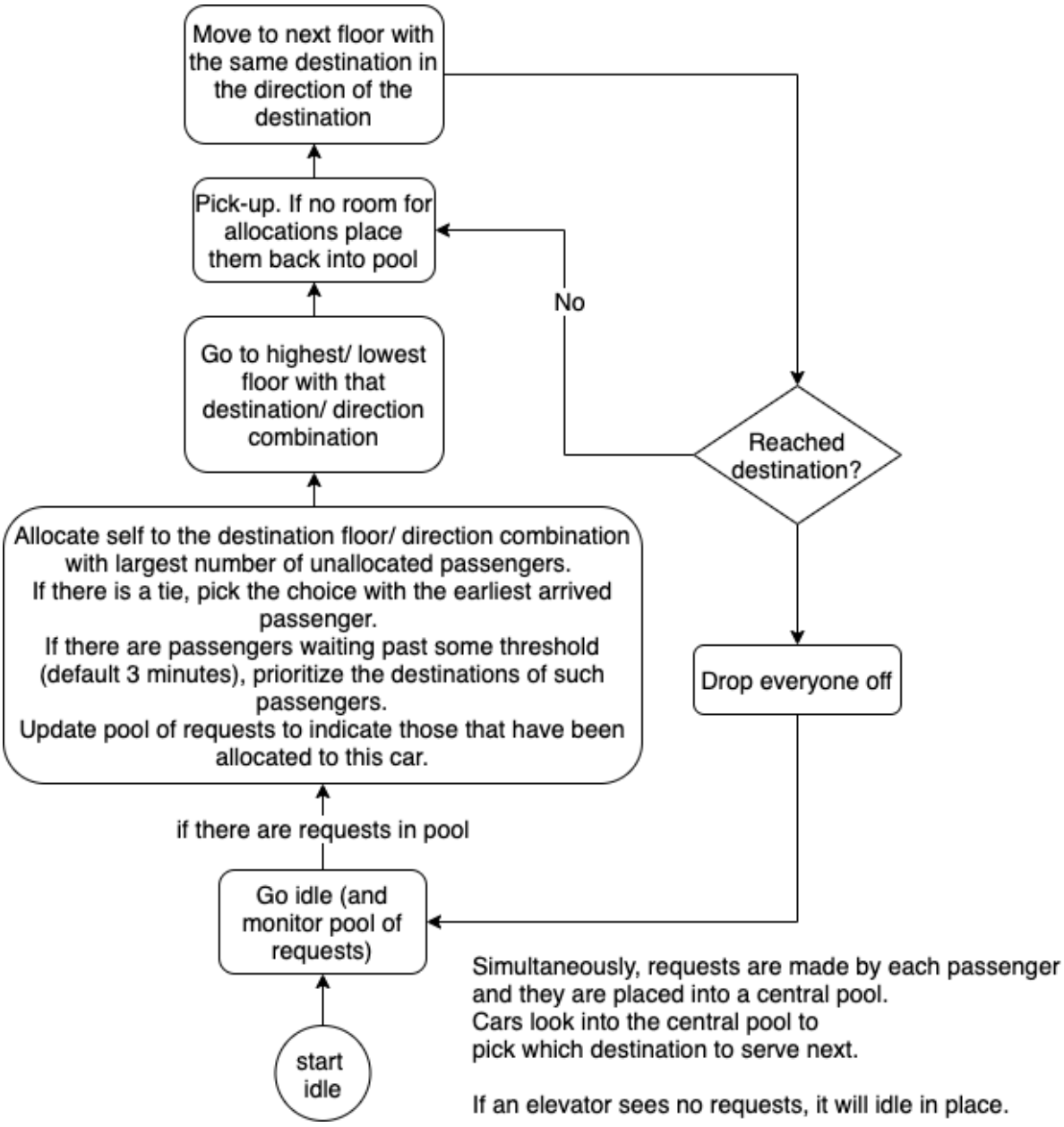


Figure 2. A complete description of the implemented destination dispatch system.

Those requests are then placed in its own *request pool*. The *source-destination queue-matrix* serves as a set of priority queues to rank destination floors by urgency and identify those with patrons waiting past some urgency threshold (3 minutes) or to identify highest time waiting patrons to act as a tie breaker for the greatest sum.

0	2	0
0	0	1
0	0	0

0	P1, P2	P3
0	0	P4, P5
0	0	0

0	1
0	1
0	0

Figure 3. Left: Source-destination matrix - number of unallocated requests by source (rows) and destination (column). Middle: Source-destination queue-matrix – priority queue of waiting patrons for each (source, destination) pair. Right: Request pool – pick-up requests for each source floor (rows) and direction (columns). One per elevator car.

Additional details and implications of the destination dispatch logic:

1. Cars should never allocate themselves to more passengers than expected. Therefore, there could be multiple cars going to the same ultimate destination.
2. On the other hand, if there is room for any unallocated passengers going to same destination, pick them up too. For example, the car in Figure 3 will pick up 2 passengers on floor 2 going to floor 3 despite only being allocated to 1.

System Properties

Input	Quantity	Units
Elevator car top speed	3.0	ms^{-1}
Elevator car acceleration/ deceleration	4.5	ms^{-2}
Inter-floor distance	4.5	m
Door open-close time	2.5	s
Time for one passenger to enter/exit car	1.0	s
Car capacity	20	passengers
Number of elevator cars	Experimental variable	#
Number of floors	Experimental variable	#
Occupants per floor	Experimental variable	#

Non-stationary arrival processes

The time-varying arrival rates for upwards traffic, downwards traffic, and cross-floor traffic are modeled by three non-stationary Poisson processes, generated using the thinning method. The arrival processes are modeled after the daily patterns of an office building, inspired by Barney section 4.4, 6.5 [1]:

	8AM – 9AM	9AM – 12:00PM	12PM – 1PM	1PM– 2PM	2PM – 6PM	6PM – 8AM
Per 5 minute % of building population	Upwards traffic (From ground floor to any floor)					
	8	0.125	2.75	5.5	0.125	0
	Downwards traffic (From any floor to ground floor)					
	0.125	5.5	2.75	0.125	8	0
	Intra-building traffic (From any floor to any floor)					
	1.0	1.0	1.0	1.0	1.0	0

Common Random Numbers

CRNs are applied to all replications to reduce the variance of the difference between scenarios through increasing covariance, thus increasing the probability of correct selection in the elevator fleet sizing procedure that follows. In situations where compared scenarios have differing performing servers, a single RNG stream may become out of sync. However in the case here, a single stream is sufficient since the shared inter-arrival times and source/ destination floors are the only random variables.

Model Assumptions

See A.1 for a list of assumptions made for the simulation models.

Experimental Design

The required number of elevator cars (fleet size) for DD is to be evaluated and compared to HUFF for each pair in {3, 5, 10, 15, 20, 25, 30} floors X {25, 50, 100, 200, 300} occupants per floor. Each pair in the cartesian product constitutes one scenario. For each scenario, the fixed budget strategy is used with a single replication to minimize bias. A fixed budget of a simulated 2 days is selected as it covers two full work-day cycles and the run-time is sufficiently small such that all scenarios can be run in a timely manner. Fixed-precision offers little benefit since run-time is costly. As a result, reaching a target precision is a luxury, and minimizing bias via 1 replication is priority. Local minimization of the MSER is used to determine the warmup period:

$$MSER(d) = \frac{1}{(m-d)^2} \sum_{i=d+1}^{m} \left(Y_i - \bar{\bar{Y}}(m,d)\right)^2 = \sum_{i=d+1}^{m} Y_i^2 - \frac{1}{m-d} \sum_{i=d+1}^{m} \left(Y_i - \bar{\bar{Y}}\right)^2$$

Fleet size requirements for each of DD and HUFF are evaluated on two different criteria, each describing different passenger needs. The first criteria is for the system to achieve $<$ 50s passenger waiting time to enter the elevator with a 95% probability at 95% confidence. Along the same lines, the second criteria is for the system to achieve a total *journey time* of less than some threshold with 95% probability at 95% confidence, with this threshold defined to scale linearly with the number of floors in this case (since journey time includes movement between floors and so this threshold must be adaptive).

For some system, this threshold is defined to be $3\frac{h_{building}}{v_{max}} + 50\ seconds$. The CI of probability $\widehat{\boldsymbol{F}}$ of a performance measure Y (wait time or journey time) not surpassing some threshold is: $\widehat{\boldsymbol{F}} = \frac{1}{n}\sum_{i=1}^{n} I(Y_i \leq \text{threshold})$; $\boldsymbol{S^2} = \left(\frac{n}{n-1}\right)\hat{F}(1-\hat{F})$; $\boldsymbol{CI} = \hat{F} \pm z_{1-\frac{\alpha}{2}}\frac{S}{\sqrt{n}}$

Results

Figure 4. Sizing requirements determined by journey time; HUFF (left), DD (right).

The fleet sizing values shown in Figures 4 and 5 are determined through a binary search where an initial space of [0, 100] cars is recursively halved and replication run until the minimum number of cars to achieve the respective criteria is found. It is clear that with increasing floor count and floor population the required fleet size increases in all cases. Some key insights are more evident when visualizing the relative improvement of DD over HUFF (Figure 6). Negative values indicate a smaller quoted fleet sizing for DD,

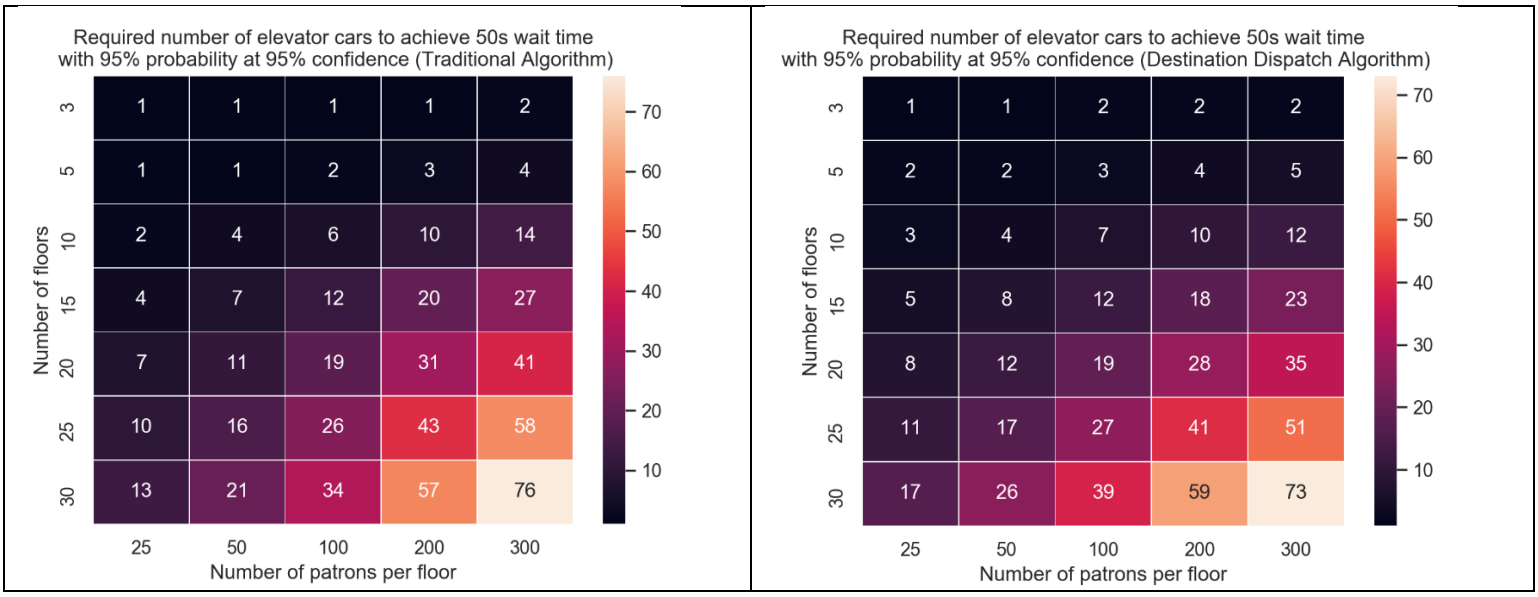


Figure 5. Sizing requirements determine by patron wait time; HUFF (left), DD (right).

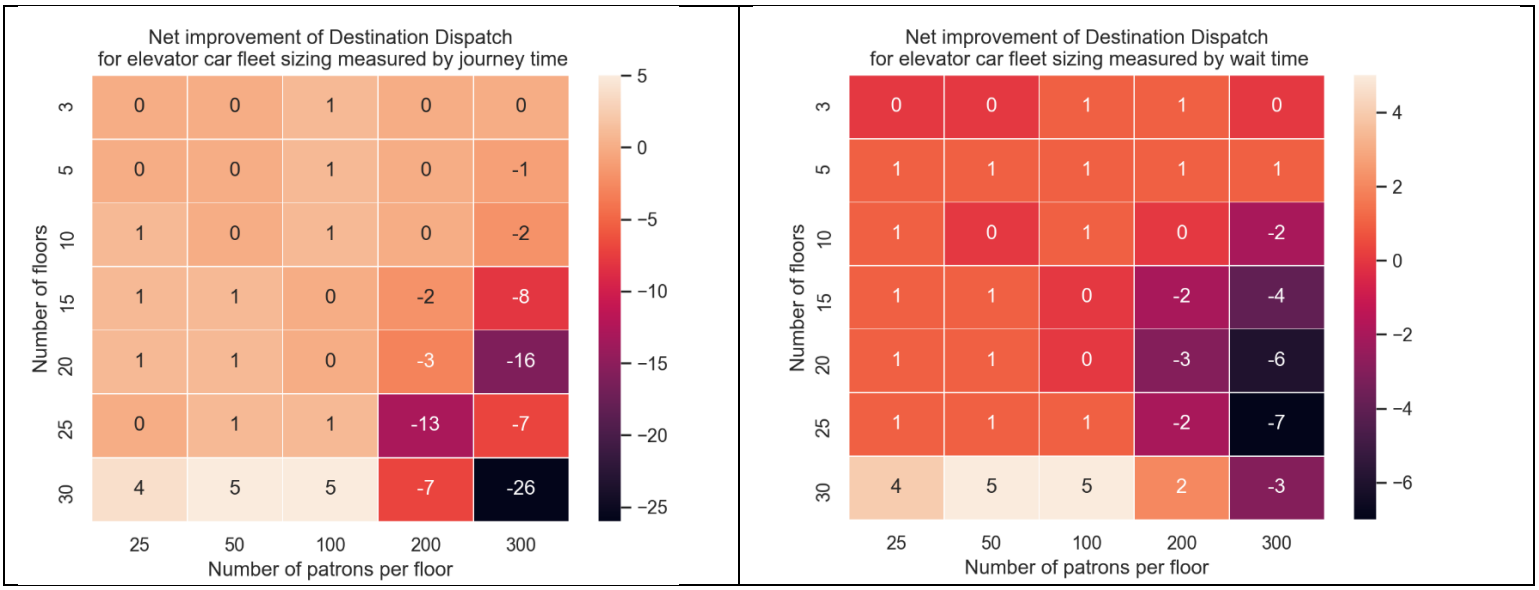


Figure 6. Delta in fleet sizing requirements from HUFF to DD for each criterion.

which represent the space of building properties where DD is recommended over HUFF. Figure 6 shows that whether evaluating by journey time or wait time, buildings with a large number of floors (10+) *and* a large occupancy per floor (150-200+) lend well to the implementation of a destination dispatch system. Using wait time as a

performance measure tends to be more forgiving to the HUFF algorithm since destination dispatch trades off low wait time for a more significantly lower travel time. The simulated benefit of DD is very substantial when sizing by journey time, demonstrating up to a 26 elevator car reduction at higher floor populations. In cases where there is a low floor population and higher floor count, HUFF actually performs better regardless of criteria. At very high floor occupancy (300+), DD is preferred regardless of floor count. As such, high floor occupancy is the principal determinant. To make this effect intuitive, at the high end of floor occupancy, DD cars can run at capacity (since there are more patrons going to each floor) while HUFF cars are also at capacity. However, DD cars only take on one destination at a time therefore journey time and as a consequence waiting times are lower.

As a secondary take-away, periods of time with high upwards traffic from a single floor (i.e. 8-9 AM in Figure 7) show *even greater marginal performance increase* in DD since DD cars only take destinations one a time while a single HUFF car can have requests to all floors. In Figure 7, high down-peak traffic (5-6PM) show similar queue sizes between DD and HUFF (controlling all other variables) while in high up-peak traffic (8-9AM) there is a 6-fold decrease in max queue size when opting for destination dispatch. In conclusion, building structures with a high occupancy per floor

(200+), moderate to high floor count (10+), and high upwards traffic are especially well suited for destination dispatch. Luxury cruise ships, for example, fulfill all conditions.

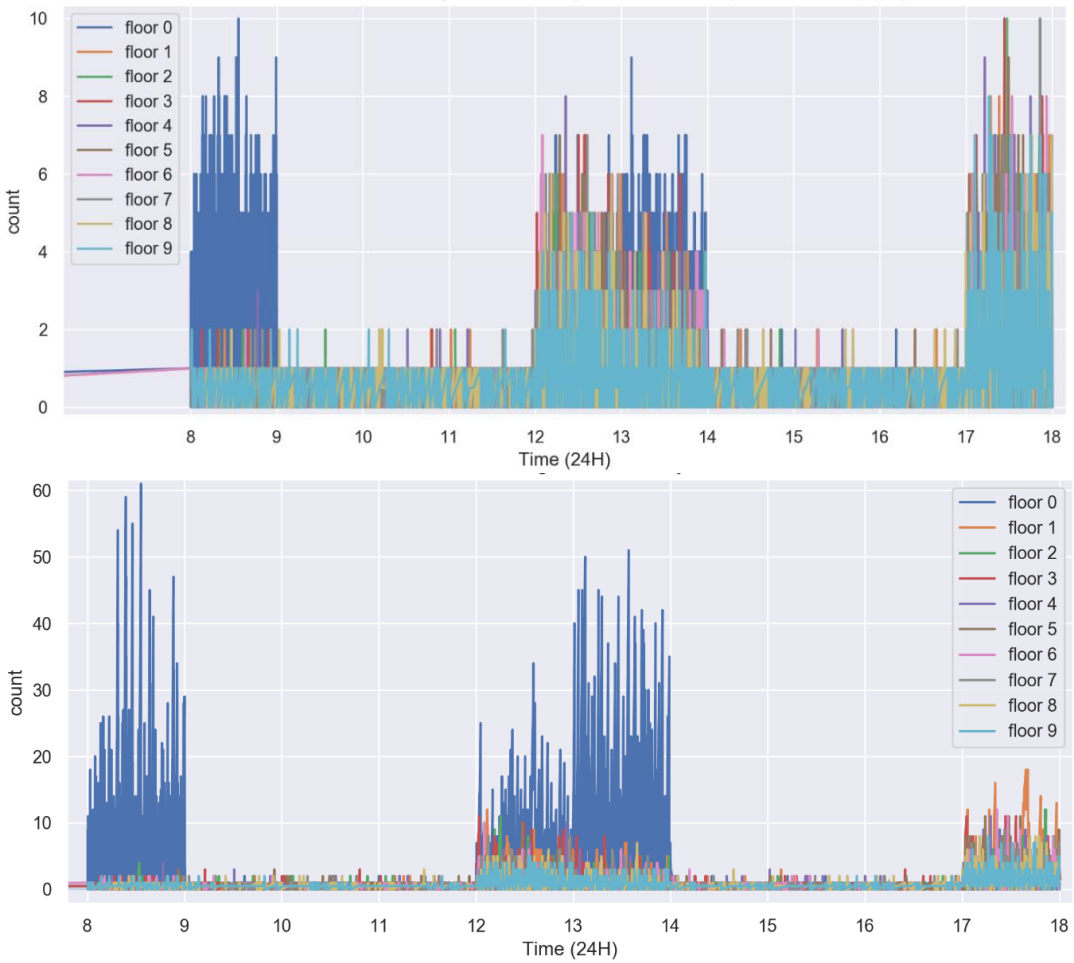


Figure 7. Queue sizes by floor over a single work day. DD (top), HUFF (bottom). 10 floors, 300 occupancy per floor, 12 cars for both.

References

[1] Barney, G. C., & Al-Sharif, L. (2016). *Elevator traffic handbook: theory and practice.*

[2] B. Nelson. (2013) Foundations and Methods of Stochastic Simulation:A First Course.

[3] R. Crites, A. Barto. (1996) *Improving Elevator Performance Using Reinforcement Learning.* NeurIPS 1996, Conference on Neural Information Processing Systems.

Appendix

A1. Model Assumptions

Passenger/ Building Assumptions

1. Equal demand to and from all non-ground floors for inter-floor traffic (uniformly distributed).
2. For destination dispatch, passengers will enter the car bound for the specific destination that they called.
3. For destination dispatch, each passenger in a group will press their floor individually.
4. For the traditional elevator, passengers will enter the first car going in their specified direction.
5. The building population per floor and arrival process are modelled after typical office buildings.
6. Building patrons do not ever use the stairs.

Elevator Logic Asssumptions

7. Elevator cars are not optimized to park idle in specific locations. They idle in place.

8. Elevator cars park idle with doors open. (whenever elevator cars move, it is guaranteed that they close the doors, move, then open the doors)

Elevator Dynamics Assumptions

9. Elevator cars have infinite jerk; they reach maximum acceleration instantaneously.
10. Elevator cars accelerate at the same magnitude as deceleration.
11. Elevator car acceleration is same up-bound and down-bound.
12. Elevator kinematic properties are based on Table 5.2 with 4.5m inter-floor distance and 60m travel [1]
13. If there is insufficient distance from source to destination to acceleration and deceleration to/from top speed, the car accelerates for half the distance then decelerates for the other half.

A2. Model Verification Checklist

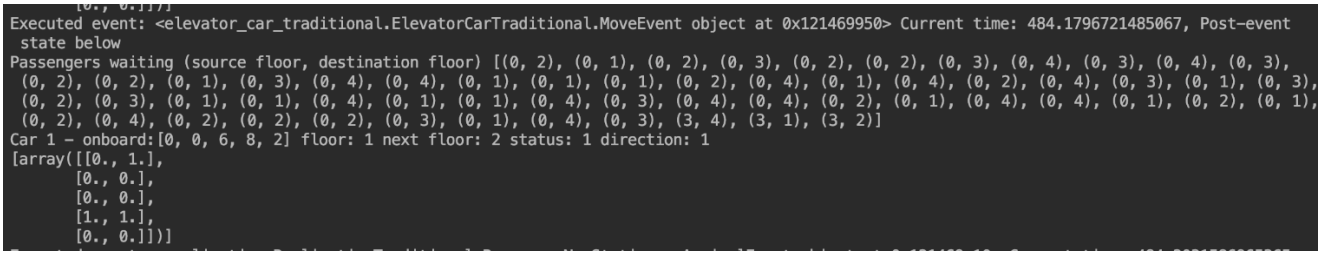
1. Traditional elevator logic is correct:

    a. Requests are allocated to cars appropriately based on priority rules

    b. Full cars will reject passengers, and passenger requests are reallocated

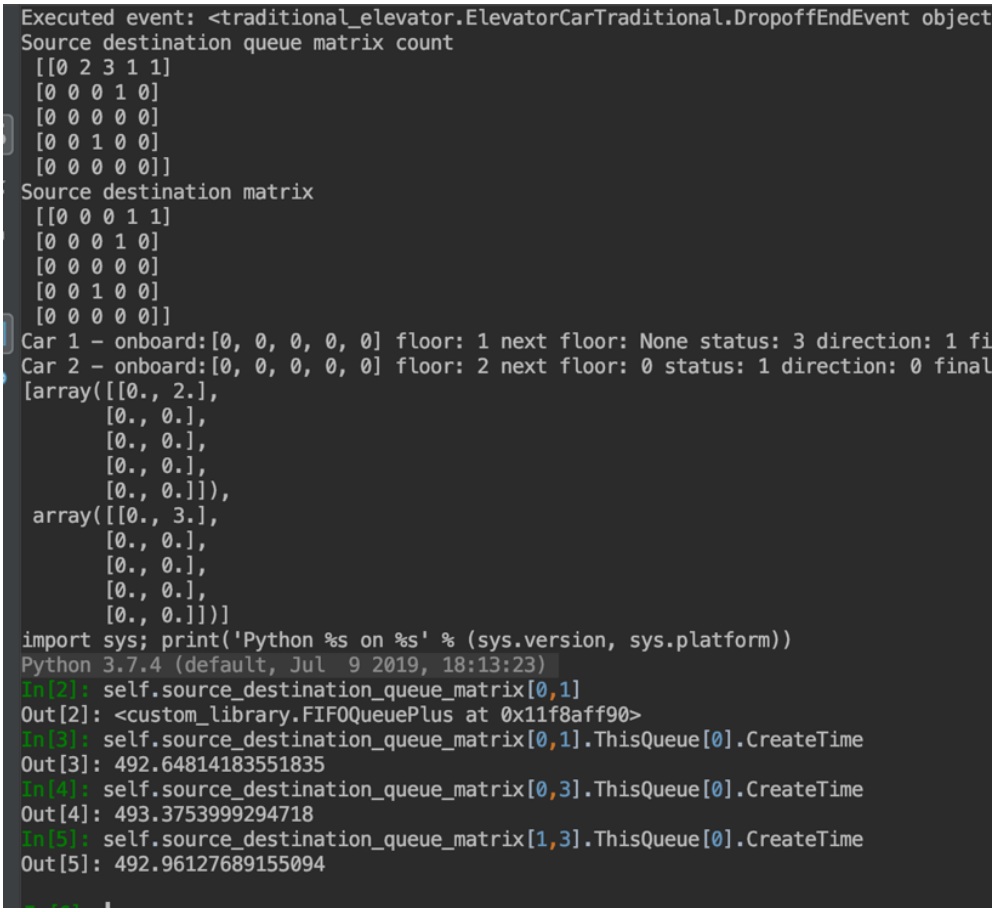
    c. Cars continue in one direction whenever possible before switching directions

    d. If a car has both picks and drops on a floor, it will do both starting with drop



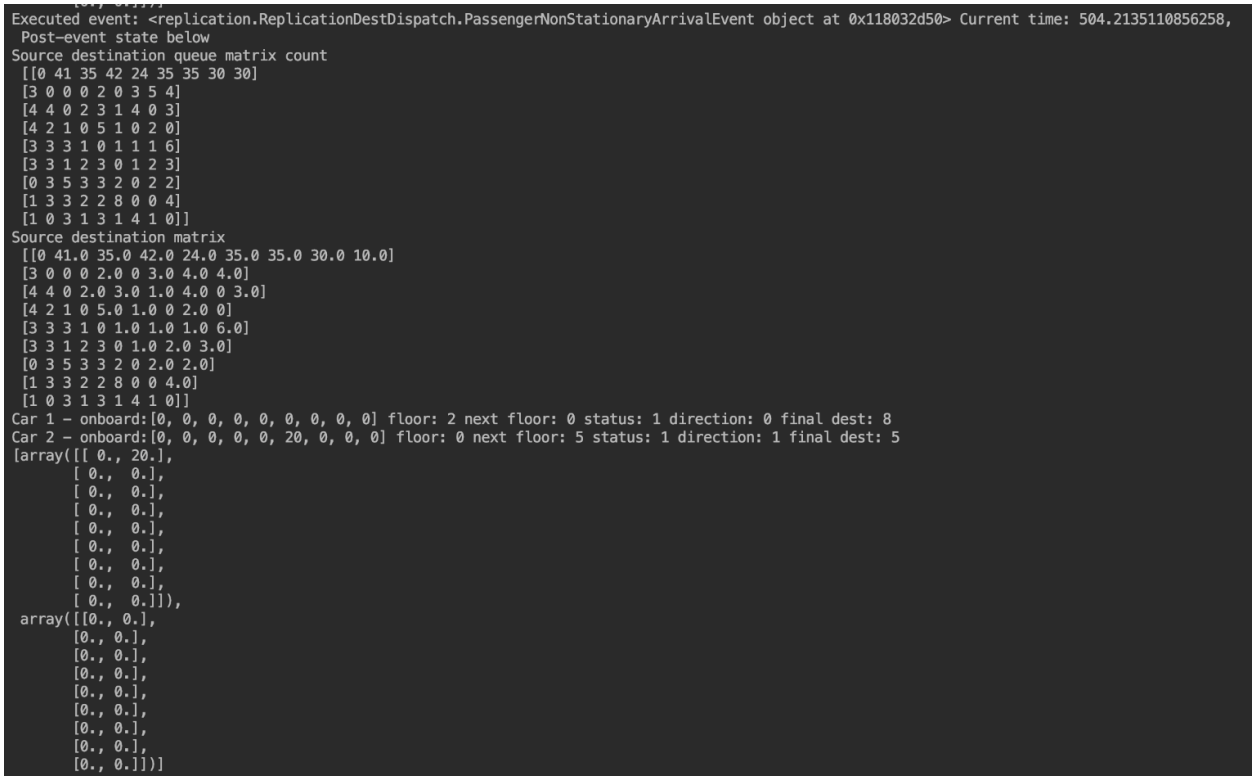
2. Destination dispatch logic is correct:

    a. Car picks the destination/ direction combination with largest number of requests, uses earliest arrival as tie-breaker



Here, floor 1 and 3 (columns 1 and 3 with zero index in the source destination queue matrix) both have 2 patrons waiting to go there. However, floor 1 has the earlier arriving patron, coming in at t=492.64814. As a result, car 1 is allocated to floor 1 as the destination, and goes to floor 0 to pick up the 2 passengers.

b. Only as many passengers a car is able to handle is allocated to each car:



As shown, car 1 is at capacity for allocations with 20 in its request array (2nd from bottom). Car 2 is also at capacity with 20 onboard (3rd array from bottom).

3. Car dynamics is correct

   a. Travel time, passenger transfer time, door open/close time

4. Passenger logic is correct

   a. Non-stationary arrival rates are correct based on schedule and building population

A3. Additional Figures

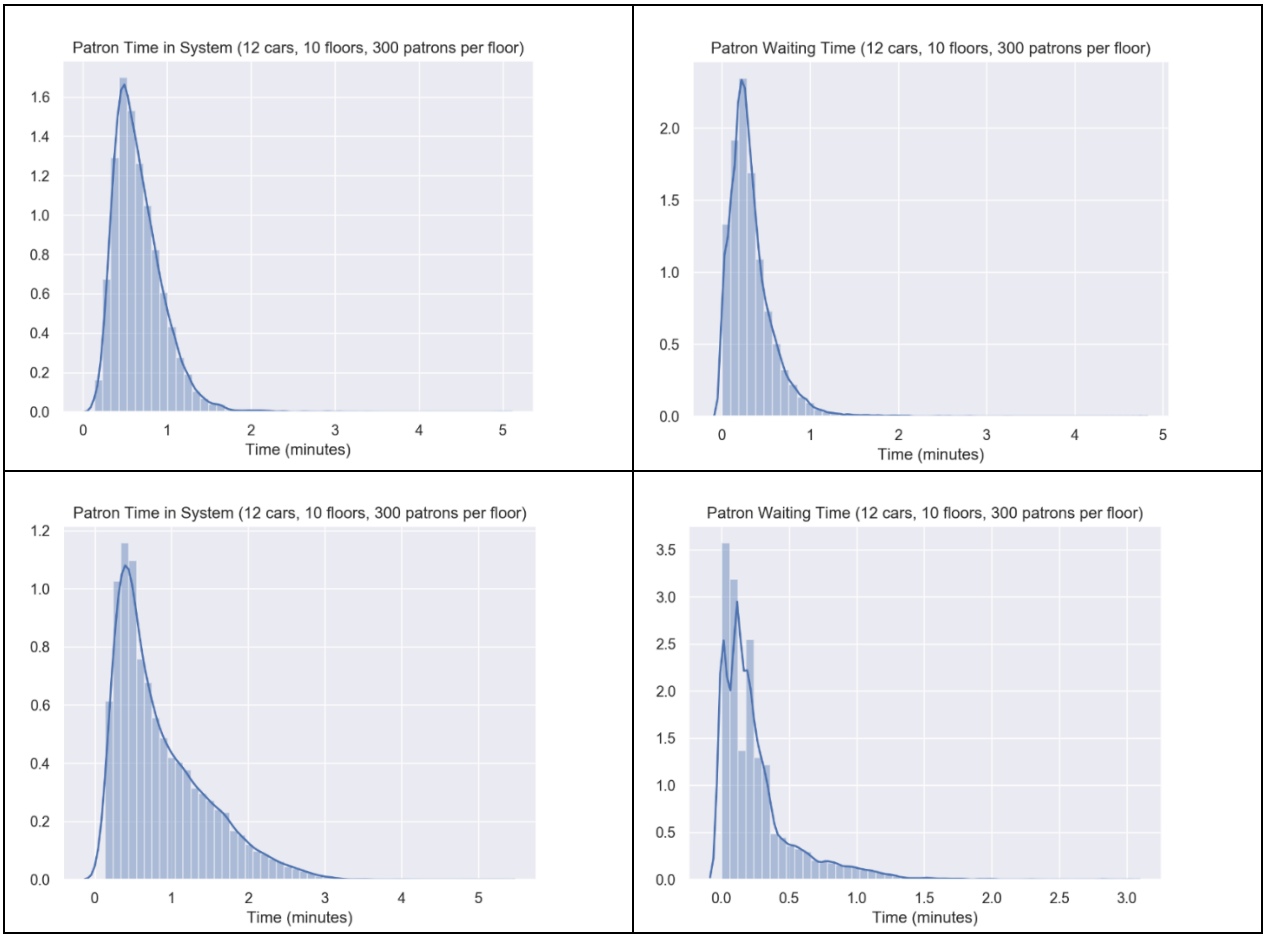


Figure. Distribution of journey time (left), and waiting time (right) for DD (top) and HUFF (bottom)

A4. Source Code

Please see https://github.com/do-ryan/elevator-simulation.