Optimizing Elevator Control with Reinforcement Learning

DRAFT

Omar Elbaghdadi

A thesis presented for the degree of
Bachelor Econometrics and Operations Research

School of Business and Economics
Vrije Universiteit Amsterdam
The Netherlands
May 10, 2018

Thesis Title

Thesis Subtitle

Author Name

Abstract

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Contents

1 Introduction

(FACTCHECK) Life is hard to imagine without elevators nowadays. According to ??, the number of tall buildings constructed keeps increasing every year. Elevators, which play a vital role as a means of transportation in multi-storied buildings, and their service quality are thus becoming increasingly important. Elevator controllers handle passenger calls while trying to optimize service quality, e.g. by minimizing passenger waiting time.

The core of classical and even state-of-the-art elevator control strategies are heuristics. Examples are [examples].

Heuristic control strategies are difficult and costly to design Not only are they suboptimal, heuristic control strategies are also inflexible, due to their inability to deal with

2 Model of the System

This section introduces the model of the elevator system studied here. The modelling approach is similar to the one used by [2]. We use this model in all further analysis. The system dynamics are parameterized as follows:

- Number of floors: 5.
- Number of elevator cars: 1. We consider only a single elevator to simplify the problem.
- Elevator floor time (time it takes for an elevator to pass a floor at full speed): ??.
- Elevator stop time (time it takes for the elevator to decelarate, open and close the doors, and accelerate again): ??.
- Elevator load time (the time it takes for a passenger to enter or exit the elevator): 1 second.
- Floor height. Set to ?? here.
- Capacity of the cars: 8 passengers.
- Maximum number of passengers waiting for an elevator: 2 passengers.

We model the elevator system as a discrete-time system. In this way, we can model the environment as a Markov Decision Process with a finite number of states and actions. Using this approach, the states and actions can be fully specified.

The size of the state space can be changed by varying the number of floors, number of elevator cars, capacity of the cars and the maximum number of passengers. The parameters have been chosen such that the state space is finite and reasonably sized. This is of great importance when using algorithms in which it is necessary to store values for each state. If the state space is too large, running time and memory capacity can become intractable. We would need to approximate the value functions.

We use several assumptions to simplify the model further. We assume that the maximum number of passengers waiting for an elevator is 2. If another passenger arrives at a floor where 2 passengers are already waiting, we treat it as if there are still only 2 passengers waiting. We do this to limit the size of the state space.

It is not possible to observe the full state of the system. After a button is pressed, the elevator does not know if another passenger arrived after the first one. One way to deal with this is assuming *omniscience*. We assume that in every state, the elevator controller knows how many passengers are waiting at a floor, when they arrived and where they are going. Although this is not a particularly realistic assumption, it simplifies analysis of the system.

2.1 Traffic and Passenger Arrival

It is important to take into account the traffic profile at a time. General building traffic profiles have been identified [1]. Four important profiles are up-peak, down-peak, inter-floor, and lunchtime. We will concern ourselves only with the down-peak traffic profile. Down-peak is a traffic pattern in which passengers are primarily moving down to the ground floor. An example is people going home at the end of a business day in an office building. We will assume every arriving passenger wants to go to the ground floor.

We model the arrival of passengers as a Poisson process with rate parameter λ being the expected number of people arriving at a floor each minute. The rate can vary across floors. We set it to ??. For every floor, once a passenger arrives, the next passenger will arrive after some amount of time. The amount of time is drawn from an exponential distribution with rate λ.

2.2 State and Action Space

Let N be the number of floors in the building, including the ground floor.

A state s is defined by several variables:

$$s = \begin{bmatrix} c, & p, & o, & d \end{bmatrix}^\top$$

where

- c is an $N-1$ length vector indicating the number of people waiting at each floor excluding the ground floor. The entries of c take values in $\{0, 1, 2\}$.
- $p \in \{0, 1, \ldots, N-1\}$ indicates at which floor the elevator is locatesd.
- $o \in \{0, 1, \ldots, \text{elevator\_capacity}\}$ indicates how many passengers are occupying the elevator.
- $d \in \{-1, 0, 1\}$ indicates the direction of the elevator. In ascending order, the values indicate a downward movement, no movement, and an upward movement respectively.

The cardinality of the state space is

$$\begin{aligned} &(\text{floor\_capacity} + 1)^{N-1} \cdot N \cdot (\text{elevator\_capacity + 1}) \cdot \text{num\_directions} \\ &= 3^4 \cdot 5 \cdot 9 \cdot 3 = 10935 \end{aligned}$$

which is a reasonably small size.

Now that we have the state space defined, we can move on to defining the actions we can take. What action the elevator is able to take will depend on the state of the system.

We define the actions as follows:

- If the elevator is moving ($d \neq 0$):
  - Stop at next floor.
  - Continue past next floor.
- If the elevator is not moving ($d = 0$):
  - Go up
  - Go down

There are additional restrictions on what actions can be taken. When at the bottom and top floors, we can not choose the action go down and go up respectively. We cannot continue past the bottom and top floors. We can not turn in a single action. If the current direction is 1, for example, we have to first stop the elevator before we can set it to -1. Taking into account passenger expectations, a car cannot turn until it has served all the calls in its present direction.

2.3 Performance Measures and Reward

We need a way to measure the performance of the method we are applying. The goal is to minimize some function of passengers' waiting time. We consider the average passenger waiting time, which is generally considered a primary objective [1]. The waiting time of a passenger is defined as the time between the passenger's arrival at the floor and the passenger's entry into a car. Other possible performance measures are system time and the fraction of passengers waiting more than T seconds, where T is typically 60. System time is defined as the waiting time combined with the passenger's travel time.

We want to define a reward such that maximizing this reward will lead to a lower average waiting time. We do this by defining the reward function

$$r(s, a, s') = -\sum_{i=0}^{N-1} c_i$$

where c_i is the ith element of c. s' is the next state observed after taking action a in state s. (MAKE NOTATION SECTION?). In other words, the reward is higher with less people in the system. (ADJUST REWARD TO TAKE INTO ACCOUNT WAITING TIME?). A reward r_t at timestep t is observed after taking an action in timestep $t - 1$.

References

[1] James Lewis. A dynamic load balancing approach to the control of multiserver polling systems with applications to elevator system dispatching. 1991.

[2] Xu Yuan, Lucian Buşoniu, and Robert Babuška. Reinforcement learning for elevator control. *IFAC Proceedings Volumes*, 41(2):2212 – 2217, 2008. 17th IFAC World Congress.