

# ECE 590 Final Project: Scheduling with N Elevators

Hansung Kang, Kevin Kuo

May 3, 2018

## 1 Extension to N Elevators

### 1.1 Construction

We start by defining the sets  $X_1 = \mathbb{Z}_N$  and  $X_2 = \{\uparrow, \downarrow\}$ . Letting  $X = (X_1 \times X_2) - \{(N, \uparrow), (1, \downarrow)\}$ , we can represent both the elevator position and the position of a person requesting an elevator:

For instance,

- If an elevator is at  $(1, \uparrow) \in X$ , then it is currently on the  $2^{nd}$  floor ( $1^{st}$  floor is at  $0 \in X_1$ ) and moving upwards.
- If a person requesting an elevator is at  $(4, \downarrow) \in X$ , then she is on the  $5^{th}$  floor and wants to move downwards.

To make  $X$  easier to visualize, we wish to arrange its elements in a circle like this:

By setting the convention that elevators can only move clockwise in this circle, we can more easily visualize their paths. For instance, if an elevator wanted

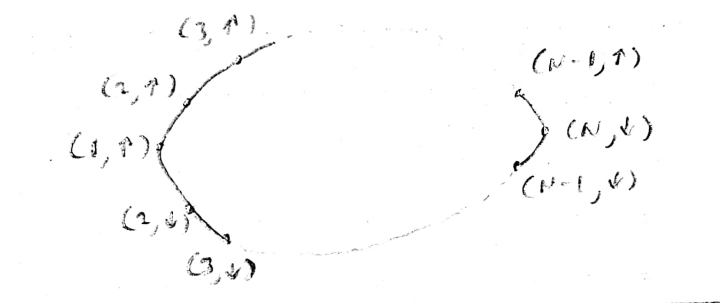


Figure 1: Representing  $X$  as a circle

to move from the  $3^{rd}$  floor heading up to the  $2^{nd}$  floor heading down, it could travel clockwise from  $(2, \uparrow)$  to  $(N, \downarrow)$  to  $(1, \downarrow)$ . (Realistically, if the elevator did not have to stop anywhere else, it could just move straight down – i.e. skip from  $(2, \uparrow)$  to  $(2, \downarrow)$ , and then go to  $(1, \downarrow)$ . But, since the following analysis just considers the worst-case timing for the elevator (i.e. stops on every floor in between), we do not need to consider this shorter path.)

To create this circle, we represent  $X$  as  $\mathbb{Z}_{2N-2}$  by defining the map  $f : X \rightarrow \mathbb{Z}_{2N-2}$  as

$$f(x) = \begin{cases} \pi_1(x) & (\text{if } \pi_2(x) = \uparrow) \\ (2N-2) - \pi_1(x) & (\text{if } \pi_2(x) = \downarrow) \end{cases}$$

where  $\pi_1(x)$  is the projection of  $x$  onto  $X_1$ , and  $\pi_2(x)$  is the projection of  $x$  onto  $X_2$ .

Thus, using Figure 1 to picture  $f(X)$ , the top half the circle represents the values 0 to  $N-1$  in  $\mathbb{Z}_{2N-2}$ , and the bottom half represents the values  $N$  to  $2N-1$ .

## 1.2 Worst-Case Time Analysis

Define the map  $t : X \times X \rightarrow \mathbb{Z}_N$  to be  $t(x_1, x_2) = f(x_2) - f(x_1) \pmod{N}$ .

**Proposition 1.** *Suppose an elevator  $e \in X$  had to travel from  $x_1$  to  $x_2$  ( $x_1, x_2 \in X$ ), and the act of moving one floor and stopping took one time unit. Then  $t(x_1, x_2)$  is the worst-case time it would take to make that travel.*

*Proof.*  $e$  must travel clockwise in Figure 1. Thus, letting  $T$  be the time  $e$  would take,  $T \leq f(x_2) - f(x_1) \pmod{N} = t(x_1, x_2)$ . This upper bound is achievable if  $e$  has to stop at every floor along the path from  $f(x_1)$  to  $f(x_2)$ .  $\square$

If a person waits at some point  $x \in X$  with destination  $d \in X$ , and elevator  $e \in X$  had to come to pick them up, then, by Proposition 1, the worst-case time to travel from  $e$  to  $x$  is  $t(e, x)$ , and the worst-case time to travel from  $x$  to  $d$  is  $t(x, d)$ . Thus, the worst-case time  $T$  it would take for elevator  $e \in X$  to pick up the person and take them to their destination would be

$$T = t(e, x) + t(x, d)$$

Let  $\{e_1, e_2, \dots, e_E\} \in X$  be a set of elevators. Then let  $T_{e_i}$  be the worst-case time defined above for elevator  $e_i$ . Thus

$$T_{e_i} = t(e_i, x) + t(x, d) \tag{1}$$

**Proposition 2.** *For some  $i \in [1, E]$ ,  $\left(T_{e_i} = \min_{j \in [1, E]} T_{e_j}\right) \Leftrightarrow \left(t(e_i, x) = \min_{j \in [1, E]} t(e_j, x)\right)$*

*Proof.* If  $T_{e_i} = \min_{j \in [1, E]} T_{e_j}$ , then, by (1),  $t(e_i, x) = T_{e_i} - t(x, d) = \left( \min_{j \in [1, E]} T_{e_j} \right) - t(x, d) = \min_{j \in [1, E]} (T_{e_j} - t(x, d)) = \min_{j \in [1, E]} t(e_j, x)$ .

If  $t(e_i, x) = \min_{j \in [1, E]} t(e_j, x)$ , then  $T_{e_i} = t(e_i, x) + t(x, d) = \left( \min_{j \in [1, E]} t(e_j, x) \right) + t(x, d) = \min_{j \in [1, E]} (t(e_j, x) + t(x, d)) = \min_{j \in [1, E]} T_{e_j}$   $\square$

To minimize the worst-case time the person at  $x$  needs to wait to get to her destination, we pick the elevator  $e_i$  such that  $T_{e_i} = \min_{j \in [1, E]} T_{e_j}$ . By Proposition 2, this means we choose the elevator  $e_i$  such that  $t(e_i, x) = \min_{j \in [1, E]} t(e_j, x)$ . Thus, by the definition of  $t$ , we choose the elevator  $e_i$  such that  $f(x) - f(e_i) \pmod{N}$  is minimized. Note that this expression does not depend on the destination floor the person wants to go to. Thus, in a rough sense, just choosing the "closest" elevator to the person requesting one is the optimal approach to minimizing the worst-case time.

### 1.3 Algorithm

The following algorithm summarizes the approach we derived:

Let  $S$  be a system of  $E$  elevators  $\{e_1, e_2, \dots, e_E\} \in X$ . An input to  $S$  would be a person represented as an element in  $X$ .

---

#### Algorithm 1 Minimize worst-case travel time

---

```

1: procedure MINIMIZEWORSTCASETIME
2:   for input  $x \in X$  into  $S$  do
3:     minTime  $\leftarrow \infty$ 
4:     minElevator  $\leftarrow \emptyset$ 
5:     for each elevator  $e \in \{e_1, e_2, \dots, e_E\} \in X$  do
6:       criterion  $\leftarrow f(x) - f(e)$ 
7:       if criterion  $<$  minTime then
8:         minTime  $\leftarrow$  criterion
9:         minElevator  $\leftarrow e$ 
10:    add  $x$  to the set of minElevator's inputs

```

---

### 1.4 Pitfalls

In this algorithm, we do not consider the following factors:

- Elevator capacity
- What happens when an elevator fails

- Time it takes an elevator to collect people at a floor
- The actual time it takes for an elevator to travel between floors.

We can address the first two bullet points: Step 5 of Algorithm 1 could be modified to not consider elevators that are either full or dysfunctional. There are two other pitfalls to this approach:

Firstly, we only considered the worst-case time. Thus, we assumed that each elevator would stop at each floor when traversing the building. We did not do an average-time analysis, which would not make that assumption.

Secondly, we did not consider an elevator's existing stopping points. This could lead to a scenario where one elevator could be assigned a lot of people and others would not; the elevators could become unbalanced in the number of people they have to accommodate. In Algorithm 1, this can be accounted for by improving the criterion set in Step 6. For instance, the criterion could be set to  $f(x) - f(e) + P_e(n)$ , where  $P_e(n)$  is some weighting factor dependent on the number of people  $n$  currently in elevator  $e$ . If  $P_e(n)$  increased monotonically with respect to  $n$ , then elevators that are more empty would become more likely to be chosen for the person at  $x$ .