ME733 Course Project Report

Up-peak traffic dispatching Control policy for elevator systems based on Markov Queueing theory

Yuan Zhou

10-May-2017

Abstract

In this course project report, we introduce an optimal dispatching control policy for elevator systems during up-peak traffic. Using Markov Dynamic Programming decision method and modeling the single elevator system as a $\mathrm{M}/M^k/1$ queue, we show that a threshold policy is the optimal policy to minimize the average waiting time in single elevator system. Several MATLAB simulations for both one elevator and two elevators systems are included to show the policy in other view and compare the optimal threshold policy to some constant threshold policy to show the optimality. Finally, modify the elevator service process by considering its relationship with the arriving process and introduce a threshold plus separated service floors policy for multi-elevators dispatch system.

1. INTRODUCTION

Nowadays, the elevator system has been an important vertical transportation device for tall buildings, and has brought human convenience and efficiency in the recently decade years. In addition; because of the height of office buildings and high-rise residences, the requirements of the elevator system control are getting higher and higher. There are several performance indicators to evaluate the elevator performance and in this report our goal is to minimize the average passenger waiting time in the main lobby.

Due to the complexity of the elevator system, mainly because of the huge structure and amounts difference of the passengers' flow, it is difficult to build a generate model to analyze and optimize the performance. A general method to analyze the elevator dispatching control problem is to decompose passenger traffic into three different models: 1) up-peak traffic; 2) down-peak traffic; and 3) interfloor traffic [1]. The up-peak traffic represents the start of the business day in an office building. The main (or all) of the passenger flow is the upstream direction and starts at the main lobby.

In this course project, we focus on the up-peak traffic situation and develop a policy for optimal dispatching. In a purely upward traffic pattern, the only control actions that we can decide is the time to close the door and dispatch the elevator, and these decisions affect how many passengers will board the elevator. Once the elevator is dispatched, no more control action we can do before the elevator comes back. Therefore, for the up-peak traffic mode, the control question or policy is to decide when to dispatch the elevator based on the information we can collected.

Intuitively, the simplest policy (decision) is to dispatch the elevator whenever a passenger got in the elevator, which is the common and unconscious decision we made in our daily life. The other policy is half-capacity plus time-out policy. It dispatches an elevator whenever half its capacity is reached or the time of the elevator waiting at main lobby is longer than a given time duration. Although the above two control decisions look simply and straightforward, they actually unmask part of the solution of this course report.

The organization of this course report is, first, based on the reference paper [2], to introduce that to minimize the average passenger waiting time for up-peak traffic, the optimal dispatching policy is a threshold policy, which means dispatch the elevator whenever the number of people in the elevator reaches or exceeds a certain threshold. The threshold is not fixed and dependent on the passenger arrive rate, elevator arrive rate and parameters we set for the cost structure. Second, do serval MATLAB simulations to show the optimality of a threshold policy. Third, go a step further, modified the model that are introduced and do some simulation to show it.

2. PROBLEM FORMULATION

In this section, we model the problem as a simple queue model. The reference paper [2] treats the system with two elevators. In this course project report, to simplify the problem, we model the elevator system as a single server batch queueing system.

We consider the queueing model as a $\mathrm{M}/M^k/1$ queue which k is the threshold that we want to decide. The passengers arriving time and the elevator coming back time is not deterministic, and we set that passengers arrive one at a time to the queue according to a Poisson process with rate λ, and it could be a constant or a variable. Each passenger arrive generates a pa event. This queue served by a server (elevator) which has a finite capacity of C(C>=k) passengers.

The time of dispatching the elevator is depended on the dispatching control decision which represents the threshold-based policy. The time for the elevator to serve a batch of passengers and come back to main lobby empty is exponentially distributed with rate μ, a constant. This completion of service generates a “car arrive” (ca) event.

3 SOULTION APPROACH

Now introduce Markov Decision Problem (MDP) to show the threshold policy of the single elevator system during up-peak traffic.

3.1 State space

The state space for this queue model is a two-dimension state space. The state space is $\mathrm{X} = \{(y,z): y = 0,1\ldots, z = 0,1\}$. $\mathrm{y}(t) \in \{0,1\ldots\}$ denotes the queue length at the first-floor lobby at time t and $\mathrm{z}(t) \in \{0,1\}$ denotes the number of elevator at the first floor which is available to dispatch.

3.2 Control action

The state transition of this model is the result of the pa or ca event happened. Whenever these events happened and the elevator is available, we need to decide whether dispatch the elevator or not. we define $\mathrm{U} = \{0,1\}$, which u=0 means holding the elevator and u=1 means dispatching the elevator. Notes that not all control actions are reachable at every state x. A set of admissible control actions $\mathrm{U}(y,z)$ needs to define and we have $\mathrm{U}(y,0) = \{0\}$

and $\mathrm{U}(y,1) = \{0,1\}$. The first item means no available elevator in the first floor, so the only control action is holding the elevator and the second item means that we do either hold the elevator or dispatch it when it is available. Next, a max-pulse equation is defined which is related to the dispatching control action. We define $[y - C]^+ = max\{y - C, 0\}$. It means that dispatching the elevator serves $\min\{y, C\}$ passengers because the capacity of the elevator is C, and leaves the main lobby queue of length $[y - C]^+$passengers.

3.3 State transition probability

Use uniformization technique to convert continuous-time MDP into and equivalent discrete-time MDP. We choose uniform rate $\gamma = \lambda + \mu$ which is the sum of the total event rate. We define $p_{ij}(u)$ as the conditional probability that the state at the next time step is j ϵ X given that the current time state is i ϵ X and the control action taken at the beginning of the current time step is u ϵ $\mathrm{U}(i)$. $p_{ij}(u)$ are given by

$$p_{ij}(0) = \lambda, i = (y, 0), j = (y + 1,0) \qquad (1a)$$

$$p_{ij}(0) = \lambda, i = (y, 1), j = (y + 1,1) \qquad (1b)$$

$$p_{ij}(1) = \lambda, i = (y, 1), j = ([y - C]^+ + 1,0)(1c)$$

$$p_{ij}(0) = \mu, i = (y, 0), j = (y, 1) \qquad (1d)$$

$$p_{ij}(0) = \mu, i = (y, 1), j = (y, 1) \qquad (1e)$$

$$p_{ij}(1) = \mu, i = (y, 1), j = ([y - C]^+ + 1,1)(1f)$$

For each$(1a) - (1f)$, the first three items represent state transitions induced by a pa event. The last three items correspond to a ca event. Notes that in $(1e)$ the ca event is fictitious event because the elevator has already waited in the main lobby and in $(1f)$ the ca event is also fictitious event because it is the control action taken at the beginning of the time step that causes the state change, and not the fictitious ca event.

3.4 State transition cost structure

We define the cost for the Kth time step by $\mathrm{C}(xk, uk)$ where uk is the control action taken at the beginning of the time step when the state is $xk = (yk, zk)$. After that, what we are looking for is an optimal stationary policy say π^* that minimizes the total discounted cost to be incurred over the infinite-horizon. We define

$$V_{\pi^*}^{\alpha}(i) = \inf_{\pi} E_\pi \left[\sum_{k=0}^{\infty} \alpha^k C(yk, zk, uk) | x_0 = i \right] (2)$$

notes that $E[.]$ represents the expectation operator, and $\alpha(0 < \alpha < 1)$ is a given discount factor.

For little's law [1], we know that the average queue length is proportional to the average waiting time. Therefore, we can assume that the cost of each time step is proportional to the queue length after control decision, and set β to bound the cost structure. We define

$$\mathrm{C}(y,0,u) = \beta y \qquad 3(a)$$

$$\mathrm{C}(y,1,u) = \beta[y-C]^{+} \quad u=1 \qquad 3(b)$$

$$C(y,1,u) = \beta y \quad u=0 \qquad 3(c)$$

3.5 Dynamic programming(DP) equation

$V_{n+1}^{\alpha}(i)$ represents from state i, the optimal cost after n time steps. Our cost structure satisfies that the one-step costs (3) are nonnegative, discounted parameter $\alpha\epsilon(0,1)$ and the control option set is finite. the dynamic programming algorithm can be represented by

$$V_{n+1}^{\alpha}(i) = min\left[C(i,u) + \alpha\sum\nolimits_{j} p_{ij}(u)V_{n}^{\alpha}(j)\right] (4)$$

Because the initial state cost $V_{0}^{\alpha}(i) = 0$, the above equation converges to an optimal function value

$$V_{\pi^{*}}^{\alpha}(i) = \lim_{n\to\infty} V_{n}^{\alpha}(i) \; (5)$$

Therefore, we can get a stable policy

$$u^{*}(i) = argmin\left[C(i,u) + \alpha\sum\nolimits_{j} p_{ij}(u)V_{n}^{\alpha}(j)\right] (6)$$

Using state probability equations (1a-1f) and one step cost equations (3a-3c), we can decompose the equation (4) in each time step.

First, for states of the form $(y,0)$, the admissible action is $u=0$. We have

$$V_{n+1}^{\alpha}(y,0) = \beta y + \alpha\lambda V_{n}^{\alpha}(y+1,0) + \alpha\mu V_{n}^{\alpha}(y,1) \; (7a)$$

In (7a), the first term is the one-step cost, the second term represents to a pa event, and the third terms represents to a ca event.

Second, for states of the form $(y,1)$, the elevator is available and there are two control actions which are waiting for more passengers to arrive or dispatch the elevator, so for the equation, we have

$$V_{n+1}^{\alpha}(y,1) = min\begin{Bmatrix} \beta y + \alpha\lambda V_{n}^{\alpha}(y+1,1) + \alpha\mu V_{n}^{\alpha}(y,1) \\ \beta[y-C]^{+} + \alpha\lambda V_{n}^{\alpha}([y-C]^{+}+1,0) + \alpha\mu V_{n}^{\alpha}([y-C]^{+},1) \end{Bmatrix} (7b)$$

The first term in (7b) corresponds to holding the elevator, and the second corresponds to dispatching the elevator. Notes that the ca event in the first item is fictitious event.

3.6 Show threshold policy

After we define the state space, control action, state transition probability, transition cost structure and dynamic programming equations of the one elevator system, we can prove the existences of the threshold with following two steps:

a. Simplify the DP equation

We define

$$A_{0,n}(y) = \beta y + \alpha\lambda V_n^\alpha(y+1,0) + \alpha\mu V_n^\alpha(y,1)$$

$$A_{1,n}(y) = \beta y + \alpha\lambda V_n^\alpha(y+1,1) + \alpha\mu V_n^\alpha(y,1)$$

$$B_{1,n}(y) = \beta[y-C]^+ + \alpha\lambda V_n^\alpha([y-C]^+ + 1,0) + \alpha\mu V_n^\alpha([y-C]^+,1)$$

$$A_0(y) = \lim_{n\to\infty} A_{0,n}(y)$$

$$A_1(y) = \lim_{n\to\infty} A_{1,n}(y)$$

$$B_1(y) = \lim_{n\to\infty} B_{1,n}(y)$$

so that (7a) and (7b) are rewritten as

$$V_{n+1}^\alpha(y,0) = A_{0,n}(y) \qquad (8a)$$

$$V_{n+1}^\alpha(y,1) = min\{A_{1,n}(y), B_{1,n}(y)\} \qquad (8b)$$

and the optimality for states of the form (y,0) and (y,1) becomes

$$V_{\pi^*}^\alpha(y,0) = A_0(y) \qquad (9a)$$

$$V_{\pi^*}^\alpha(y,1) = min\{A_1(y), B_1(y)\} \qquad (9b)$$

b. Find the relationship of $A_1(y)$ and $B_1(y)$, and get the threshold result

Set $F(y) = A_1(y) - B_1(y)$, $\Delta(y) = F(y+1) - F(y)$

When $0 \le y < C$, $[y-C]^+ = 0, [y+1-C]^+ = 0$

So $\Delta(y) = \begin{Bmatrix} [\beta(y+1) + \alpha\lambda V_n^\alpha(y+2,1) + \alpha\mu V_n^\alpha(y+1,1)] \\ -[\beta[y+1-C]^+ + \alpha\lambda V_n^\alpha([y+1-C]^+ + 1,0) + \alpha\mu V_n^\alpha([y+1-C]^+,1)] \end{Bmatrix} -$

$$\begin{Bmatrix} [\beta y + \alpha\lambda V_n^\alpha(y+1,1) + \alpha\mu V_n^\alpha(y,1)] \\ -[\beta[y-C]^+ + \alpha\lambda V_n^\alpha([y-C]^+ + 1,0) + \alpha\mu V_n^\alpha([y-C]^+,1)] \end{Bmatrix}$$

$$= \beta + \alpha\lambda[V_n^\alpha(y+2,1) - V_n^\alpha(y+1,1)] + \alpha\mu[V_n^\alpha(y+1,1) - V_n^\alpha(y,1)]$$

It is easy to show that $V_n^\alpha(y+2,1) > V_n^\alpha(y,1)$ and $V_n^\alpha(y+1,1) > V_n^\alpha(y,1)$, which means the cost of the longer queue is bigger than the shorter queue. Therefore, we have $\Delta(y) > 0$, and F(y) is strictly increased. Notes that when y=0, which means queue length is 0, we have no reason to dispatch the elevator. In other word, $A_1(0) < B_1(0)$. When y>C, which means queue length is larger than C, we have to dispatch if the elevator is available. In other word, $A_1(y) > B_1(y)$ when y>C. Combined the property that F(y) is strictly increased, there must exist a y that F(y) equals to 0 and y is a threshold.

So far, we have generated the threshold policy for our one elevator up-peak system. A detailed theorem and policy for multiple-elevators system is showed in the reference paper [2] which is also a threshold type policy and have N threshold corresponded to N elevators.

4 MATLAB SIMULATION

In this section, we present serval examples to show the threshold policy and its optimality through MATLAB simulation. First simulation is a one-elevator with different passenger arrive rate to show the optimal thresholds, and there can be a performance penalty when the thresholds are chosen arbitrarily. Second simulation is a two-elevators case use standard clock method [3] to show the optimal threshold-based policy in multi-elevators system.

4.1 One-elevator system

In this MATLAB simulation, we develop a discrete-event simulator of a one-elevator system. In this process, passengers arrive one at a time according to a poison process with rate λ. In order to make a clear visualization of the threshold policy, we set λ as a variable from 2/15 to 9/15. For the elevator, the service time is a random variable drawn from an exponential distribution with a rate μ=1/15. The elevator has a capacity of C=15, and the simulation process are as follow: For each different arrive rate simulation there are 15 thresholds we will simulate and find the optimal threshold for each one and compared the threshold one and threshold 7 (half capacity).

Fig.1 shows the response surface. We treat the passengers' average main lobby waiting time as performance value and is on the z axis. The passenger arrive rate is on the x axis. The threshold is on the y axis. Each point on the plot is obtained by averaging the passengers main lobby waiting times over ten simulation runs, with each simulation run length is 10000 passengers.

The block dots show the optimal thresholds for each different arrive rates. The blue dots represent the threshold one policy and the green dot represent the threshold seven (half capacity) policy. From the trajectory of the block dots we can find that with the enlargement of the arrive rate, the optimal threshold becomes bigger and bigger. This is reasonable because when intensive passengers arrived, if the elevator waits a little bit more time at the main lobby, although it will increase the time of the passengers who have already arrived, it will decrease next passengers' waiting time greatly. We can also find that there are significant performance penalties for the two constant threshold policies. For the half-Capacity policy, when arrive rate is relatively small, it performance is really bad. In this simulation, when arrive rate is 2/15, the optimal average waiting time is 12.2946 unit time and its waiting time is 18.7798 unit time. About 50% longer than the optimal wait. For the threshold one policy, when arrive rate is relatively big, it performance is relatively not good. When arrive rate is 9/15, the optimal average waiting time is 19.4281 unit time and its waiting time is 23.1975 unit time. About 23% longer than the optimal wait

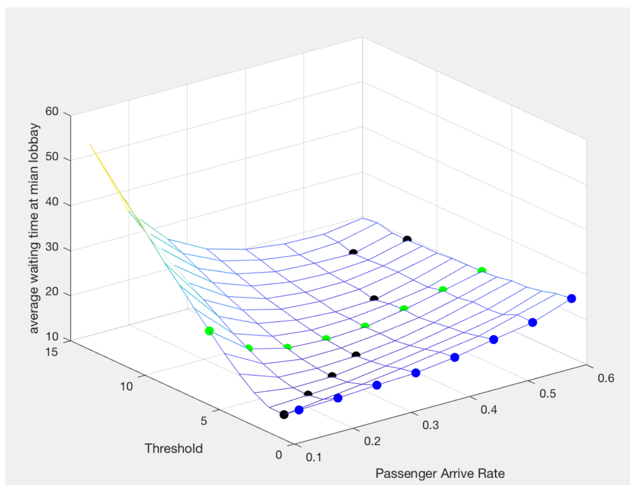

Fig.1. Response surface for a pa rate from 2/15-9/15

.

4.2 Two-elevators system

For this example, we extend our system to the two-elevators, which is the start of the multi-elevators dispatch system. Based on the theoretical prove in the reference paper [2]. It is still a threshold-based policy to get optimal average waiting time. However, in this case we have two thresholds to optimize.

For this two-elevators system, a standard clock method is used to do the efficient simulation. The details of the standard clock method can be found in [1], [3]. Each elevator has a capacity of C=10 passengers. We have two thresholds θ_1 and θ_2. The control policy works as follows: when one elevator is available and the queue length $y \geq \theta_1$, then up to C=10 passengers are immediately loaded and one elevator is dispatched. Otherwise up to

2C=20 passengers are loaded and dispatched. We treat the service time is a random variable drawn from an exponential distribution with a rate of $\mu = 1/10$ for both elevators. For each threshold couple, we run simulation 10 runs and in each run our simulation length is 8000 passengers

We do two simulation examples with different arrive rate. For the first we set passengers arrive process as a Poisson process at rate $\lambda = 1/12$. This is an interesting case because in this case the intensity of the arrive rate is smaller than the service rate. Intuitively the best threshold policy is (1,1) which means dispatching the elevator as soon as a passenger arrived. Fig.2 shows the performance surface and it also shows that the (1,1) threshold couple is optimal.

Second one is kind of a normal example, we set passengers arrive process as a Poisson process at rate $\lambda = 1/3$. From Fig.3, the optimal threshold couple is (5,4) and the optimal value is 27.4574-unit time. The 'bowl' shape of the response surface represents a potentially substantial penalty for the wrong threshold couple.

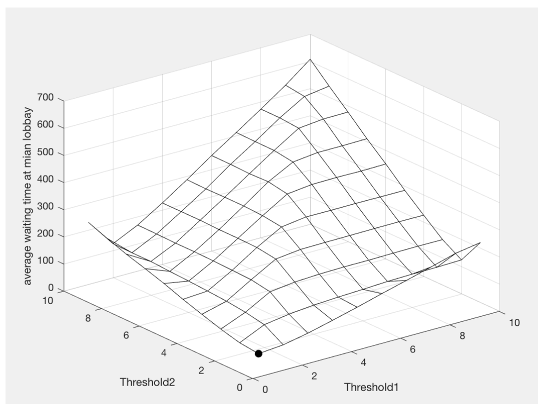

Fig.2. Response surface for a pa rate =1/12

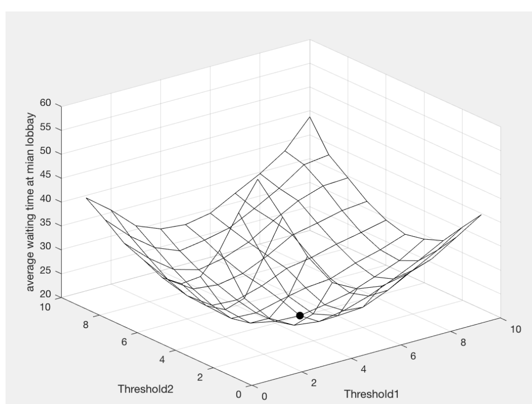

Fig.3. Response surface for a pa rate =1/3

5 MODIFIED MODEL AND SIMULAITON

In this part, we go a step further to modify the model of the elevator system. We will do two modifications. First, for one-elevator system, we modify the serve process that do not treat the rate as a constant but a variable relative with the passenger number and the building floors. Second, for the two-elevator system, we introduce a threshold plus separated serve floors policy to improve the performance in certain circumstance.

5.1 Serve process modification

The reference paper [2] treats the serve time as a random variable with exponential distribution of a *constant* rate. This is a doable option in the circumstance that we do not have enough information about the serve process. However, if we decompose the serve process, it basically has two parts which are elevator moving and passenger leaving, and they have relationship between the passengers in the elevator and the total building floors.

Actually, in elevator industry area, there exists a terminology called Round-Trip Time (RTT) [4] that represents this kind of service process.

$$\mathrm{RTT} = 2\mathrm{H}t_v + St_s + 2pt_p (10)$$

In which, H-Max arrive floor; S- expectation number of stop time; p-one trip passenger number; t_v- elevator running time between floors; t_s-elevator stop time; t_p-one passenger transmit time.

In the modified model, we treat the serve rate $\mu = \frac{1}{RTT}$ to simulate the one-elevator system. In addition, we need information to get H and S. we assume the total building floor is 5 and for each trip of passengers their destination floor is a uniform distribution. We treat that the expectation number of stop time S equals to half of the max arrive floor. Based on my test from my home which is a 9-floor residence, we assume the $t_v = t_s = 2t_p = 1$ unit time. For the following MATLAB, we set arrive rate $\lambda = \frac{1}{4}$, and service rate $\mu = \frac{1}{RTT}$. The capacity C=15. For each threshold, we do twenty simulation to get the average waiting time at the main lobby. For each run, the simulation length is 10000 passengers. Fig.4 is the curve with different threshold. It still shows the existence of an optimal threshold.

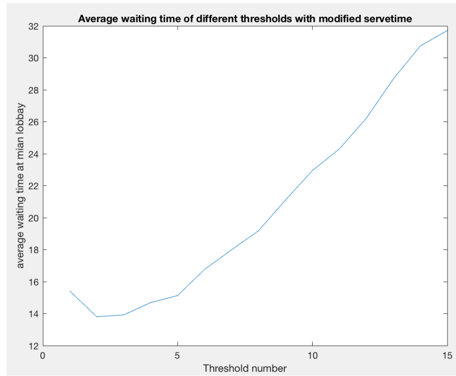

Fig.4. performance curve of a pa rate=1/4

5.2 Threshold plus separated serve floors policy

For this modified model, we focus on the two-elevators dispatch system and extensions to the N>2 elevators case can also follow in this method. In the previous two-elevators model, the reference paper shows [2] the policy and we simulate that the system exists an optimal threshold couple for average passengers waiting time. However, in this model both elevators serve the whole building, intuitively when arrive flow rate is relatively large, and the target building is relatively high, it is wise to separate the elevators service floor area. In two elevators system, one elevator serves the lower part of the building and the other elevator serves the higher part of the building. We let the passengers to know this information so they can select the right elevator. The reference [3] also shows the justification of this argument. Notes that the optimality of this separated serve floors policy depends on the intensity of the arrive flow, it works well when arrive flow intensity is large and passengers' destination floors are uniform distribution. Otherwise, it is wise to decide the service floor area based on the target distribution. An extreme case is that if the arrive rate is very large, but most of the passengers are going to the lower part of the building. In this case, if we do not differentiate the passengers' significance, it is wise to let both elevators serve the lower part of the building.

For the follow MATLAB simulation, we model the system as two paralleled one elevator queues and the passengers arrive process for both two queue are Poisson processes at rate $\lambda = 1/3$ which is the same as the second two-elevators simulation. For the serve process in pure threshold model, we set the service time is a random variable drawn from an exponential distribution with a rate of $\mu = 1/10$ for both elevators. In this case, in order to represents the separated floors policy, we assume the destination floor is uniform distribution, therefore, set $\mu_1 = 1/5$ which represents the lower floors service and $\mu_2 = 1/15$ which represents the higher floors service. For both queues, we simulate 8000 passengers and do ten runs. We collect waiting time for both two queues and do the average. Capacity for both elevators are ten passengers. Fig.5 shows the performance surface and the block dot on the surface is the optimal threshold couple value which is 9.5546 unit-time and the optimal threshold couple is (2,5). The black dot above the surface is just to make a better visualization of the plot. The interesting thing is that for the Threshold plus separated serve floors policy, the performance surface does not show an obvious threshold characteristic like we did in the pure threshold policy. I think the reason is that the model is two paralleled one elevator queues which is not the model that reference paper and I proved.

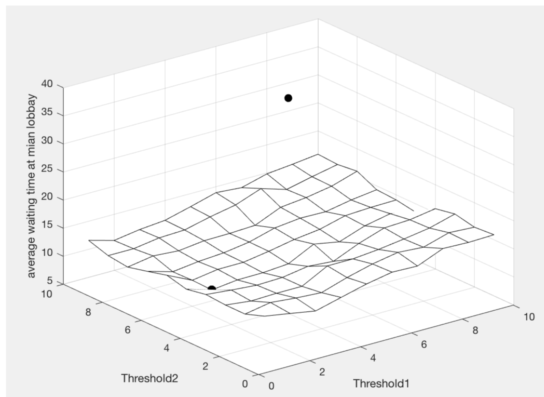

Fig.5. Threshold plus separate serve area policy with λ$= 1/3$, $\mu_1 = 1/5$ and $\mu_2 = 1/15$

6 CONCLUTION AND EXTENSIONS

In this course project report, we treated the up-peak elevator dispatching problem as a batch service queueing system. First, show that a threshold policy is the optimal policy to minimize the average waiting time in single elevator dispatch system. Second, do a MATLAB simulation for single elevator system and treat the arrive rate as a variable and show the optimality of a threshold policy compared with other constant threshold policy. Third, move forward to the two-elevator system and do a MATLAB simulation to show the existence of the threshold-based policy in multi-elevators system. Fourth, modify the service process and introduce a threshold plus separate floor areas policy to improve the performance and do two MATLAB simulations.

The drawback of this course project report is that the way to modify the model can be more delicate. For the modified service process, we can dig it relationship between the arrive process deeper. For the threshold plus separate floors policy, the further work could be focused on finding specific condition that we can get better performance by using threshold plus separated floors policy.

REFERENCES

[1] C. G. Cassandras and S. Lafortune, "*Introduction to discrete event system*" Springer, NY,2008.

[2] D. L. Pepyne and C. G. Cassandras, "Optimal dispatching control for elevator systems during uppeak traffic," IEEE Trans. Contr. Syst. Technol., vol. 5, pp. 629–643, 1997.

[3] P. Vakili, "A standard clock technique for efficient simulation," Operations Res. Lett., vol. 10, pp. 445–452, 1991.

[4] G. C. Barney and S. M. dos Santos, Elevator Traffic Analysis Design and Control, 2nd ed. London: Peter Peregrinus, 1985.