

Independent Elevator Simulation Project

Wanxiang Xie

April 2020

Contents

1	Problem description	3
1.1	Research about feasible parameters	3
1.2	Parameter	4
1.3	Result assumption	4
2	Design	5
2.1	Function	6
3	Visualization	7
3.1	Choose Mode	7
3.2	Normal Mode	8
3.3	HC5 Mode	9
4	Output File	11
5	Conclusion	14
5.1	Result	14
5.2	Analysis	15

1 Problem description

1.1 Research about feasible parameters

The elevator system uses a medium-sized commercial office building as a usage scenario to simulate running on a floor of 30F, each floor is 4 meters high, so the total height is 120 meters. This building is equipped with 3 elevators for the office building, each with a maximum capacity of 10 people. According to the standard, the maximum speed of the elevator is no less than the total height of the building / 60 = $120/60 = 2\text{m/s}$. So, we use an elevator with $V_{\text{max}} = 2\text{m/s}$ to simulate.

Assuming that when the elevator is within the weight limit, as shown in the following figure, it starts and stops at a constant acceleration.

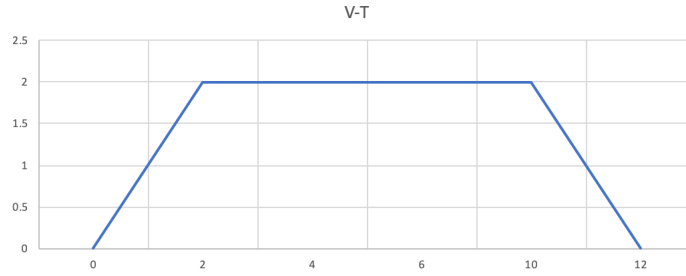


Figure 1: v - t graph

According to the GB/T10058-2009 3.3.3 standard, when the rated speed of the elevator is within $1.0\text{m/s} \leq V \leq 2.0\text{m/s}$, the acceleration and deceleration speed is not less than 0.5m/s^2 , therefore we use the acceleration $a = 1\text{m/s}^2$.

Refer to the British "CIBSE GUIDE" standard, here are several important criteria for measuring users experience:

AI (INT): average running interval time (the same group of elevators, how long will an elevator arrive on average)

AWT: average waiting time (the time from when the user enters the elevator hall to when the elevator arrives). According to CIBSE Standard, AWT is generally in the range of 30-50s.

HC5: Five Minute Handling Capacity (the ratio of the number of people that the elevator can carry to the total service number in 5 minutes during the peak period) is acceptable within 10

AJT: average travel time (the time after the user enters the elevator until the elevator reaches the destination and opens the door)

ATTD: the time from when the user enters the elevator hall until it reaches the destination floor completely

Note: All the above concepts refer to the peak period (morning or evening)

Usually the most important indicators to measure a building are the reference HC5 and INT, sometime need to measure AWT at the same time. So My programming are mianly measure these three parameters.

1.2 Parameter

building's total height:120m

numbers of floor: 30F

each floor height: 4m

numbers of elevator: 3

elevator max speech: 2m/s

elevator launch and stop acceleration: 1m/s

each elevator capacity: 10 people

1.3 Result assumption

First of all, let us make a rough calculation of the above parameters according to my design:

AWT: Assuming that passengers are evenly distributed on each floor, with the middle floor 15F as the standard, the AWT of a single elevator = $(4 + 116/2 + 10) / 2 = 36s$

AI(INT): It's hard to assume it, because I can't predict passengers choice before I implement.

HC5: According to the total number of 500 people in the office building, during the early peak period, it is assumed that the opening and closing time of the elevator is 10s each time, and the round-trip time of an elevator is $(4 + 116/2 + 10) * 2 = (4s + 58s + 10s) * 2 = 72s * 2 = 144s$, then the three elevators run at full load for 5 minutes at the same time, and the number of people that can be loaded is $60s * 5 / 144s * 30 = 62.5$ people Thus, the lift capacity of this elevator system in 5 minutes $HC5 = 67.1 / 500 * 100$

Note: More accurate and detailed indicator calculations will be given based on the data later and I will compare my assumption and result in the

end.

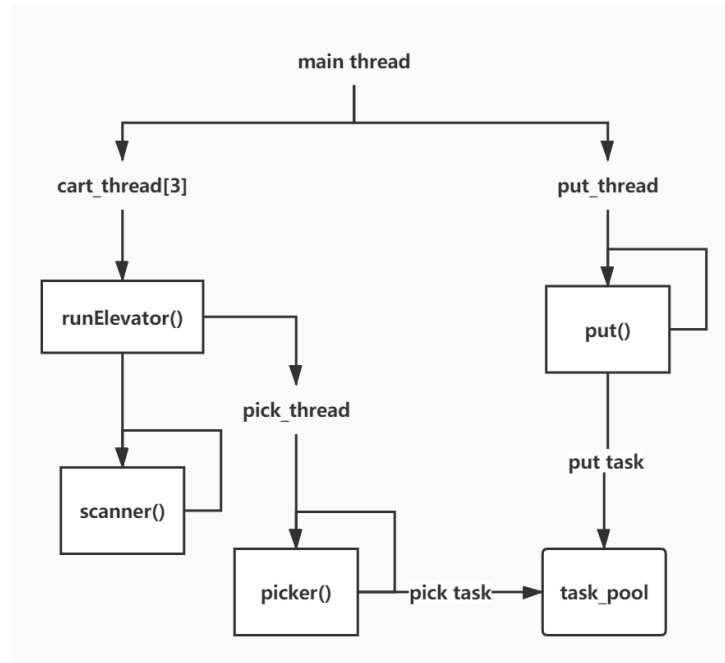
2 Design

There are 3 questions:

- 1.How many threads will be used?
- 2.How to put task and pick task(the task just like people press button outside the elevator)?
- 3.Which algorithm to use to do task?
- 4.How to set target floor(target floor just like people enter in cart and press button)?

In the main thread, I create 3 threads `cart_thread[3]`, each thread control a single cart to run and pick tasks.And I also create a `put_thread` to put task into `task_pool`.

In each `cart_thread`, beside running to move up and down(using `scanner()`), I create a `pick_thread` to pick task from `task_pool`.



2.1 Function

`runElevator()`: two main functions, one is to create `pick_thread`; Other is let cart do `scanner()` in while loop.

`scanner()`: the cart will scan from current floor up or down, the first find clicked button will set as move direction and do `goUp_wayDown()` then do `goDown_wayDown()` or vice versa.

`put()`: this function will create random tasks and put to `task_pool`. (use `hash_set`)

`picker()`: this function has two pick ways.

1. when cart isn't running, cart will directly use Proximity algorithm to do task.

2. when cart is running, cart will try to find tasks in the same direction and use Waydown algorithm to do task.

Obviously, We can find answers for Q1,Q2,Q3 above. But how about Q4?

In my design, I have a `setFloor()` function to set Current Floor whenever cart arrive a task floor(Like passenger has been taken). And then this function will immediately set new target.

You may ask, how to set new target?

Well, I have two modes in my program. If you choose Normal mode, only the same direction's button will be set. If you choose HC5 Test mode, the cart will always set up direction because in this mode, we need to simulate Early peak time, people just press button at basement(0F) or ground(1F) to go up, and then press higher target floor in cart, right?

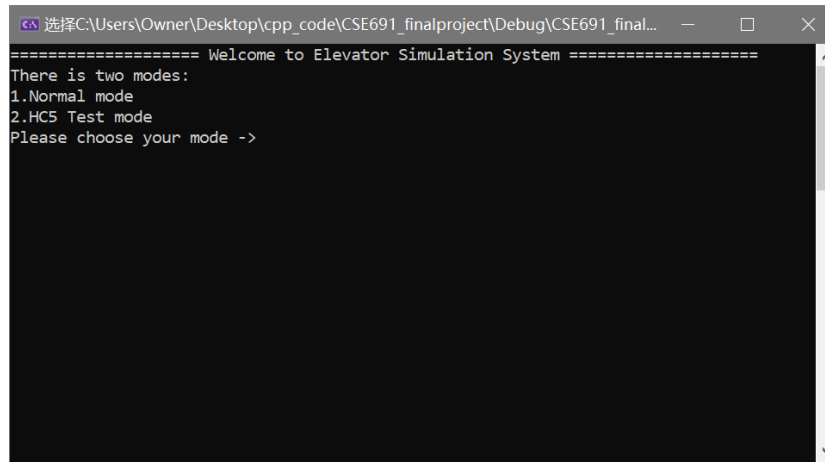
3 Visualization

3.1 Choose Mode

When program is running, you should choose one mode to run.

1.Nomral: program will never end, it will show you elevators running processing and buttons panel status.

2.HC5 Test: program will stop at 500 seconds, the timing is started when the scheduler assigns tasks(not running time).



```
选择C:\Users\Owner\Desktop\cpp_code\CSE691_finalproject\Debug\CSE691_final...
===== Welcome to Elevator Simulation System =====
There is two modes:
1.Normal mode
2.HC5 Test mode
Please choose your mode ->
```

Figure 2: Choose Mode View

3.2 Normal Mode

There are three section:building simulation area, cart button panel simulation area and parameter area.

1.Building simulation: to simulate 3 carts go up and down, when it arrived clicked floor, it will open and close door(wait 10s).

2.Cart button panel simulation: to simulate each cart inner button panel, when button select, "A" will be shown.

3.Parameter area: to show various parameter, like Run time, Responed Task Count,Arrived Target Count,AWT(average waiting time),ALINT(average running interval time).


```

C:\Users\Owner\Desktop\cpp_code\CSE691_finalproject\Debug\CSE691_finalproject.exe
===== Building =====
Cart No. 1 2 3
30F *
29F *
28F
27F
26F
25F
24F
23F
22F
21F
20F *
19F
18F
17F
16F
15F
14F
13F
12F
11F
10F
9F
8F
7F
6F
5F
4F
3F
2F
1F
0F
Current F:30 29 20
===== Button Panel =====
[*]: current [A]: selected []: unselected
No. 0F 1F 2F 3F 4F 5F 6F 7F 8F 9F 10F 11F 12F 13F 14F 15F 16F 17F 18F 19F 20F 21F 22F 23F 24F 25F 26F 27F 28F 29F 30F
1
2
3
cart button panel simulation
Mode: Normal
Run time: 151.184s
Responded Task Count: 6 Arrived Target Count: 4
AWT(average waiting time): 42.026s
AI_INT(average running interval time): 36.3124s
Programming is running, You can stop it at any time and check log.txt file.
parameter area

```

Figure 3: Normal mode

3.3 HC5 Mode

There are three section:building simulation area, cart button panel simulation area and parameter area. What's different with Normal mode is Parameter area.

3.Parameter area: Only HC5 Test mode will show HC5(five minute handling capacity) parameter.

```
C:\Users\Owner\Desktop\cpp_code\CSE691_finalproject\Debug\CSE691_finalproject.exe
===== Building =====
Cart No. 1 2 3
30F
29F
28F
27F
26F
25F
24F
23F
22F
21F
20F
19F
18F
17F
16F
15F
14F
13F
12F
11F
10F
9F
8F
7F
6F
5F
4F
3F
2F
1F
0F
Current F: 1 11 10
===== Button Panel =====
[*]: current [A]: selected []: unselected
-----
No. 0F 1F 2F 3F 4F 5F 6F 7F 8F 9F 10F 11F 12F 13F 14F 15F 16F 17F 18F 19F 20F 21F 22F 23F 24F 25F 26F 27F 28F 29F 30F
1 A *
2
3
-----
Mode: HC5 Test
Run time: 329.131s
Responded Task Count: 26 Arrived Target Count: 17
AWT(average waiting time): 85.1535s
AI_INT(average running interval time): 29.5382s
HC5(five minute handling capacity): 12%
Programming is running, Please wait for completion...
-
```

Figure 4: HC5 Test mode - running

When HC5 is calculated finished, Console will show prompt below. You can see final HC5 value.

```
C:\Users\Owner\Desktop\cpp_code\CSE691_finalproject\Debug\CSE691_finalproject.exe
===== Building =====
Cart No. 1 2 3
30F
29F
28F
27F
26F
25F
24F
23F
22F
21F
20F
19F
18F
17F
16F
15F
14F
13F
12F
11F
10F
9F
8F
7F
6F
5F
4F
3F
2F
1F
0F
Current F:30 1 14
===== Button Panel =====
[*]: current [A]: selected []: unselected
No. 0F 1F 2F 3F 4F 5F 6F 7F 8F 9F 10F 11F 12F 13F 14F 15F 16F 17F 18F 19F 20F 21F 22F 23F 24F 25F 26F 27F 28F 29F 30F
1
2
3
=====
Mode: MCS Test
Run time: 590.782s
Responded Task Count: 38 Arrived Target Count: 23
AWT(average waiting time): 101.961s
AI_INT(average running interval time): 34.4538s
MCS(five minute handling capacity): 15%
MCS time is up, You can stop the programming and check test_HC_log.txt file.
```

Figure 5: HC5 Test mode - finished

4 Output File

8. There are three output file.

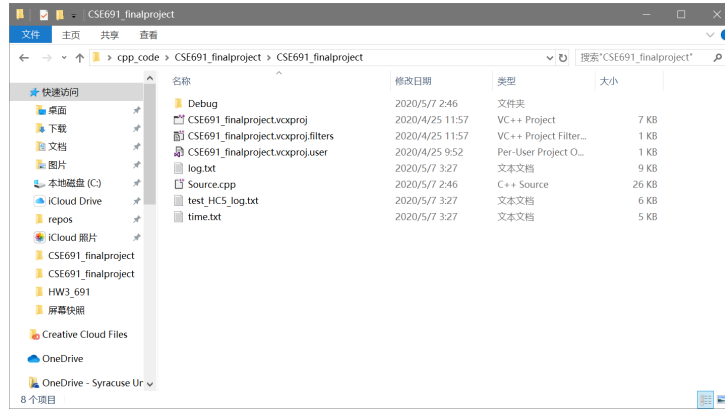


Figure 6: HC5 Test mode - finished

log.txt is mainly to record every important operation.(most important)

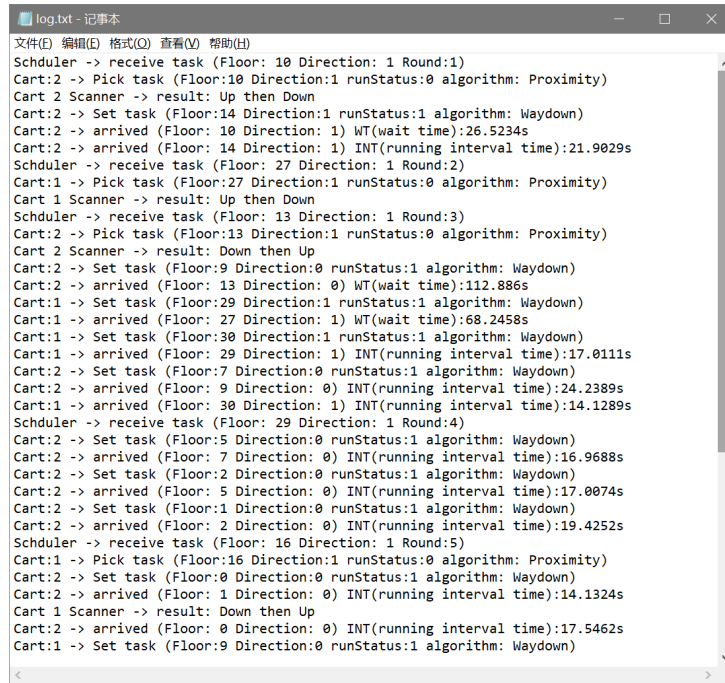


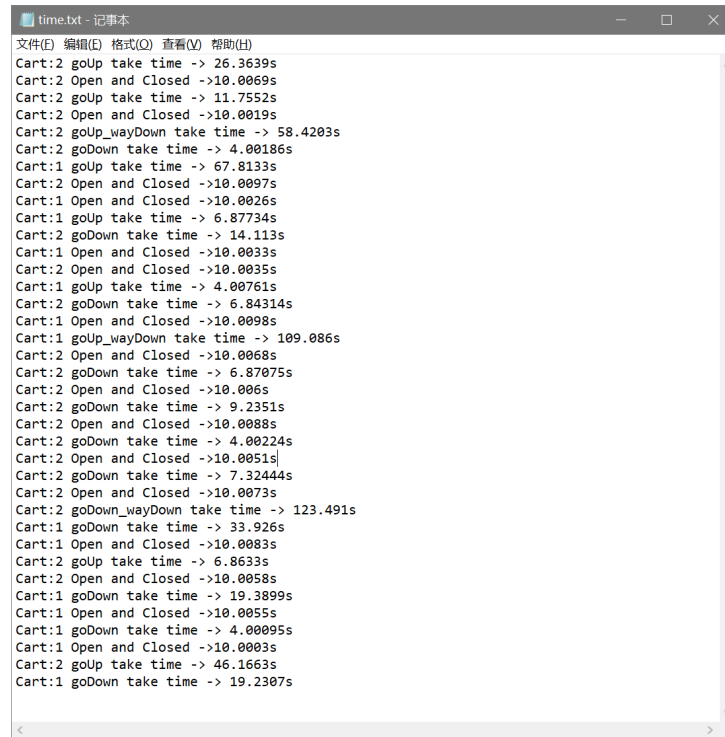
Figure 7: HC5 Test mode - finished

test_HC5_log.txt only record operatorion about HC5.

```
test_HC5_log.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
Scheduler -> receive task (Floor: 1 Direction: 1 Round:32)
Scheduler -> receive task (Floor: 0 Direction: 1 Round:32)
Scheduler -> receive task (Floor: 0 Direction: 1 Round:33)
Scheduler -> receive task (Floor: 1 Direction: 1 Round:33)
Scheduler -> receive task (Floor: 0 Direction: 1 Round:34)
Scheduler -> receive task (Floor: 1 Direction: 1 Round:34)
Scheduler -> receive task (Floor: 1 Direction: 1 Round:35)
Scheduler -> receive task (Floor: 0 Direction: 1 Round:35)
Scheduler -> receive task (Floor: 0 Direction: 1 Round:36)
Scheduler -> receive task (Floor: 1 Direction: 1 Round:36)
Scheduler -> receive task (Floor: 0 Direction: 1 Round:37)
Scheduler -> receive task (Floor: 0 Direction: 1 Round:38)
Scheduler -> receive task (Floor: 0 Direction: 1 Round:39)
Scheduler -> receive task (Floor: 0 Direction: 1 Round:40)
Scheduler -> receive task (Floor: 1 Direction: 1 Round:41)
Scheduler -> receive task (Floor: 0 Direction: 1 Round:41)
Scheduler -> receive task (Floor: 0 Direction: 1 Round:42)
Scheduler -> receive task (Floor: 1 Direction: 1 Round:42)
Scheduler -> receive task (Floor: 1 Direction: 1 Round:43)
Scheduler -> receive task (Floor: 0 Direction: 1 Round:43)
Scheduler -> receive task (Floor: 0 Direction: 1 Round:44)
Scheduler -> receive task (Floor: 1 Direction: 1 Round:44)
Scheduler -> receive task (Floor: 0 Direction: 1 Round:45)
Scheduler -> receive task (Floor: 1 Direction: 1 Round:45)
Scheduler -> receive task (Floor: 1 Direction: 1 Round:46)
Scheduler -> receive task (Floor: 0 Direction: 1 Round:46)
Scheduler -> receive task (Floor: 0 Direction: 1 Round:47)
Scheduler -> receive task (Floor: 1 Direction: 1 Round:47)
Scheduler -> receive task (Floor: 1 Direction: 1 Round:48)
Scheduler -> receive task (Floor: 0 Direction: 1 Round:48)
Scheduler -> receive task (Floor: 1 Direction: 1 Round:49)
Scheduler -> receive task (Floor: 0 Direction: 1 Round:49)
Scheduler -> receive task (Floor: 0 Direction: 1 Round:50)
Scheduler -> receive task (Floor: 1 Direction: 1 Round:50)
Time is up
Total number is 500, HC5 mode pickup count is 60
HC5 = 12%
```

Figure 8: HC5 Test mode - finished

time.txt is to record elevator operator on each timepoint.



```
time.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
Cart:2 goUp take time -> 26.3639s
Cart:2 Open and Closed ->10.0069s
Cart:2 goUp take time -> 11.7552s
Cart:2 Open and Closed ->10.0019s
Cart:2 goUp_wayDown take time -> 58.4203s
Cart:2 goDown take time -> 4.00186s
Cart:1 goUp take time -> 67.8133s
Cart:2 Open and Closed ->10.0097s
Cart:1 Open and Closed ->10.0026s
Cart:1 goUp take time -> 6.87734s
Cart:2 goDown take time -> 14.113s
Cart:1 Open and Closed ->10.0033s
Cart:2 Open and Closed ->10.0035s
Cart:1 goUp take time -> 4.00761s
Cart:2 goDown take time -> 6.84314s
Cart:1 Open and Closed ->10.0098s
Cart:1 goUp_wayDown take time -> 109.086s
Cart:2 Open and Closed ->10.0068s
Cart:2 goDown take time -> 6.87075s
Cart:2 Open and Closed ->10.006s
Cart:2 goDown take time -> 9.2351s
Cart:2 Open and Closed ->10.0088s
Cart:2 goDown take time -> 4.00224s
Cart:2 Open and Closed ->10.0051s
Cart:2 goDown take time -> 7.32444s
Cart:2 Open and Closed ->10.0073s
Cart:2 goDown_wayDown take time -> 123.491s
Cart:1 goDown take time -> 33.926s
Cart:1 Open and Closed ->10.0083s
Cart:2 goUp take time -> 6.8633s
Cart:2 Open and Closed ->10.0058s
Cart:1 goDown take time -> 19.3899s
Cart:1 Open and Closed ->10.0055s
Cart:1 goDown take time -> 4.00095s
Cart:1 Open and Closed ->10.0003s
Cart:2 goUp take time -> 46.1663s
Cart:1 goDown take time -> 19.2307s
```

Figure 9: HC5 Test mode - finished

You should delete all of them before you restart the program.

5 Conclusion

5.1 Result

Remember my previous assumption?

AWT:

Assuming: 36s

Result: 28-50s in first 20 tasks, 50-86 in further tasks in HC5 mode. 34-130S in Normal mode

AI(INT):

Assuming: nil

Result: 27-31s in Normal mode; 35-48s in HC5 mode

HC5:

Assuming: 12.5%
Result: 12-18%

5.2 Analysis

As we can see in result, the AI(INT) and HC5 is acceptable, But AWT is pretty much higher than we assume.

Why AWT is so higher, especially as time progresses?

I think it is because when the task is continuously added, the next task needs to wait for the previous task to be completed before it gets the desired and there is a 10 second opening and closing overhead each time.