Moving More With Less: Examining the Efficiency of the TWIN Elevator System in the CODA building

Chuyun Sun
chuyunsun@gatech.edu
Georgia Institute of Technology
Atlanta, Georiga

Youyi Shi
shiy7@gatech.edu
Georgia Institute of Technology
Atlanta, Georiga

Yong Jian Quek
yjquek@gatech.edu
Georgia Institute of Technology
Atlanta, Georiga

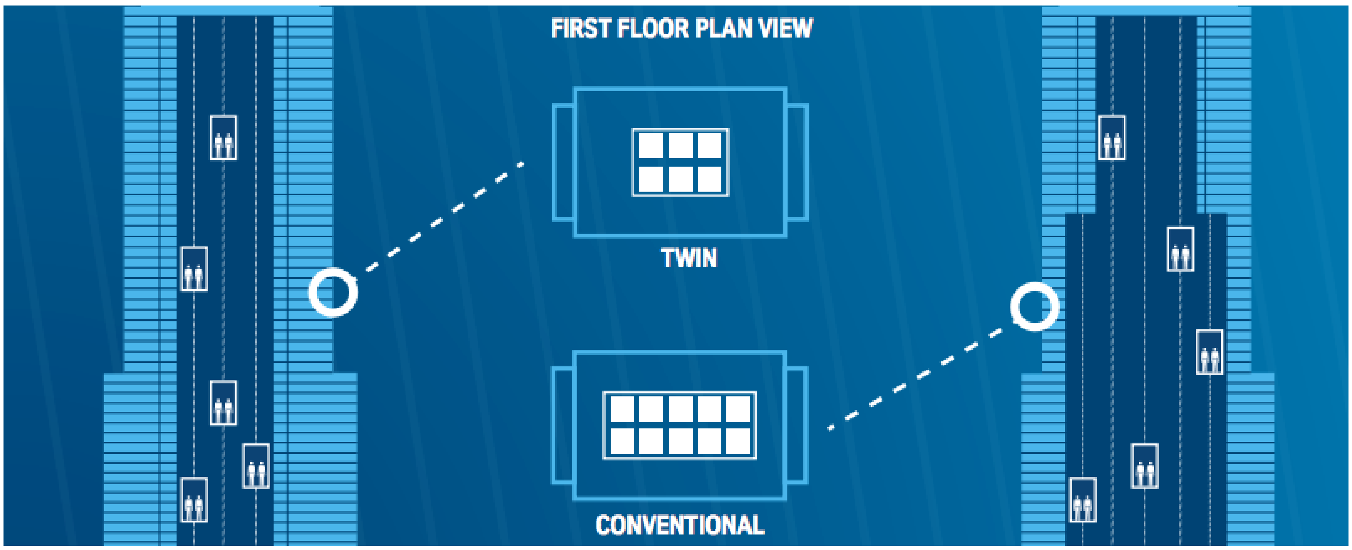

Figure 1: Example of a double cab lift system

ABSTRACT

1 INTRODUCTION

The boom in finance and technology services have seen companies and institutions cluster together in limited physical space, increasing the premium placed on land space, and with it, a rapid growth in the number of high rise buildings [2]. Moving people in high rise buildings however, remains relatively inefficient as determining the order in which to ferry people is a NP-Complete problem [9]. Multiple solutions exist to alleviate the situations, for example, introducing express elevators and skylobbies [10] or by using double-deck cabs that stop at two floors simultaneously [5]. The TWIN system by ThyssenKrupp however, utilizes two independent cabs in each elevator shaft and is advertised as being more efficient than traditional systems, both in terms of energy consumption and user throughput [11].

2 LITERATURE REVIEW

There are many studies on simulating the elevator system. Henriques, et al. used SIMIO software to simulate an elevator system with discrete event simulation in 3D animation. Their analysis of many key performance indicators, such as the time in the lift, the occupation ability of the elevator, and the waiting time for the elevator, can be used as a reference to help the group to set up the conceptual model [7]. Davidrajuh developed a MATLAB Toolbox called ElevatorSIM for discrete-event simulation for the elevator system, which can provide useful information like waiting time and service time [6]. In the terms of minimize waiting time in lift, Sliva, et al. conducted a similar research. They utilized the heuristic method to analyze queues and processors/servers [4]. The double-deck elevator system (DDES), where two cars are connected in a shaft and move simultaneously, has been also used to increase the efficiency of the elevator. Kim, et al. modelled both single-cage lift and double-cage lift using discrete event simulation in the CYCLONE system and observed a 38% decrease in lifting time during the morning peak, which agrees with the group's expectation [13]. To improve the efficiency of elevator system, Hirasawa, et al. designed an elevator group supervisory control systems(EGSCSs) which applied genetic network programming (GNP) and tested with both double-deck and single-deck elevators. With this new system, it was confirmed that DDES has reduced space and increased efficiency compare with the traditional single-deck elevator system [8]. Ding, et al. also applied genetic algorithm for the group optimization dispatching. By doing so, they successfully reduce the average waiting time, long waiting time incidence and the numbers of elevator stops [3]. Finally, Liew, et al. discussed other strategies can be useful in a multi-car elevator system, such as control strategies, car collision avoidance strategies [14].

3 PROBLEM DESCRIPTION

The TWIN elevator differs from traditional elevator systems by having two cabs in one elevator shaft. These two cabs are completely independent and can move in different directions simultaneously. The system is able to select the optimal cab to dispatch to the user as they select their destination before entering the elevator [14]. In designing an optimal system for the users, we hope to achieve these goals.

(1) Maximize Throughput of users
(2) Minimize average waiting time
(3) Minimize average time spent in the lift

This can be achieved by varying the algorithm used to dispatch the optimal cab by varying the following parameters

(1) Algorithm to assign an elevator to a user
(2) Elevator behavior when at maximum capacity?
(3) Operating range of each elevator cab

The system is also subjected to the constraint where cabs have no means of passing through each other. Thus the upper cab always remains on top of the lower cab, with the upper cab unable to reach the lowest floor and the lower cab unable to reach the highest floor.

Furthermore, stochastic elements exist within this problem as the arrival of users, as well as their destination, is of a random nature. Thus, determining an optimal algorithm for this system remains complex and in such a situation, a Discrete Event Simulator (DES) provides better insights as it mimics real world situations. This can be seen as an extension of using DES to model traditional elevator systems. [1].

4 CODA BUILDING TWIN ELEVATOR SYSTEM

The CODA building in Atlanta, GA, is the first implementation of the TWIN elevator system in North America [12] and will thus provide the parameters needed for the model. This system consists of 21 floors and 6 elevator shafts for a total of 12 cabs.

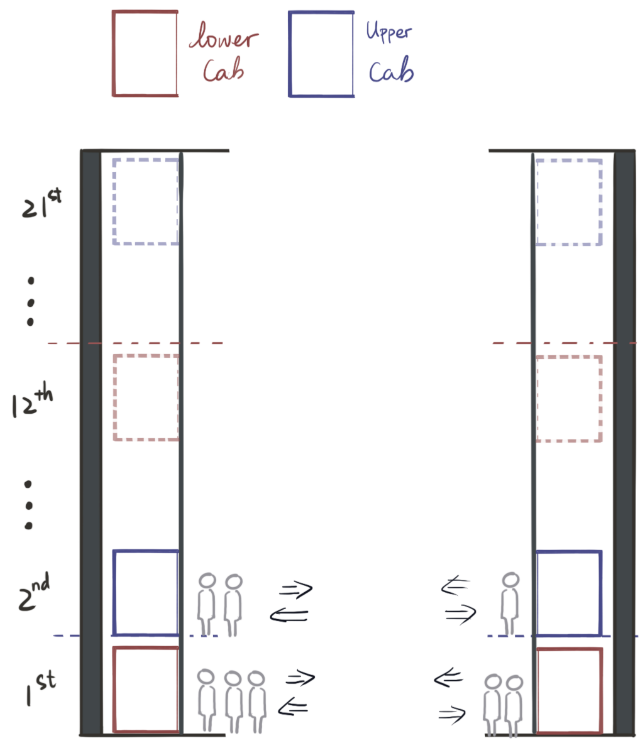

Figure 2: Elevator System in CODA building

5 CONCEPTUAL MODEL

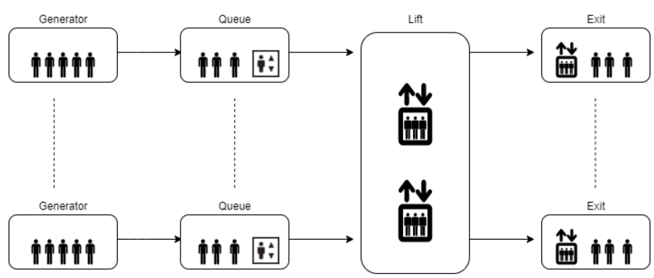

Figure 3: Conceptual model of the TWIN elevator system

5.1 Simplifications and Assumptions

- The upper cab is restricted to floors 2 to 21 and the lower cab is restricted to floors 1 to 12. This models the existing system in the CODA building.
- Each lift can only be assigned when it is completely unoccupied and has yet to be assigned.
- Each group that arrives in the model is below the maximum number of occupants allowed in the cab.
- Time taken to load and unload the users are negligible.
- If a cab is unable to reach its destination as another cab is blocking. It will wait for the other cab to reach it destination, and if it is still blocked, the other cab will be moved one level away from the original cab's destination.

5.2 Structural View

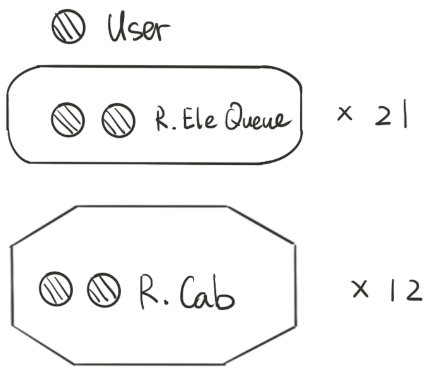

Figure 4: Structural View

5.2.1 *Entities.*

User A Consumer entity with a scope of Class. Represents a group of users (less than the maximum number allowed onto a single cab) that are heading to the same destination.

Cab A Resource entity with a scope of 12. Represents a single elevator cab.

EleQueue A Queue entity with a scope of 21. Each floor has its own EleQueue that represents the users waiting for the elevator on each floor.

5.3 Behavioural View

5.3.1 *Actions.*

UserArrival Generates a stream of users. Users are directly placed into the EleQueue of the corresponding floor with a generated destination.

UserExit User exits the system.

5.3.2 *Activities.*

Travelling Represents the user in the elevator moving from the source to the destination.

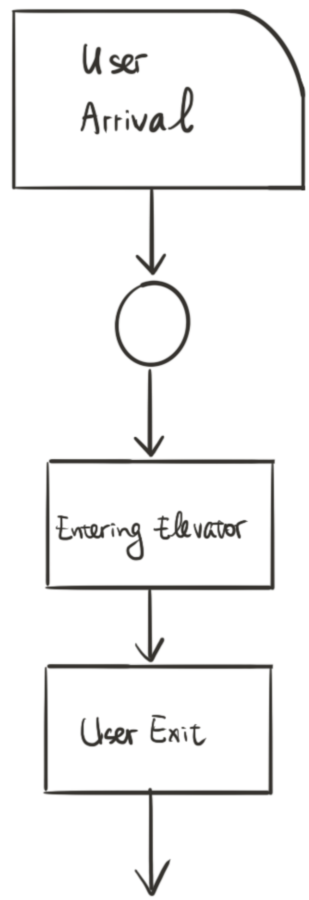

Figure 5: Behavioural View

5.4 Structural Components

Table 1: Constants

Name	Description	Value
TRAVEL_TIME	Time taken for a lift to traverse 1 floor	2s

Table 2: Consumer Class: User

Attributes	Description
startTime	Time the user arrived
waitForEle	Time spent waiting for the elevator
timeInEle	Time spent waiting in the elevator
dest	User's destination

Table 3: Resource Set[12]: Cab

Attributes	Description
curr	Indicates what floor the cab is currently on.
pos	Indicates if the cab is an upper or lower cab. 0 if lower cab, 1 if upper cab.
avail	Indicates if cab can be assigned to a user. 1 if cab is occupied or assigned, 0 otherwise.
dest	Destination of the cab if it is assigned. 0 otherwise.

Table 4: Queue Set[21]: EleQueue

Attributes	Description
q	A list of users in the queue in first in, first out order
n	Number of users in the queue

5.5 Behavioral Components

Time Units	Seconds
Observation Interval	Starts at t = 0 to t = 3600 to represent lunch hour

Table 5: Random Variate Procedures

Name	Description	Data Model
UserRandArr(src)	Returns the time to the next user arriving on floor src	To be determined from survey data
UserRandDest(src)	Returns a possible random destination for the user from the src	To be determined from survey data

Table 6: Action: UserArrival

Time Sequence	UserRandArr(User.src)
Event SCS	User.dest ⟵ UserRandDest(User.src) User.startTime ⟵ 0 User.waitForEle ⟵ 0 User.timeInEle ⟵ 0 InsertQ(EleQueue[UserRandSrc()],User)

Table 7: Action: UserExit

Time Sequence	UserRandArr(User.src)
Event SCS	User.exit()

Table 8: Activity: Travelling

Precondition	CabAssign() and User front of queue
Event SCS	User ⟵ RemoveQ(EleQueue[NextQ]) User.waitForEle ⟵ t - User.startTime + abs(User.dest - User.src) User.timeInEle ⟵ abs(Cab.curr - User.src))
Duration	TRAVEL_TIME × (abs(User.dest - User.src) + abs(Cab.curr - User.src))
Event SCS	User.waitForEle ⟵ TRAVEL_TIME × (abs(Cab.curr - User.src))

6 SIMULATION MODEL

6.1 Architecture

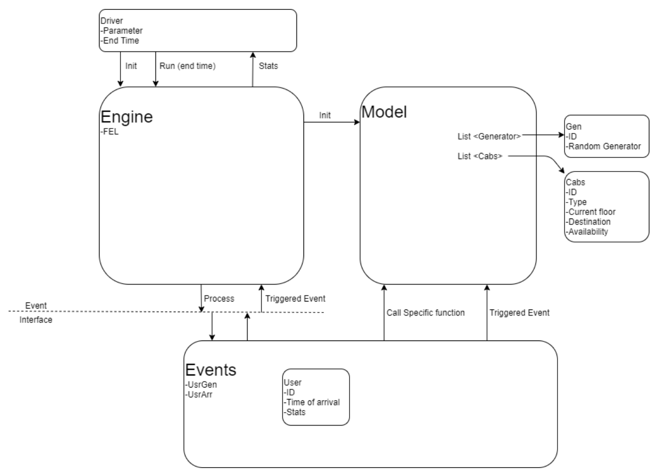

Figure 6: Elevator System in CODA building

The architecture is based on Object-Oriented Principles and utilizes polymorphism to simplify the Engine component. Below is a description of each major component.

Driver	Test driver program to run simulation. Contains hard coded parameters for now.
Engine	Main engine that consumes and schedules the future event list.
Model	Contains information on all state variables. Processes events according to state.
Events	Abstract event class with specific events extended from it. Allows for dynamic binding.
Generators	Random generators for user interarrival rates.
Cabs	Holds availability and other state information for elevator cabs

The driver will initialize the engine with the required parameters. The engine then initializes the model which builds the required components and sends the initial user list to the engine. The engine can then consume the event list and schedule incoming triggered events until the designated time.

6.2 Progress

All basic functionality of the components have been implemented. However, data collection is still ongoing and thus the generators have a placeholder probability distribution. Configuration input needs to be changed from hardcoded parameters to an input configuration file. Level bounds for cabs also need to be implemented along with a non-naive algorithm for assigning a cab. Thus the next steps are as follows.

(1) Finish data collection
(2) Implement bounded cabs and configuration file
(3) Implement non-naive alogrithm
(4) Implement random generator
(5) Validation
(6) Experiment and Results

7 TEST SCENARIOS

The test scenarios are also modeled after the existing elevators in the CODA building and thus have limited operating range to reduce the complexity of the system. The lower cab will be limited to floors 1 to 12, while the upper cab will be limited to floors 2 to 21. This can be seen in Figure 2. A special scenario exists when users depart from either the 1^{st} or 2^{nd} floor. From the 1^{st} floor, users can only reach floors 3 to 12. Users who wish to reach floors 13 to 21 must take the escalator to the 2^{nd} floor first before riding the elevator. Any elevator currently bringing users to their destinations are considered as unavailable and will not be dispatched. Thus, no elevators will be stopped midway. Any elevators en route to pick a user up is also considered unavailable unless the users are already on the same floor. We will experiment with different limits on the number of stops an elevator can make.

For this project, we will be testing two different algorithms for assigning an elevator to a user.

(1) This algorithm is based on a greedy paradigm, where the least utilized lift cab will be assigned. This means that the system will first search for the closest cab in a shaft where both cabs available. If all shafts have at least one cab in use, then the closest cab that is able to serve the user's destination is assigned. Finally, if all cabs are in use, the cab with a final destination closest to the user will be assigned.
(2) This algorithm builds on the first scenario by taking into account the destination of the user and is run independently on each elevator shaft. The distance from the selected cab from each shafts to the user is then calculated and the nearest selected cab is assigned.
  (a) If no cabs are available, the user cannot use an elevator from this shaft.
  (b) If the user is on the 1^{st} floor, the user must use the lower cab.
  (c) If the user is on the 2^{nd} floor, the user must use the upper cab to go above the 13^{th} floor.
  (d) If the user is above the 2^{nd} and is above both the upper cab and the lower cab. The user will take the upper cab as seen in 7A.
  (e) If the user is in between both cabs as seen in Figure 7B:
    (i) If the user is between the 3^{rd} floor and 12^{th} floor, and the destination is between the 2^{nd} and 12^{th} floor, the user will take the closest cab.
    (ii) If the user is above the 12^{th} floor, the user will take the upper cab.
    (iii) If the user is above the 2^{nd} floor and the destination is above the 12^{th} floor, the user will take the upper cab.
  (f) If the user is below both cabs as seen in Fig.7C:
    (i) If the user is below the 13^{th} floor, the user will take the lower cab
    (ii) If the user is above than 12^{th} floor, the user will take the upper cab.

Table 9: State Variables in Model

Variable	Value
La	Current Location
D	Destination
L'	Location of the Lower Cab
U'	Location of the Upper Cab
L	Lower Cab
U	Upper Cab

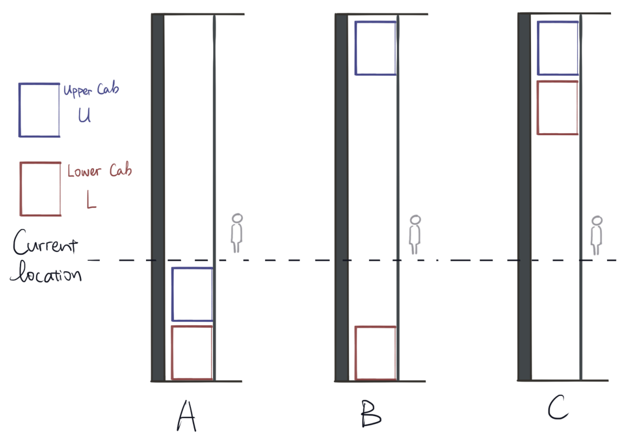

Figure 7: Test scenario

8 DATA COLLECTION

Data needs to be collected for the two stochastic elements, the user interarrival rate and the user's destination. We will attempt to get the data directly from the CODA management or from ThyssenKrupp themselves. Should they not releast the information to us, we will conduct a physical survey in the building to determine the interarrival rate. We will conduct a personal survey of users to determine the distribution of user's destination.

Algorithm 1 Elevator Assignment: Greedy-Improved Algorithm

```
function AssignElevatorCab(La, D, L′, U′, U, L)
    Cab ← ∅
    if La == 1 or D == 1 then
        Cab ← L
    end if
    if La == 2 then
        Cab ← U
    end if
    if L' < U' < La then
        Cab ← U
    end if
    if L' < La < U' then
        if La <=12 and D <= 12 then
            Cab ← ClosestCab
        else
            Cab ← U
        end if
    end if
    if La < L'< U' then
        if La <= 12 then
            Cab ← L
        else
            Cab ← U
        end if
    end if
    return Elevator
end function
```

9 PROJECT DISTRIBUTION

Table 10: Project Distribution

Member	Report	Code
Chuyun	Literature Review, Conceptual Model	Engine, Driver
Youyi	Literature Review, Simulation Model	Component, Events
Yong Jian	Conceptual Model, Simulation Model	Model, OOP Design, Statistics

REFERENCES

[1] Lutfi Al-Sharif and Mohamed D Al-Adem. 2014. The Current Practice of Lift Traffic Design Using Calculation and Simulation. *Building Services Engineering Research & Technology* 35, 4 (jul 2014), 438–445.

[2] Martin Neil Baily and Nicholas Montalbano. 2017. *Clusters and Innovation Districts: Lessons from the United States Experience.* Technical Report. Washington, D.C.

[3] Qing-chao Li Jin Zhang Bao Ding, a and Xiao feng Liu. 2013. Twin elevator group optimization dispatching based on genetic algorithm. *Applied Mechanics and Materials* 415 (Sep 2013), 95–100.

[4] Luiza R. Vitor-Daywes Pinheiro Neto Leandro Kazu Wesley Pacheco Calixto Clebes Andre Silva, Danilo Fernando A. Silva. 2017. Simulation and minimization of waiting time in rows of elevators of public buildings. *2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)* (Oct 2017), 1–6.

[5] Otis Elevator Company. 2001. *Burj Khalifa.* Retrieved January 18, 2020 from https://www.otis.com/en/hk/projects/showcase/burj-khalifa/

[6] Reggie Davidrajuh. 2019. Developing a Toolbox for Modeling and Simulation of Elevators. *Int. J. Simul. Syst. Sci. Technol* 20 (Mar 2019), 1–1.

[7] Luís M.S Dias ; Guilherme A.B. Pereira Marcelo F.N. Henriques, António A.C. Vieira and José A. Oliveira. 2019. ANALYSIS OF AN ELEVATOR SYSTEM USING DISCRETE EVENT SIMULATION: CASE STUDY. *International Journal for Quality Research* 13, 4 (Dec 2019), 823–836.

[8] Kotaro Hirasawa ; Toru Eguchi ; Jin Zhou ; Lu Yu ; Jinglu Hu ; Sandor Markon. 2008. A Double-Deck Elevator Group Supervisory Control System Using Genetic Network Programming. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38, 4 (2008).

[9] Daniel Nikovski and Matthew Brand. 2003. Decision-Theoretic Group Elevator Scheduling. In *Proceedings of the Thirteenth International Conference on International Conference on Automated Planning and Scheduling.* AAAI Press, Palo Alto, CA, 133–142.

[10] Amber C. Snider. 2017. *Paying Homage to the Past: The Original Twin Towers.* Retrieved January 18, 2020 from https://theculturetrip.com/north-america/usa/articles/paying-homage-to-the-past-the-original-twin-towers/

[11] ThyssenKrupp. 2016. *TWIN from ThyssenKrupp Elevators.* Retrieved January 18, 2020 from https://twin.thyssenkrupp-elevator.com/home

[12] ThyssenKrupp. 2019. *Coda building in Atlanta, featuring revolutionary TWIN elevators from thyssenkrupp, now open to tenants.* Retrieved January 28, 2020 from https://www.prnewswire.com/news-releases/coda-building-in-atlanta-featuring-revolutionary-twin-elevators-from-thyssenkrupp-now-open-to-tenants-300856254.html

[13] Taehoon Kim Seung Woo Kim Hong-Ku Jung Hunhee Cho Wansoub Kim, Dongmin Lee and Kyung-In Kang. 2016. Evaluation of lifting efficiency of double-cage construction lift in supertall building construction. *SARC 2016 - 33rd International Symposium on Automation and Robotics in Construction* (2016), 787–795.

[14] Michael Loong Peng Tan-Chee Wei Tan Yeong Cherng Liew, Cheng Siong Lim. 2015. A review of multi-car elevator system. *Jurnal Teknologi* 73, 6 (2015).