Elevator Scheduling Optimization

Tzu-En Huang
Dept. of EECS, NTHU
Hsinchu, Taiwan

You-Ting Liu
Dept. of EECS, NTHU
Hsinchu, Taiwan

Jyun-Siang Huang
Dept. of EECS, NTHU
Hsinchu, Taiwan

Chih-Yun Lin
Dept. of EECS, NTHU
Hsinchu, Taiwan

Shan-Jou Huang
Dept. of EECS, NTHU
Hsinchu, Taiwan

Yun Huang
Dept. of EECS, NTHU
Hsinchu, Taiwan

***Abstract*—Though elevators are among the most energy-consuming facilities in buildings, traditional elevator dispatch algorithms usually take time efficiency as the first priority while neglecting energy consumption. In this project, we are dedicated to designing a new algorithm which prioritizes energy efficiency instead, using the knowledge we have learned. A previous study has shown that the GA-based method can significantly reduce both time and energy consumption compared to FCFS, demonstrating the advantages of algorithms incorporating randomness in the elevator scheduling problem. Our study formulates the elevator dispatch problem as a partially observable problem — to get as close to a real-world situation as possible, we assume that the agent doesn't know which floor the passenger wants to go to until they enter the elevator. Besides, the effects of different movement states, the number of passengers on energy cost are considered. We plan to solve the problem with the hill climbing algorithm, the elevator selects movements based on a cost function combining energy use, action time, and passenger queuing delays, weighted by priority. The system only considers floors with active requests, reflecting real-world constraints. To avoid local minima, equal-cost moves and random restarts are incorporated. The process continues until all internal and external demands are satisfied. Our goal is to develop an energy-optimized agent capable of significantly reducing electricity usage and passenger waiting times, offering a promising solution for sustainable living environments.**

***Index Terms*—elevator scheduling, hill-climbing algorithm, multi-objective optimization, partially observable**

I. Motivation, Goal, and Challenges

A. Motivation

We have observed that the SCAN algorithm often demonstrates excellent time performance in disk scheduling. However, while SCAN focuses primarily on minimizing scanning time, real-world elevator dispatching is considerably more complex and dynamic. In fact, elevators are among the top three energy-consuming facilities in residential and commercial buildings, second only to air conditioning and lighting. Most existing energy-saving measures for elevators focus on hardware improvements, such as the adoption of variable frequency technology to reduce power consumption.

B. Goal and Challenges

Considering that acceleration and deceleration require significantly more energy than constant-speed travel, our goal is to optimize elevator route planning to minimize unnecessary stops and displacements. For example, situations where no passengers can board or exit the elevator because it is already full. By addressing these inefficiencies, we ultimately seek to achieve greater energy savings.

When addressing elevator scheduling for power optimization, several major challenges must be considered. First, the problem is inherently partially observable: future passenger requests cannot be fully predicted, requiring the system to make decisions based on incomplete information. Furthermore, the search space at each decision point is extremely large, making exhaustive search impractical and necessitating efficient search reduction strategies. Another critical constraint is the need for real-time computation under limited hardware resources. The scheduling algorithm must deliver results within a short time frame while minimizing computational overhead to avoid defeating the purpose of energy conservation.

In addition, there is a trade-off between transportation efficiency and energy consumption. While minimizing waiting and travel times is important, it often leads to frequent acceleration and deceleration, which can increase energy usage. Therefore, a key challenge lies in balancing service quality with energy efficiency. Finally, the scheduling strategy must be scalable in various building sizes and remain simple enough to be implemented and maintained within the practical limitations of typical elevator control systems.

These challenges highlight the need for a lightweight, adaptive scheduling approach tailored specifically to the constraints of elevator operations.

II. Related Work

Elevator scheduling is a classic optimization problem, where the primary goals typically include minimizing passengers "waiting time", total travel time of elevator, and the energy consumption.

Zhu(2024) proposed a method of optimizing elevator scheduling through the development of mathematical models and the application of computer simulation. [1] The study focuses on a 33-floor building with four elevators, and each elevator was modeled using parameters such as passenger capacity, speed, and door operation time. In order to balance user satisfaction and system performance, its optimization goal was to minimize the weighted sum of passenger waiting time and elevator running time.

The study first implemented three traditional scheduling strategies, including First-Come-First-Serve(FCFS), Shortest Seek Time First (SSTF), and a Zoning strategy that assigned each elevator to a fixed range of floors. In addition, in order to enhance system performance, it also used the Genetic Algorithm to solve the problem. The simulation experiment result demonstrated that the GA-based method achieved a 38% reduction in total waiting time and a 20% reduction in the energy consumption compared to FCFS.

This study shows that algorithms with randomness (such as genetic algorithms) can perform better in problems that have a large solution space, where it is difficult to find the optimal solution through exhaustive search. Inspired by this observation, we plan to use local search methods like hill climbing algorithms, which incorporate randomness and can quickly find better solutions in complex spaces by gradually improving the solution. And our goal is to improve elevator scheduling performance particularly in terms of dynamic adaptability and energy saving.

III. Problem Formulation

Considering that the total energy cost varies depending on different movement states, the number of passengers, and other factors, the problem is defined by the following elements:

A. Building Configuration

- Number of floors
- Floor height
- Number of elevators (default = 1)

B. Motor Configuration

- Energy requirement from the Nth floor to the Mth floor (precomputed and stored in a table)

C. Elevator State

- **Current floor**: the current location of the elevator
- **In-pending**: number of passengers inside the elevator going to each floor
- **Out-pending**: number of passengers waiting on each floor
- **Scheduling window size**: default as the number of requested floors

D. State Representation

- **Complete state**: visiting all requested floors (requests from both inside and outside the elevator) in a certain order.
- **Initial State**: A random complete state is generated by visiting all requested floors (from passengers both inside and outside) in a random order.

E. Successor Function

From the current floor, the elevator determines the visiting order of each requested floor. Upon arrival at a floor:

- Passengers whose destination matches the current floor exit.
- Passengers waiting on the floor board the elevator, if the capacity allows.
- The visiting order is rescheduled using the Hill-Climbing algorithm.

The neighbor definition is based on the permutation of the floors in the request list.

F. Goal Test

All passenger requests have been satisfied.

G. Path Cost

The cost is evaluated with the following priorities:

1) **Work done by the elevator**, based on the equation:

$$W = m \times a \times S$$

where m is the total mass (elevator + passengers), assuming each passenger has equal weight.

2) **Travel time**, calculated by:

$$S = V \times t + \frac{1}{2}at^2$$

where V is velocity, a is acceleration, S is distance, and t is time. Dwell time and passenger decision time are also considered.

3) **Queuing time**:
   - For passengers outside:

$$queuingtime = pickuptime - requesttime$$

   - For passengers inside:

$$queuingtime = arrivaltime - requesttime$$

H. Constraints

The elevator has a limited capacity and cannot exceed its maximum passenger load. Besides, the agent won't know which floor the passengers waiting outside the elevator are going to until they enter the elevator.

IV. Plan Method

We plan to use the hill climbing algorithm to find an energy-efficient route plan for the elevator(s).

A. Successor Function

The successor function is defined in Section III-E.

B. Cost Function

The cost function to be minimized is defined as:

$$c(n) = \alpha W + \beta T + \gamma Q \tag{1}$$

where α, β, and γ are weighting factors satisfying $\alpha > \beta > \gamma$.

1) Energy Consumption W: The energy consumption W for the next action is:

$$W = \sum (\text{Energy for each scenario}) \tag{2}$$

The energy calculations for different movement scenarios are:

- Accelerating upward or decelerating downward:
$$W = m(g + a)S \tag{3}$$
- Accelerating downward or decelerating upward:
$$W = m(g - a)S \tag{4}$$
- Moving at constant velocity (upward or downward):
$$W = mgS \tag{5}$$
- Stop (including door open/close operations): a small constant energy value.

where m is the total mass (elevator + passengers), g is the gravitational acceleration, a is the elevator's acceleration/deceleration rate, and S is the travel distance.

2) Time Consumption T: The time required for the next action is:

$$T = T_{\text{move}} + T_{\text{stop}} \tag{6}$$

where:

- T_{move}: Time to move between floors.
- T_{stop}: Sum of door opening time, door closing time, and passenger exchange time.

3) Queuing Time Q: The queuing time is composed of external and internal waiting times:

$$Q = Q_{\text{out}} + Q_{\text{in}} \tag{7}$$

where:

- Q_{out}: Waiting time for people outside the elevator.
- Q_{in}: Delay for people inside the elevator to reach their destination floor.

C. Goal State

All internal and external requests have been served.

D. Algorithm

The hill climbing algorithm we propose is defined as follows:

Algorithm 1 Hill Climbing

```
current_state ← initial state
equal_cost_steps ← 0
while not goal_reached(current_state) do
    successors ← generate all one-move successors
    best_successor ← successor with lowest c(successor)
    if c(best_successor) < c(current_state) then
        current_state ← best_successor
        equal_cost_steps ← 0
    else if c(best_successor) = c(current_state) and equal_cost_steps < MAX_EQUAL_STEPS then
        current_state ← best_successor
        equal_cost_steps ← equal_cost_steps + 1
    else
        current_state ← random restart
        equal_cost_steps ← 0
    end if
end while
return current_state
```

V. Expected Results

To evaluate the effectiveness of our design, we decide to generate several test cases and do simple simulation. Comparing the performance of our algorithm against the SCAN algorithm, we can assess whether our approach can indeed save more energy compared to this widely used method and thus analyze if our agent could be improved. Furthermore, If time permits, we plan to come up with some more complex agents which are capable of solving the multi-elevator dispatching problem, or even the elevator dispatching problem with a time limit to better reflect the real-world scenarios.

We expect that our algorithm will make a considerable reduction in electricity usage compared to the traditional SCAN algorithm, especially in the scenarios which exist uneven or dynamic request distributions. As for the case of multi-elevator, we believe the method with proper communication can achieve further energy efficiency and better passenger waiting time.

References

[1] R. Zhu, "Research on Optimizing Elevator Operation Management Strategy Based on Mathematical Model Algorithm," SHS Web of Conferences, vol. 200, Art. no. 02010, 2024.