Elevator Simulation Design

CSE 6730 Project 3

Hanyuan Zhang
Computation & Computer Science
Georgia Institute of Technology
Atlanta, Georgia
unablezhy@gatech.edu

Beishen Wang
Computation & Computer Science
Georgia Institute of Technology
Atlanta, Georgia
wangbeishen@gatech.edu

Hantao Yang
Computation & Computer Science
Georgia Institute of Technology
Atlanta, Georgia
hyang320@gatech.edu

Shenwei Gao
Computation & Computer Science
Georgia Institute of Technology
Atlanta, Georgia
sgao68@gatech.edu

Xiao Han
Computation & Computer Science
Georgia Institute of Technology
Atlanta, Georgia
xhan90@gatech.edu

***Abstract*—This report presents a process based on a approach to model and simulate an elevator. The model is implemented by a queue system and simulates multiple elevators in different numbers of total floor, elevator capacity as well as requests rate. Besides, the simulation test result and performance will be based on those parameter and find out a better estimation and combination for each specific building situations.**

***Keywords*—simulation, multiple elevators, queue system, waiting time, priority queue.**

I. Introduction

In this project, we simulate a multiple elevators operation in an office building with multiple floors. By creating this model, we provide some reference to the elevator operation measures in the peak time. The efficiency of the elevator will be based on the data that collect and calculate the average waiting time as well as total passengers delivered.

II. Problem Definition

There are five main states of the elevator simulator. Firstly user presses the button to make a request. Then an elevator would be scheduled and move to pick up the user. Next user takes the elevator. Then schedule and move elevator to pick up other users. Lastly when reach the destination, user get off the elevator. The simulator has the following variables: elevators, users (requests) and floors. The variables of elevator would be: current number of passengers, current floor, current moving direction and maximum capacity. The elevator could change direction, load requests and upload requests. The variables of user would be: request and current floor. User could send request, enter elevator and take off elevator. We used random generator to generate the number of users, original floor and destination floor.

III. Simulation Programming

A) Event Class and Event handler Interface

Event class is the most basic event model. It has attributes of event type, event ID and event time. Also, it can call event handler interface to handle event. To ensure the simulator time is constituent with system time, the event class has a function to check event time.

B) Elevator Class

Elevator class creates instance of elevators for building and has following features:

- Two priority queues: goingUpRequests and goingDownRequests, consisting of requests upwards and downwards respectively;
- Elevator basic properties, including *elevator capacity* (maxCapacity), *elevator max floor number* (maxFloor), *elevator direction* (direction), *starting floor* (location), *ending floor* (headingLocation) and *real-time load* (load);
- Three condition GOING_UP, GOING_DOWN and HOLDING;
- Two major methods

  load: This method takes priority queues goingUpRequests or goingDownRequests from each floor as input, and inserts all legitimate requests into elevators' own priority queue, where legitimate means elevators remain under max capacity after requests being loaded. Meanwhile, method also keeps track of loading time in order to calculate the waiting time for each request.

  unload: This method takes each *floor instance* as input, and check elevators' two priority queues to find requests whose ending floor matches the *floor* from input, and then deletes them from queues.

C) *Floor Class*

Every floor of the building is an instance of a floor class. Floor class has two queues: goingUpRequest and goingDownRequest. goingUpRequest contains all the guests that want to go upstairs in this floor while guests that want to go down stairs are put in the goingDownRequest.

Every floor will handle the FloorEvent when a elevator comes to this floor. There are 4 cases:

If the coming event is **GOING_UP,** first the elevator will be moved to the target floor, and then the behavior of elevator depends on the conditions of request of guests in this floor and request of guests in the elevator.

- If there is no guest outsides the elevator who wants to go up in this floor or no guest insides the elevator who wants to get off the elevator, the elevator should not stop but keep going up. Therefore, it will be scheduled a new event GOING_UP and go to next floor.
- Otherwise, the elevator will be scheduled a new event LOADING, which will allow the elevator to stop at this floor and load or unload guests.

If the coming event is **GOING_DOWN**, the process is almost the same with GOING_UP, the only difference is that insteading of considering whether there are guests want to go upstairs, it will only consider whether there are guests want to go downstairs.

If the coming event is **LOADING**:

- Condition 1: If the elevator is empty, elevator will check goingUpRequest in this floor first. If is not empty, elevator will pick these guests and be scheduled a new event GOING_UP. Otherwise, it will check goingDownRequest and be scheduled a new event GOING_DOWN when it's not empty. If no guests in this floor, elevator will be be scheduled a new event HOLDING.
- Condition 2: If the elevator is not empty, the elevator will unload all the guests who want to get off the elevator in this floor. It is implemented by poll request in the elevator's queue. If all the guests get off the elevator, it will go to Condition 1. Otherwise, then, it will pick up guests who have the same direction with the elevator and be scheduled a new event GOING_UP or GOING_DOWN.

If the coming event is **HOLDING**, elevator will stop in this floor because there are no guests in this floor and elevator is empty. And when in HOLDING, elevator will check the goingUpRequest and goingDownRequest for every floor to see whether there are guests need an elevator. If find, elevator will go to pick the guest who come first. Therefore, it will be scheduled a new event GOING_UP or GOING_DOWN according to the where the guest is. If there are just no new guests coming, elevator will be scheduled event HOLDING again until there are guests need to pick up.

D) *FloorEvent Class*

FloorEvent class is implemented for airport event model. It is inherited from event class.

E) *Request Class & SimulateEngine Class*

Request class defines features including starting floor, ending floor, request load (default as 1),elevator direction and request time which is used to calculated waiting time for every single request.

SimulateEngine class is implemented with constant generation of requests to keep our simulation in process within simulation time.

IV. Simulation results and analysis

After the the implementation of the elevator simulation, we did multiple test cases to see how's the elevator performed and the efficiencies for different kind of building situations.

The output of our program are listed below, including the timestamp for each generated request, elevator number, current status, number of people loaded/unloaded, as well as starting/ending floors. At last, it will calculate the total number of passengers delivered, total number of request and average waiting time. To give us a better understanding of efficiency of the building operation, data has been collected into table and plot, which are shown below.

```
Time: 490.60 Elevator 3 Load: 14 people in the floor 9
Time: 491.10 Elevator 2 unLoad 1 people in the floor 8
Time: 491.10 Elevator 2 Load: 3 people in the floor 8
Time: 491.70 Elevator 3 unLoad 3 people in the floor 8
Time: 491.70 Elevator 3 will not accept more load!
Time: 491.70 Elevator 3 Load: 6 people in the floor 8
Time: 491.90 Elevator 1 Load: 9 people in the floor 8
Time: 492.20 Elevator 2 unLoad 8 people in the floor 9
Time: 492.80 Elevator 3 unLoad 5 people in the floor 7
Time: 492.80 Elevator 3 will not accept more load!
Time: 492.80 Elevator 3 Load: 5 people in the floor 7
Time: 493.00 Elevator 1 Load: 8 people in the floor 7
Time: 493.30 Elevator 2 unLoad 5 people in the floor 10
Time: 493.90 Elevator 3 unLoad 9 people in the floor 6
Time: 493.90 Elevator 3 Load: 4 people in the floor 6
Time: 494.10 Elevator 1 unLoad 6 people in the floor 6
Time: 495.00 Elevator 3 unLoad 8 people in the floor 5
Time: 495.00 Elevator 3 Load: 11 people in the floor 5
Time: 495.20 Elevator 2 Load: 21 people in the floor 3
Time: 496.10 Elevator 3 unLoad 10 people in the floor 4
Time: 496.10 Elevator 3 Load: 7 people in the floor 4
Time: 496.30 Elevator 1 unLoad 2 people in the floor 4
Time: 496.30 Elevator 2 unLoad 3 people in the floor 4
Time: 496.30 Elevator 2 Load: 10 people in the floor 4
Time: 497.20 Elevator 3 unLoad 10 people in the floor 3
Time: 497.20 Elevator 3 Load: 2 people in the floor 3
Time: 497.40 Elevator 1 unLoad 2 people in the floor 3
Time: 497.40 Elevator 2 unLoad 3 people in the floor 5
Time: 497.40 Elevator 2 Load: 7 people in the floor 5
Time: 498.30 Elevator 3 unLoad 13 people in the floor 2
Time: 498.30 Elevator 3 Load: 3 people in the floor 2
Time: 498.50 Elevator 1 unLoad 6 people in the floor 2
Time: 498.50 Elevator 1 Load: 1 people in the floor 2
Time: 498.50 Elevator 2 unLoad 10 people in the floor 6
Time: 498.50 Elevator 2 Load: 6 people in the floor 6
Time: 499.40 Elevator 3 unLoad 17 people in the floor 1
Time: 499.60 Elevator 1 unLoad 3 people in the floor 1
Time: 499.60 Elevator 2 unLoad 11 people in the floor 7
Time: 499.60 Elevator 2 Load: 3 people in the floor 7
Simulator stopping at time: 500.0
The total number of people loaded: 6766
The total number of requests: 6900
The total waiting time of people: 59651.7999999999
The avarage waiting time per person: 8.816405557197738
```

Fig 1: Simulation output generated from IDE

The output of the simulation project is primarily based how's your input from, which are primarily the elevator numbers, elevator capacities, number of floors, and request numbers. After multiple test and simulation, and by changing different input parameters, we come up with several graphs to indicate how the simulation model performs.

a) Number of Floors (3 Elevators + Capacity 20)

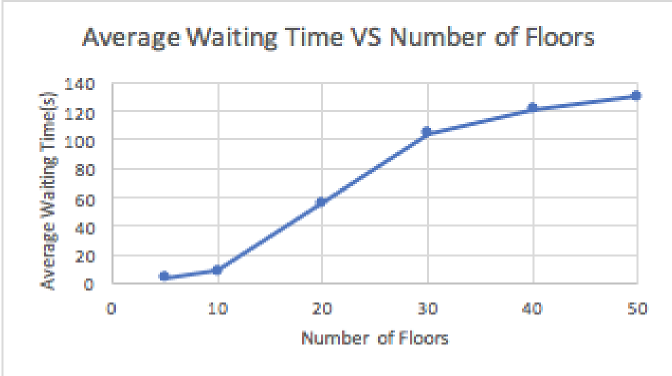

Fig 2: Relation between average waiting time with number of floors

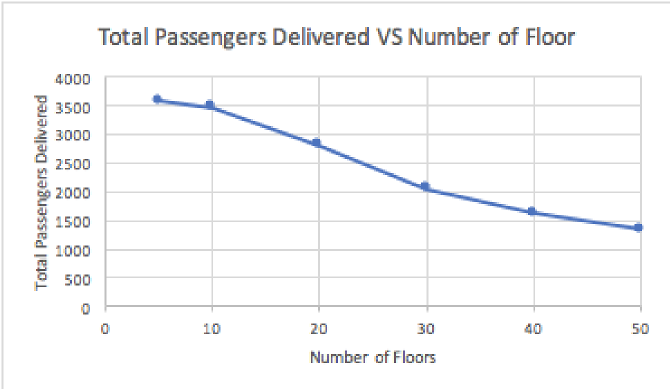

Fig 3: Relation between total passengers delivered with number of floors

As you can see from the graph above, the average waiting time of each passenger increases and total passengers delivered decreases when we increase the number of floors by 10 each time, and all three elevators are operating with each capacity of 20. This may due to the the higher the building, it takes more time for elevator to arrive at destinations floor for picking up, where it's also hard to delivers as much as passengers in a specific time (500s simulation time).

b) Number of Elevators (10 Floors + Capacity 20)

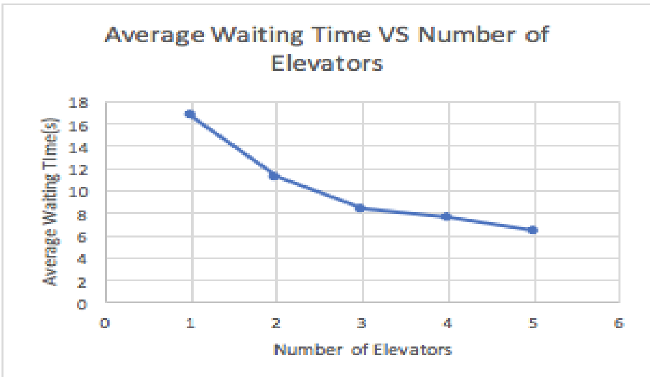

Fig 4: Relation between average waiting time with number of elevators

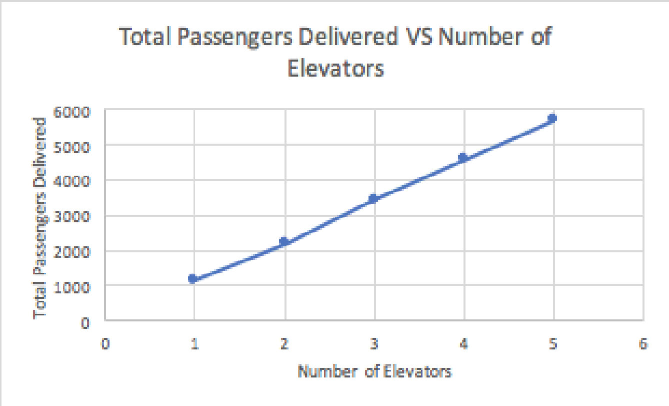

Fig 5: Relation between total passengers delivered with number of elevators

From the graph we can see the the similar result as we increase the number of elevators, and every single additional elevator would largely reduce the average waiting time for each passenger and somehow double the total passengers delivered in a specific time frame, which by implementing multiple elevators in a working building that during the peak hour, could effectively increase the efficiency of the building operation.

c) Elevators' Capacity (3 Elevators + 10 Floors)

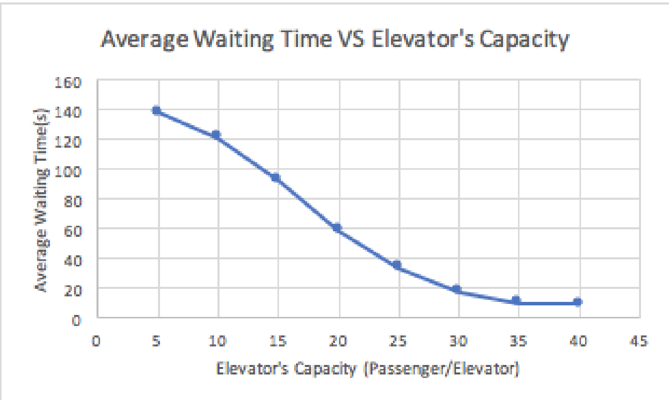

Fig 6: Relation between average waiting time with elevators' capacity

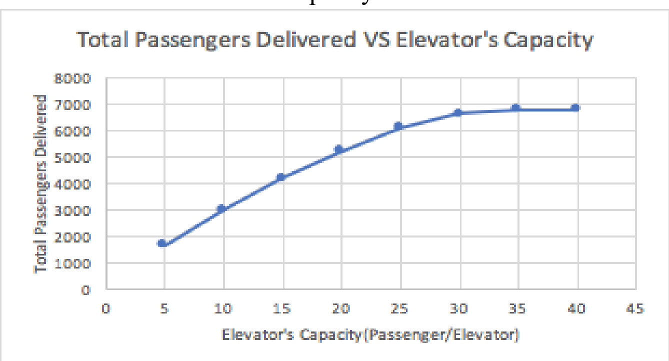

Fig 7: Relation between total passengers delivered with elevator's capacity

The graph above shows how the capacity of elevator affect the performance of the building's operation. As we allow more people to load in one elevator, obviously, the average waiting time decrease but eventually reaching close to 0, which makes sense that there is no way that the waiting time goes to zero because there is always people waiting after press the request button. In addition, since almost everyone can be loaded with larger elevator's capacity, the slope of total passengers delivered is actually decreased due to the request number is way smaller than the capacity of the elevators.

d) Comparison between 1000 and 2500 requests

In addition, as request number (people coming rate) is another dominant input parameter for our simulation model, we firstly used 1000 as our initial input to see how number of floor, elevator number and elevator capacity vary the output of average waiting time and total passengers delivered. Then we use 2500 as our total request number to see how's the result varies. Results are shown below.

The load/request was used to calculate the passing rate that how many passengers have successful get on the elevator and arrive at their destinations.

3 Elevators + Capacity 20 + Simulation Time 500s + Request 1000			
Number of Floors	Average Waiting Time	Total Passengers Delivered	Loaded/Request
5	2.797392177	997	0.997
10	5.181827309	996	0.996
15	7.342698892	993	0.993
20	9.362550607	988	0.988
25	12.86217617	965	0.965
30	16.09514964	969	0.969
35	19.95655315	969	0.969
40	21.91861702	940	0.94
45	22.16709957	924	0.924
50	29.95287107	923	0.923

10 Floors + Capacity 20 + Simulation Time 500s + Request 1000			
Number of Elevators	Average Waiting Time	Total Passengers Delivered	Loaded/Request
1	16.68693878	980	0.98
2	7.598690836	993	0.993
3	5.189669007	997	0.997

3 Elevators + 10 Floors + Simulation Time 500s + Request 1000			
Elevator Capacity	Average Waiting Time	Total Passengers Delivered	Loaded/Request
5	6.502721774	992	0.992
10	4.655755756	999	0.999

Table 1: Simulation result using 1000 requests in 500s simulation times at increasing levels of floors, number of elevators and elevator's capacity.

3 Elevators + Capacity 20 + Simulation Time 500s + Request 2500			
Number of Floors	Average Waiting Time	Total Passengers Delivered	Loaded/Request
5	3.146345382	2490	0.996
10	6.867569753	2473	0.9586
15	10.59734369	2447	0.7396
20	28.58723588	2319	0.6036
25	45.0250116	2155	0.5046
30	63.12556077	1917	0.4254
35	77.06859931	1742	0.3864
40	82.22635177	1609	0.3582
45	90.81172137	1493	0.3278
50	106.0028744	1322	0.3076

10 Floors + Capacity 20 + Simulation Time 500s + Request 2500			
Number of Elevators	Average Waiting Time	Total Passengers Delivered	Loaded/Request
1	75.06081159	1725	0.69
2	10.71416256	2436	0.9744
3	6.86440404	2475	0.99
4	5.153038229	2485	0.994
5	4.429433052	2487	0.9948

3 Elevators + 10 Floors + Simulation Time 500s + Request 2500			
Elevator Capacity	Average Waiting Time	Total Passengers Delivered	Loaded/Request
5	93.76156969	1478	0.5912
10	17.09559748	2385	0.954
15	7.408032454	2465	0.986
20	6.445193861	2476	0.9904
25	5.81614205	2478	0.9912
30	5.623429952	2484	0.9936

Table 2: Simulation result using 2500 requests in 500s simulation times at increasing levels of floors, number of elevators and elevator's capacity.

These graphs below indicates the efficiency of the elevator under different number of requests. Since both number of floors, number of elevators and elevators' capacity are dominant component that affect the efficiency of the elevator operation, then here we only consider two cases (1000 requests vs 2500 requests) to see how's the performance.

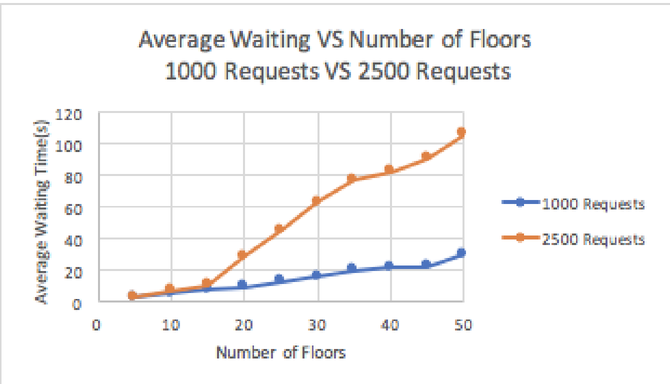

Fig 8: Average Waiting Time VS Number of Floors

As the people coming rate increase, we can see that average waiting time increases as the number of floor increase, which mean that building has reach the a maximum number of request that it can process and lots of people are waiting at the gate of elevator in which increase the average waiting time.

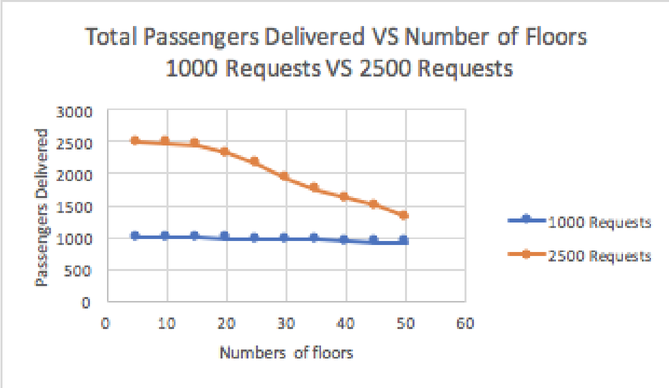

Fig 9: Total Passengers Delivered VS number of Floors

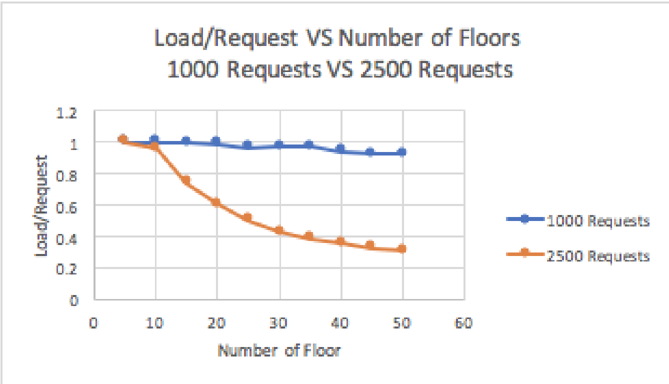

Fig 10: Load/Request VS Number of Floors

e) Peak Request/ Request Tolerance

The next test is to find out the maximum number of requests that a building can handle at a specific time in order to avoid large amount of waiting time. Results are below.

3 Elevators + Capacity 20 + Simulation Time 500s		
total request	generate frequency	avg waiting time
10	1/50s	1.3
25	1/20s	1.336
50	1/10s	1.342
100	1/5s	1.357
250	1/2s	1.4316
500	1/1s	3.164257028
1000	2/1s	4.806376518
2000	4/1s	6.029183468
3000	6/1s	7.512928402
4000	8/1s	9.681364097

Table 2: Simulation result using multiple requests in 500s simulation times at increasing total request.

Generate frequency indicates how many requests coming up in one second, which could be used to come up with different result at peak hour and night hour.

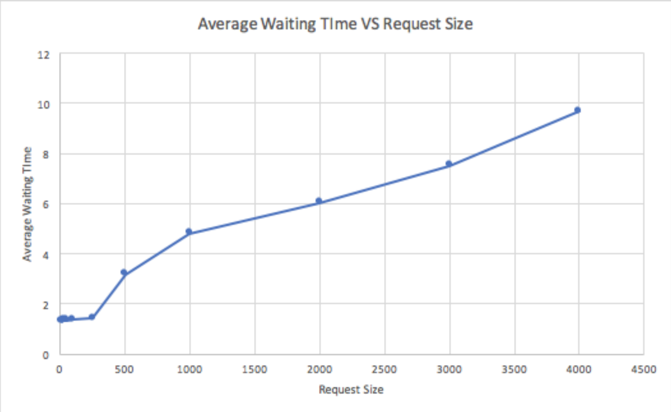

Fig 11: Relation between average waiting time and request size

The graph can indicate the a large increase at the total request around 250, which means it has reached to the limit number of request that 3 elevators with capacity of 20 can handle, people are congested at the gate which the average waiting time start to climb up. Under this model we predict and estimate the total number of elevators needed according to its peak hour request number.

V. Conclusion

Elevator simulation model is a great tool to monitor and operate a specific building according to the stream of people, building height. Our model can provide a legitimate estimate result for building manager with how many elevators that we should use, what's the capacity of each elevator. In the future study, other features for some advanced building, like limited floor rage (10 - 20 floor only), and elevator speed monitor can be implemented in our elevator simulation model.

VI. Reference

[1] Browne, James J., and James J. Kelly. "Simulation of elevator system for world's tallest buildings." *Transportation Science* 2.1 (1968): 35-56.

[2] Bartz-Beielstein, Thomas, Mike Preuss, and Sandor Markon. "Validation and optimization of an elevator simulation model with modern search heuristics." *Metaheuristics: Progress as Real Problem Solvers*. Springer US, 2005. 109-128.