

Submodular Function Maximization for Group Elevator Scheduling

Srikumar Ramalingam

School of Computing,
University of Utah, USA
srikumar@cs.utah.com

Arvind U. Raghunathan, Daniel Nikovski

Mitsubishi Electric Research Laboratories (MERL),
Cambridge, MA, USA
{raghunathan, nikovski}@merl.com

Abstract

We propose a novel approach for group elevator scheduling by formulating it as the maximization of submodular function under a matroid constraint. In particular, we propose to model the total waiting time of passengers using a quadratic Boolean function. The unary and pairwise terms in the function denote the waiting time for single and pairwise allocation of passengers to elevators, respectively. We show that this objective function is submodular. The matroid constraints ensure that every passenger is allocated to exactly one elevator. We use a greedy algorithm to maximize the submodular objective function, and derive provable guarantees on the optimality of the solution. We tested our algorithm using Elevate 8, a commercial-grade elevator simulator that allows simulation with a wide range of elevator settings. We achieve significant improvement over the existing algorithms.

Introduction

Group elevator scheduling refers to the problem of assigning passenger requests to specific elevators. We exemplify this using a specific example below. Consider a building with $F = 8$ floors and $C = 3$ elevator cars as shown in Figure 1. Passengers requesting elevator service press the call buttons on their respective floors to signal the direction of their rides. We refer to these as *hall calls*. For example, there are two hall calls on the floors 2 and 7 requesting service to higher and lower floors respectively. From the standpoint of the elevators, the direction of travel of the hall calls are known. However, the number of passengers behind a hall call and their destinations are typically not available in most elevator systems. The waiting passengers are unaware of the locations of individual cars, their moving directions, or the cars assigned to service them. For example in Figure 1, the first elevator car $C1$ is at the floor 3 and moving upward to service a passenger in the car going to floor 5. This information is only available after the passenger enters the car and presses the particular floor button in the elevator car. We refer to such requests from passengers in the cars as *car calls*. The elevator cars $C2$ and $C3$ are moving downwards and upwards to service their respective passenger requests. At each floor, we have hall call buttons in upward and downward

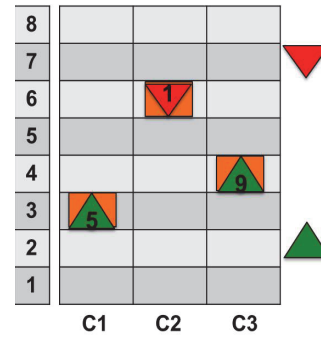


Figure 1: A typical elevator system with 3 elevator cars and 8 floors. The three elevator cars $C1$, $C2$, and $C3$ are moving to serve the requests of the passengers that are in their cars. Passengers indicate the preference for their rides by pressing the upward or downward buttons.

directions. In a building with F floors we can have a maximum of $2F - 2$ hall calls (there can be hall calls in only one direction at the lobby and the top floors) at any time. With C elevator cars, we can have a maximum of C^{2F-2} assignments of hall calls to elevators.

Given this incomplete information of the waiting passengers and the huge search space, the *scheduler* in an elevator system has the responsibility of *assigning* a particular elevator to service each of the hall calls. In general, the schedulers assign the hall calls to elevator cars to optimize some criteria such as the average waiting time, total travel time of the passengers, energy consumption, etc. In addition, the potential human-system interactions only serve to complicate the decision-making as even a gentle gesture of keeping the elevator door open for arriving passengers can render a particular assignment suboptimal. As a result, the schedulers must constantly improvise and potentially change the assignments of hall calls to elevators as new information is revealed over time. In fact, such reassignments are completely oblivious to the waiting passengers. Consequently, the problem of group elevator scheduling is a hard problem and continues to be researched due to its practical significance.

In this paper, we address the problem of assigning the hall calls to elevator cars so as to minimize the average waiting

time (AWT) of the passengers. The waiting time of a single passenger is the time taken from the moment the passenger presses the hall call button at a floor to the instant at which the passenger is picked up.

We address the more prevalent elevator systems wherein:

- Hall calls do not provide information on destination floors.
- Number of passengers behind a hall call is not known.
- No prediction of future passengers is available.
- Reassignment of hall calls is allowed.

In this setting, it is not possible to compute the **exact waiting time**. Consequently, we use information on the current car calls and the motion of the elevator car to compute an **estimate of the waiting times**. Reassignment of hall calls to elevators, wherein the scheduler can modify the previous assignments as long as the hall call has not been serviced, helps to mitigate the effect of the uncertainty in the hall calls. We propose an approximation algorithm to determine the assignments that minimize the estimated average waiting times.

Prior Work

Most of the existing methods address one or more of the following scenarios:

- Sensing and User Inputs: Algorithms that rely on additional user-interface devices that allow the users to enter their destination floors, or the use of sensors to detect the number of passengers waiting at each floor.
- Traffic-patterns: Algorithms that are tailor-made for specific traffic patterns.
- Design Criteria: Algorithms that are motivated by different design criteria such as estimated time of arrival, average waiting time, etc.

Sensing and User Inputs

In a destination control elevator system, the passengers provide the destination information to the scheduler before they board the cars. Several AI techniques (Koehler and Ottiger 2002) have been used to address the scheduling of destination control systems, and this problem has been shown to be NP-hard even in the simplest settings (Seckinger and Koehler 1999). In the case of full destination control scenario, the problem of group elevator scheduling can be formulated using planning domain definition language (PDDL) (Koehler and Schuster 2000; Koehler and Ottiger 2002).

In the presence of complete information, elevator scheduling problem has been addressed using mixed integer linear programming (MILP) (Ruokokoski, Ehtamo, and Pardalos 2015; Xu and Feng 2016). Genetic algorithms have also been proposed to generate control policy for elevator scheduling (Dai et al. 2010).

In studies dating back to 1995, computer vision algorithms have been proposed to automatically count the number of passengers waiting at each floor, but they are not commonly used due to the cost of retro-fitting existing instal-

lations (Schofield, Stonham, and Mehta 1995). The availability of complete information may become feasible in the case of new and high-rise buildings, but the market for such buildings is much smaller than existing installations. Hence, improving the performance of schedulers with incomplete information continues to be a very relevant and challenging problem to date.

Traffic-patterns

Elevator traffic patterns can broadly be classified into three types: up-peak, down-peak, and inter-floor. In the up-peak case, passengers always enter the elevator system at the lobby and request upward rides. In the case of down-peak, the passengers request downward rides from all floors to the lobby. In the case of inter-floor the passengers request upward or downward rides between different floors, but not to or from the lobby. For up-peak traffic case, special purpose algorithms are developed based on queuing theory (Pepyne and Cassandras 1997). In the down-peak case, efficient algorithms have been developed using Finite Inter-visit Minimization (FIM) and Empty the System Algorithm (ESA) (Bao et al. 1994). The intensity of each traffic pattern can be represented using fuzzy variables and fuzzy logic can be used for the assignment of cars to hall calls (Ujihara and Tsuji 1988; Ujihara and Amano 1994). In particular, the fuzzy rules can be used to identify the specific pattern and this can allocate a specific control algorithm for scheduling. Most of the time, the traffic patterns can be seen as the combination of these three basic patterns, e.g., at lunch time it could be 45 % down-peak, 45 % up-peak, and 10 % inter-floor. One of the biggest challenges in exploiting this traffic pattern is the difficulty in accurate detection and the high frequency at which these patterns change.

Neural networks and reinforcement learning methods have also been used for elevator scheduling problems (Crites and Barto 1996; Markon, Kita, and Nishikawa 1994). In particular, Crites and Barto combined neural networks and Q-learning to demonstrate improved performance over FIM and ESA for one specific down-peak scenario (Crites and Barto 1996). However, it took about 60,000 hours of training time and thereby impractical for real elevator systems. On the other hand, with the recent progress in deep learning methods with the use of GPUs, this might be an interesting future direction.

There have been algorithms that use dynamic programming to compute the exact minimization of average waiting time (AWT) (Nikovski and Brand 2003). Their approach was to model it as Markov decision process (MDP). However, they assume that the number of passengers waiting behind a hall call is known.

Design Criteria

One of the popular elevator scheduling policy is collective control, where the cars stop at the nearest call in the running direction (Strakosch 1998). In particular, the system computes the travel distances between hall calls and elevators. The scheduler assigns the hall call to the "nearest" car in terms of the distance. This strategy is not optimal and usu-

ally results in bunching, where several cars arrive at the same floor at the same time.

We can also allocate hall calls by minimizing the estimated time of arrival (ETA) instead of the distance. To avoid the problem of bunching, we can use other approaches such as zoning or sectoring, where we partition the building into zones and each car can serve only one zone. In other words, we consider the same number of zones as the number of elevators. This strategy improves performance when the traffic is downwards towards the lobby. It has been shown that this strategy is also suboptimal when too many passengers arrive in the same zone (Barney and dos Santos 1985; Strakosch 1998).

Relative system response (RSR) is a scheduling strategy used by Otis Elevator company to optimize a heuristic criterion based on a weighted sum of penalties and bonuses computed for every car, without any explicit relation to the actual AWT (Bittar 1982). Another criteria used by Otis that is well correlated with AWT is remaining response time (RRT), which is the time required by a car to reach the floor of the hall call (Powell and Williams 1992). Approximate methods have been used to compute the average waiting time (AWT) by predicting the number of stops for passenger pickups and the most likely floor at which the car reverses its direction (Siikonen 1997; Cho, Gagov, and Kwon 1999; Barney and dos Santos 1985).

Our Contributions

- To the best of our knowledge, we are not aware of any group elevator scheduling algorithm that is based on submodularity. In this work, we show that the problem of group elevator scheduling can be formulated as the maximization of submodular functions under a matroid constraint. Submodularity is one of the key concepts in machine learning, computer vision, economics, and game theory. Recently it has found application in such diverse domains as: sensor placement (Guestin, Krause, and Singh 2008), outbreak detection (Leskovec et al. 2007), word alignment (Lin and Bilmes 2011), clustering (Liu et al. 2013), viral marketing (Kempe, Kleinberg, and Tardos 2003), and finding diverse subsets in structured item sets (Prasad, Jegelka, and Batra 2014).
- Our formulation enables the use of a simple greedy algorithm with guarantees on the optimality of the solution.
- We use as objective a quadratic Boolean function that approximates the average waiting times of passengers.
- Existing efforts in the literature demonstrate their advantages using customized simulation data, which makes it harder to evaluate and understand their usefulness. We evaluate using Elevate 8 and show consistent and significant improvement over the implemented standard scheduling algorithms in Elevate 8 over a wide variety of elevator settings.

Background

In this section, we briefly define useful entities such as set functions, submodularity, and matroids.

Notations: Let \mathbb{B} denote the Boolean set $\{0, 1\}$ and \mathbb{R} the set of reals. We use \mathbf{x} to denote vectors.

Definition 1. A set function $F : 2^E \rightarrow \mathbb{R}$, where E is a finite set, maps a set to a real number. A set function can also be seen as a pseudo-Boolean function (Boros and Hammer 2002) that takes a Boolean vector as argument and returns a real number.

Definition 2. A set function $F : 2^E \rightarrow \mathbb{R}$ is submodular if for all $A, B \subseteq E$ with $B \subseteq A$ and $e \in E \setminus A$, we have:

$$F(A \cup \{e\}) - F(A) \leq F(B \cup \{e\}) - F(B). \quad (1)$$

This property is also referred to as diminishing return since the gain from adding an element to a small set B is never smaller than adding it to a superset $A \supset B$ (Nemhauser, Wolsey, and Fisher 1978).

Definition 3. A set function F is monotonically increasing if for all $A, B \subseteq E$ and $B \subseteq A$, we have:

$$F(B) \leq F(A). \quad (2)$$

Definition 4. A matroid is an ordered pair $M = (E, \mathcal{I})$ consisting of a finite set E and a collection \mathcal{I} of subsets of E satisfying the following three conditions:

1. $\emptyset \in \mathcal{I}$.
2. If $I \in \mathcal{I}$ and $I' \subseteq I$, then $I' \in \mathcal{I}$.
3. If I_1 and I_2 are in \mathcal{I} and $|I_1| < |I_2|$, then there is an element $e \in I_2 - I_1$ such that $I_1 \cup \{e\} \in \mathcal{I}$.

The members of \mathcal{I} are the independent sets of M . Note that there exist several other definitions for matroids that are equivalent. For more details, one can refer to (Oxley 1992).

Definition 5. Let E_1, \dots, E_n be the partition of the finite set E and let k_1, \dots, k_n be positive integers. The ordered pair $M = (E, \mathcal{I})$ is a partition matroid if:

$$\mathcal{I} = \{I : I \subseteq E, \quad |I \cap E_i| \leq k_i, 1 \leq i \leq n\}. \quad (3)$$

Problem Formulation

In this section, we formulate group elevator scheduling as a quadratic Boolean optimization problem. The qualifier “quadratic” is due to our choice of model for the waiting times of passengers.

To set the stage, suppose there are N hall calls that need to be assigned to elevators. Note that these N hall calls include previously assigned calls that are considered for reassignment and new hall calls that have just requested service. We do not allow reassignment when the assigned car is already close to servicing the assigned hall call.

The decision variables in the problem are denoted by x_i^c . The variable x_i^c is a Boolean variable to denote the assignment of the hall call i to elevator car c :

$$x_i^c = \begin{cases} 1 & \text{if the hall call } i \text{ is assigned to car } c, \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

Let $\mathbf{x} \in \{0, 1\}^{N \cdot C}$ denote the vector storing all the assignments $\{x_1^1, x_2^1, \dots, x_N^1, \dots, x_N^C\}$. In any *feasible assignment* of hall calls to elevator cars, each hall call must be

assigned to exactly one elevator car. This can be imposed through the constraint,

$$\sum_{c=1}^C x_i^c = 1, \quad \forall i \in \{1, \dots, N\}. \quad (5)$$

The number of feasible assignments is C^N . If we denote the waiting time associated with a feasible assignment \mathbf{x} as $w^{\text{true}}(\mathbf{x})$ then the group elevator scheduling problem can be posed as minimizing $w^{\text{true}}(\mathbf{x})$ over all feasible assignments \mathbf{x} . Clearly, computing the terms in the objective of such an optimization problem is an onerous task and is infeasible for real elevator systems in which decisions are typically made every second or fractions of a second.

To alleviate this bottleneck we propose the following approximation for the total waiting time to serve the hall calls,

$$g(\mathbf{x}) = \sum_{i=1}^N \sum_{c=1}^C w_i^c x_i^c + \sum_{i=1}^N \sum_{j=i+1}^N \sum_{c=1}^C w_{ij}^c x_i^c x_j^c \quad (6)$$

where w_i^c, w_{ij}^c are computed based on the current state of the elevator car c as explained next. The term w_i^c is the waiting time for car c to pick up the passenger(s) for hall call i given the current set of car calls for the car c . Consider the assignment of the hall calls to car C1 in Figure 1. We first consider the assignment of hall call on floor 3 to car C1 as shown in Figure 2(a). Since C1 already has a car call to floor 5 the waiting time for the hall call can be computed as

$$w_i^c = t_{3 \rightarrow 5} + t_{\text{door}} + t_{5 \rightarrow 2}$$

where $t_{i \rightarrow j}$ represents the time to travel from floor i to floor j and t_{door} includes the sum of door opening, dwell and closing times. Car C1 moves from a state of rest to state of rest when it travels from floor 5 to floor 2. On other hand, for the journey from floor 3 to 5 the car may already be in motion or at rest. Hence, the time required needs to be computed according to the kinematic state of the elevator car. Such kinematic formulas have been derived in (Peters 1993) as a function of the velocity, acceleration and jerk of the car and the maximum limits for the same. Note that the other hall calls are completely ignored in performing this computation. We refer to this as *unary* term since the influence of other hall calls are completely ignored in the computation. In a similar manner, we can compute the waiting time w_j^c for picking the passenger requesting hall call from floor 7 (refer Figure 2(b)).

The term w_{ij}^c represents the excess over the $w_i^c + w_j^c$ that is incurred when both hall calls i and j are assigned to the same car c . We refer to this term as *pairwise* term. In other words, $(w_i^c + w_j^c + w_{ij}^c)$ is the total waiting time to pick up passengers for hall calls on floors i and j given the current set of car calls for car c and can be obtained as

$$w_{ij}^c = t_{3 \rightarrow 5} + t_{\text{door}} + t_{5 \rightarrow 7} + t_{\text{door}} + t_{7 \rightarrow f} + t_{\text{door}} + t_{f \rightarrow 2} - w_i^c - w_j^c$$

where f represents the *unknown* destination floor of the passenger on floor 7. In the computation of w_{ij}^c we have chosen to serve the hall call at floor 7 before serving the hall call at floor 2. This choice is dictated by typical elevator movement rules. Once a car is empty,

- The car maintains its upward (downward) direction of motion if there are up (down) hall calls at floors above (below) the current floor of the elevator car.
- If not, the car moves to the highest (lowest) floor at which downward (upward) hall call exists.
- If not, then the car moves downward (upward) until the lowest (highest) floor with hall call in the upward (downward) direction exists.

Unlike in the computation of the unary terms, notice that we require knowledge of the destination floor f for the first hall call that is serviced. Since this information is not available we instead compute the pairwise term as an expectation over all possible destination floors for that hall call,

$$w_{ij}^c = \sum_{f \in \{1, \dots, 6\}} \omega_f \cdot \left(t_{3 \rightarrow 5} + t_{\text{door}} + t_{5 \rightarrow 7} + t_{\text{door}} + t_{7 \rightarrow f} + t_{\text{door}} + t_{f \rightarrow 2} \right) - w_i^c - w_j^c$$

where ω_f is the probability of floor f being the destination. If there exists additional information on the probabilities these can be easily incorporated in the computation for the pairwise term. However, in the absence of such information we assume that destination floors are equally likely. Observe that the set of destination floors is only $\{1, \dots, 6\}$ since the hall call requests downward service from floor 7. As in the computation of the unary terms, the remaining hall calls are ignored in performing this computation. We can state the following result on the pairwise term.

Lemma 1. $w_{ij}^c \geq 0$.

Proof. Suppose two hall calls i, j are assigned to the same car and that hall call i is serviced first. The waiting time for hall call i is exactly w_i^c . If the second hall call is also on the same floor then the total waiting time is $w_i^c + w_j^c$. If not the waiting time for hall call j is greater than w_j^c since the car makes an intermediate stop and there is also time associated with door operation to pick up the passenger for hall call i . Hence, the pairwise term is always nonnegative. \square

Clearly, $g(\mathbf{x})$ is *exact* when no more than 2 hall calls are assigned to each elevator car. For higher number of assignments to a car this is only a *proxy* for the actual waiting times of the passengers. However, our choice of this quadratic form for $g(\mathbf{x})$ is motivated by:

- *nice* properties that (we prove in the next section) that allow us to derive simple algorithms with provable guarantees
- reduced computational effort in computing the objective - $O(N \cdot C + N(N-1) \cdot C)$.

Let \mathbf{w} denote the vector storing all the weights $\{w_1^1, \dots, w_1^C, w_2^1, \dots, w_N^C, w_{12}^1, \dots, w_{N-1N}^C\}$.

We write the group elevator scheduling problem to minimize the waiting time as the following quadratic Boolean

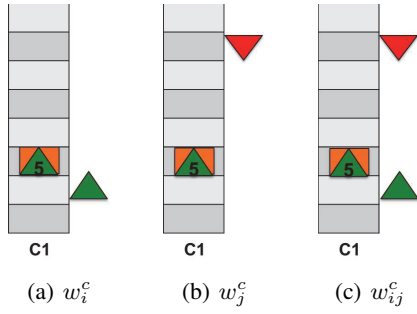


Figure 2: Computation of unary and pairwise terms.

optimization problem:

$$\begin{aligned} \min_{\mathbf{x}} \quad & g(\mathbf{x}) \\ \text{s.t.} \quad & \sum_{c=1}^C x_i^c = 1, \quad \forall i \in \{1, \dots, N\}, \\ & \mathbf{x} \in \{0, 1\}^{N \cdot C}. \end{aligned} \quad (7)$$

Submodular Maximization & Matroid

In the section we show that the quadratic Boolean optimization in Equation (7) can be posed as a submodular maximization problem over a matroid. To begin with, we pose the problem in Equation (7) as a maximization using negations in the objective function:

$$\begin{aligned} \max_{\mathbf{x}} \quad & -g(\mathbf{x}) \\ \text{s.t.} \quad & \sum_{c=1}^C x_i^c = 1, \quad \forall i \in \{1, \dots, N\}, \\ & \mathbf{x} \in \{0, 1\}^{N \cdot C}. \end{aligned} \quad (8)$$

We introduce some related notation on set functions that will allow us to formulate the problem over sets and matroid constraints.

Let us consider a finite set E comprising the assignments:

$$E = \{1 \rightarrow 1, \dots, N \rightarrow 1, \dots, 1 \rightarrow C, \dots, N \rightarrow C\}$$

where $i \rightarrow c$ denotes that hall call i is assigned to car c . The objective function $g(\mathbf{x})$ can be equivalently defined over the subsets of E as follows. Let $h : 2^E \rightarrow \mathbb{R}$ be a set function defined for any set $A \subseteq E$ as:

$$h(A) = -g(\mathbf{x}), \quad (9)$$

$$\text{where } x_i^c = \begin{cases} 1 & i \rightarrow c \in A, \\ 0 & \text{otherwise} \end{cases}, \quad \forall \begin{cases} c \in \{1, \dots, C\}, \\ i \in \{1, \dots, N\} \end{cases}. \quad (10)$$

Lemma 2. *The function $h(A)$ is submodular.*

Proof. Consider two sets A and B where $B \subseteq A$ and $A, B \subseteq E$. Let $i \rightarrow c \in E \setminus A$. We can observe that:

$$h(A \cup \{i \rightarrow c\}) - h(A) = -w_i^c - \sum_{(j,c): j \rightarrow c \in A} w_{ij}^c$$

$$h(B \cup \{i \rightarrow c\}) - h(B) = -w_i^c - \sum_{(j,c): j \rightarrow c \in B} w_{ij}^c$$

Since unary and pairwise terms are non-negative (refer Lemma 1) and $B \subseteq A$,

$$h(A \cup \{i \rightarrow c\}) - h(A) \leq h(B \cup \{i \rightarrow c\}) - h(B)$$

which proves the claim. \square

Now we consider the constraint in Equation (5) modeling the assignment of a hall call to exactly one elevator car. We show that this can be formulated using partition matroid constraint. In order to do that, let us consider N disjoint subsets of the ground set E as shown below:

$$\begin{aligned} E_1 &= \{1 \rightarrow 1, \dots, 1 \rightarrow C\} \\ &\vdots \\ E_N &= \{N \rightarrow 1, \dots, N \rightarrow C\}. \end{aligned}$$

Let $M = (E, \mathcal{I})$ denote a partition matroid such that each independent set shares no more than one element with each of the disjoint subsets as shown below:

$$|I \cap E_i| \leq 1, \quad I \in \mathcal{I}, \forall i \in \{1, \dots, N\}. \quad (11)$$

Let us consider the following submodular function maximization under a partition matroid constraint:

$$\begin{aligned} \max_{A \subseteq E} \quad & h(A) \\ \text{s.t.} \quad & A \in \mathcal{I}, M = \{E, \mathcal{I}\}. \end{aligned} \quad (\text{SFM-1})$$

Note that the above optimization is different from the maximization problem given in Equation 8 because the constraints are different. The partition matroid only enforces that each hall call is assigned to no more than one elevator car (i.e., $|A \cap E_i| \leq 1$). However, it does not enforce that each hall call is assigned to at least one elevator car (i.e., $|A \cap E_i| \geq 1$ is not enforced). In order to do that, we add a penalty term such that an assignment that violates equation (5) (i.e. $|A \cap E_i| < 1$) has a lower objective than any feasible assignment. This will ensure that each hall call is assigned to at least one elevator car. We consider the following additional term $h_1(A)$ in the objective function:

$$h_1(A) = - \sum_{i=1}^N p_i \cdot (C - |A \cap E_i|) \quad (12)$$

$$\text{where, } p_i = \max_{c \in \{1, \dots, C\}} \left(w_i^c + \sum_{j=1, j \neq i}^N w_{ij}^c \right).$$

The term $h_1(A)$ reduces the objective function by $C \sum_{i=1}^N p_i$ if there is no assignment, i.e., $A = \emptyset$. Since $p_i \geq 0$, we increase the objective function for every assignment of hall call to a car ($i \rightarrow c$) included in the set A . Further, the function $h_1(A)$ satisfies the following property for any $i \rightarrow c \notin A$,

$$h_1(A \cup \{i \rightarrow c\}) - h_1(A) = p_i. \quad (13)$$

The following lemma shows that the combined objective function is submodular.

Lemma 3. *The function $h(A) + h_1(A)$ is submodular.*

Proof. Consider two sets $A, B \subset E$ with $B \subset A$. Let $i \rightarrow c \in E \setminus A$. Then, by Equation (13) we have that

$$h_1(A \cup \{i \rightarrow c\}) - h_1(A) = h_1(B \cup \{i \rightarrow c\}) - h_1(B) = p_i.$$

In other words, $h_1(A)$ satisfies the condition for submodularity (Equation (1)) in Definition 2 as an equality. The sum of two submodular functions is submodular and thus $h(A) + h_1(A)$ is submodular. \square

We consider the following submodular function maximization problem that is equivalent to the quadratic Boolean optimization problem in Equation 8.

$$\begin{aligned} \max_{A \subseteq E} h(A) + h_1(A) \\ \text{s.t. } A \in \mathcal{I}, M = \{E, \mathcal{I}\}. \end{aligned} \quad (\text{SFM-2})$$

In this work, we use a greedy algorithm to solve the maximization of the submodular function under the matroid constraint. The use of greedy algorithm is motivated by the following theorem:

Theorem 1. (Nemhauser, Wolsey, and Fisher 1978) *For maximizing monotonically non-decreasing submodular functions under a matroid constraint, the optimality of the greedy algorithm is characterized by the following equation:*

$$f(A_{\text{greedy}}) \geq \frac{1}{2} f(A_{\text{OPT}}), \quad (14)$$

where $f(\emptyset) = 0$.

In the submodular optimization problem given in Equation (SFM-2), the objective function is not equal to zero when $A = \emptyset$. The optimality bound given in Theorem 1 applies for submodular function maximization where the function is equal to zero when the solution is an empty set. From Equation (12), $h(\emptyset) + h_1(\emptyset) = -C \sum_{i=1}^N p_i$. By adding a constant term ($C \sum_{i=1}^N p_i$) we can ensure that the requirement of Theorem 1 can be satisfied. Further, the addition of a constant to a submodular function ($h(A) + h_1(A)$), also ensures submodularity of the resulting function $h(A) + h_1(A) + C \sum_{i=1}^N p_i$ (refer Definition 2). We consider the following optimization problem which is equivalent to SFM-2:

$$\begin{aligned} \max_{A \subseteq E} h(A) + h_1(A) + C \sum_{i=1}^N p_i \\ \text{s.t. } A \in \mathcal{I}, M = \{E, \mathcal{I}\}. \end{aligned} \quad (\text{SFM-3})$$

In order to apply Theorem 1 to the above optimization problem, we also have to show that the objective function is monotonically non-decreasing.

Lemma 4. *The function $h(B) + h_1(B) + C \sum_{i=1}^N p_i$ is monotonically non-decreasing.*

Proof. The constant term $C \sum_{i=1}^N p_i$ does not affect the monotonicity of a function. Thus we need to only show that $h(B) + h_1(B)$ is monotonically non-decreasing for any

$B \subset E$. Suppose that $i \rightarrow c \notin B$ is added to the set B . From Equations (11) and (13),

$$\begin{aligned} h(B \cup \{i \rightarrow c\}) + h_1(B \cup \{i \rightarrow c\}) - h(B) - h_1(B) \\ = -w_i^c - \sum_{(j,c):j \rightarrow c \in B} w_{ij}^c + p_i \end{aligned}$$

Since the unary and pairwise terms are non-negative (refer Lemma 1), it can be readily seen that:

$$\begin{aligned} p_i &= \max_{c \in \{1, \dots, C\}} \left(w_i^c + \sum_{j=1, j \neq i}^N w_{ij}^c \right) \\ &\geq w_i^c + \sum_{(j,c):j \rightarrow c \in B} w_{ij}^c \\ \implies -w_i^c - \sum_{(j,c):j \rightarrow c \in B} w_{ij}^c + p_i &\geq 0. \end{aligned}$$

Hence

$$h(B \cup \{i \rightarrow c\}) + h_1(B \cup \{i \rightarrow c\}) \geq h(B) + h_1(B). \quad (15)$$

Further any set A with $B \subseteq A \subseteq E$ can be obtained by incrementally adding to set B the elements in $A \setminus B$. Thus, iterative application of Equation (15) yields

$$h(B) + h_1(B) \leq h(A) + h_1(A)$$

proving the claim. \square

For the sake of completeness, we list the steps of the greedy algorithm used in maximization of submodular function under a matroid constraint (Nemhauser, Wolsey, and Fisher 1978). Let the objective function in Equation (SFM-3) be denoted by $f(A) = h(A) + h_1(A) + C \sum_{i=1}^N p_i$.

Greedy Algorithm:

1. Initialize $S = \emptyset$.
2. Let $s = \arg \max_{s' \in E} f(S \cup \{s'\}) - f(S)$ such that $S \cup \{s'\} \in \mathcal{I}$.
3. If $s \neq \emptyset$ then $S = S \cup \{s\}$ and go to step 2.
4. S is the required solution.

Note that the objective function used in Equation (SFM-3) is not just the average waiting time. We also add a penalty $h_1(A)$ and a constant term $C \sum_{i=1}^N p_i$. Thus the actual bound on the optimality using greedy algorithm is

$$h(A_{\text{greedy}}) + \sum_{i=1}^N p_i \geq \frac{1}{2} \left(h(A_{\text{OPT}}) + \sum_{i=1}^N p_i \right) \quad (16)$$

Note the penalty term evaluates to $h_1(A) = -(C - 1) \sum_{i=1}^N p_i$ for valid assignments where every hall call is assigned to one elevator car. Taking into account the constant term $C \sum_{i=1}^N p_i$ yields the offset in Equation (16).

We can also consider higher order terms to impose penalty on non-balanced assignment of passengers to elevator cars,

i.e., assigning most of the passengers to single elevator cars. For example, we can impose a penalty whenever three passengers are assigned to a single elevator car. This penalty can be imposed using a third degree term, i.e., $x_i^c x_j^c x_l^c$. By adding a higher order term of order k with negative coefficients to the function $h(A)$ the resulting function is known to be submodular (Boros and Hammer 2002; Kolmogorov and Zabih 2004).

Experiments

We implemented our submodular maximization based greedy algorithm within the elevator simulator **Elevate 8**¹. This is a commercial-grade simulator that allows the selection of the number of floors, the number of elevator cars, the speed/acceleration of cars, height of the floors, etc. It allows the user to choose different traffic patterns such as up-peak, down-peak, and inter-floor. In particular, the simulator provides several industry-standard methods such as group collective control, estimated time of arrival (ETA), destination control, etc.

In addition to the details described earlier, we outline a few other enhancements to our algorithm:

- We considered only non-destination control scenarios. Since we do not know the destination floors, we consider all possible destinations and use their average to compute the pairwise term.
- For the elevator cars that are close to capacity, we use a penalty term to avoid assigning additional passengers. This is achieved by using a high unary cost for assigning additional passengers to these elevator cars. This ensures that these calls are assigned by the greedy algorithm in the later stage. After the greedy assignment, we remove these assignments from the respective elevators.
- Door status determines the amount of time that elapses before the elevator can move away from a floor. This is important for correct assignment in low traffic conditions. The simulator provides information on the door status and we appropriately include the additional time due to door operation.
- We give a bonus to hall calls assigned to a car with existing car calls for the hall call floor. The bonus is provided in the form of reduction of the unary term associated with the particular assignment. The reduction is specified as,
$$w_i^c = w_i^c - \min(0.2 \cdot w_i^c, 10). \quad (17)$$
- We penalize assignments of too many hall calls to the same elevator car. This is achieved using higher order terms. For example, by adding the term $w_{i_1 i_2 \dots i_k}^c x_1^c x_2^c \dots x_k^c$ in the objective function we increase the waiting time by $w_{i_1 i_2 \dots i_k}^c$ if k passengers board the same elevator car c .

It is important to note that the above changes to the cost function still preserve submodularity.

We used the following experimental setup to evaluate the different scheduling algorithms. We studied:

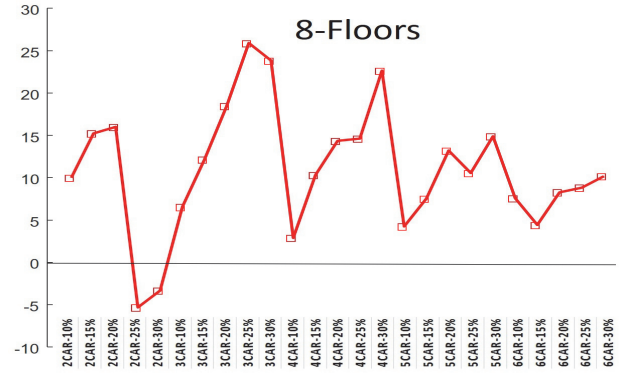


Figure 3: We show the percentage decrease in the average waiting time using pairwise terms.

- three different buildings with 8, 10, and 12 floors
- for each building we consider a 1-hour period of inter-floor traffic scenario with 5 different arrival rates specified as percentage of passengers in 5 minutes - 10%, 15%, 20%, 25% and 30%
- for each building we considered 2, 3, 4, 5, and 6 elevator cars.

For each specific building, arrival rate of passengers and fixed number of elevator cars, we measure the average waiting time achieved by the scheduling algorithm as the average over 10 different instances of the traffic scenario.

We compare our methods with group collective control and estimated time of arrival (ETA) methods. The said methods are known to work well under different traffic conditions and elevator settings. The implementation in Elevate 8 employs several heuristics for improved performance: accounting for future demands, priority for coincident calls, not stopping an elevator car when it is full, intelligent decisions about keeping the door open before getting the hall call, detecting up-peak or down-peak scenarios automatically, etc. However, such heuristics are not included in an explicit objective function and it is hard to determine the impact of some of these improvements. In all our experiments, we consistently outperform these methods in a wide variety of elevator settings.

We begin by describing an ablation study on our algorithm. Figure 3 plots the percentage savings in average waiting time when using both unary and pairwise terms in the objective function as opposed to using only the unary term in the objective. For the latter case we simply set $w_{ij}^c = 0$. The plot provides the savings at different number of cars and arrival rates of traffic. In this study, we did not include the bonus specified in Equation (17). Figure 3 clearly shows that pairwise term is critical to obtaining higher savings in average waiting time across all buildings. The average reduction over all scenarios is about 10.9 %.

Figure 4 considers the effect of including the bonus in Equation (17). The plot shows the reduction in average waiting times for the objective using unary and pairwise terms with the bonus from Equation (17) over the case where the

¹<https://www.peters-research.com/index.php/8-elevate/58-elevate-8>

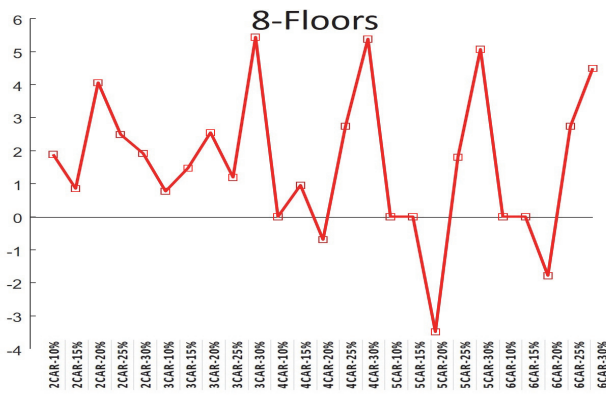


Figure 4: We show the percentage decrease in the average waiting time using coincident call bonus.

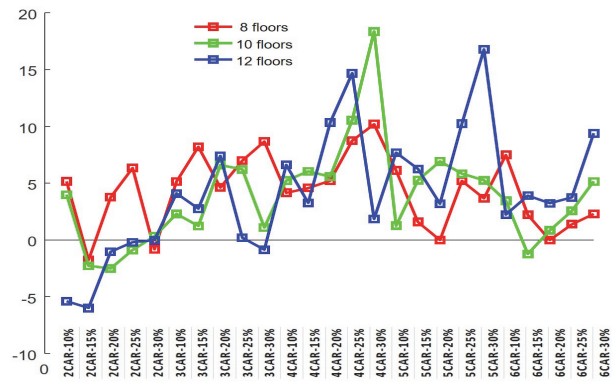


Figure 6: We show the percentage decrease in the average waiting time of our submodular approach with respect to ETA.

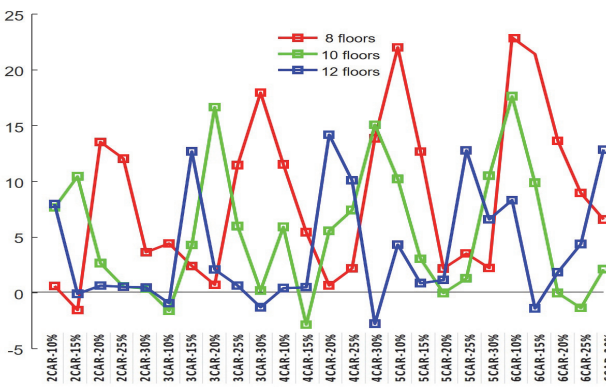


Figure 5: We show the percentage decrease in the average waiting time of our submodular approach with respect to the group collective control.

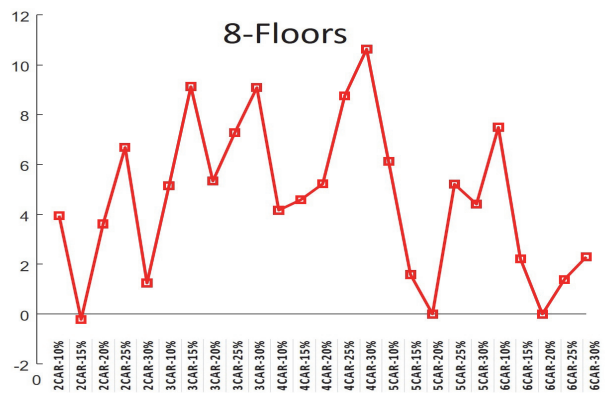


Figure 7: We show the percentage decrease in the average waiting time using higher order submodular function with respect to ETA.

bonus is not included. The average reduction over all scenarios is about 1.6 %.

We now compare our scheduling algorithms against the ones in Elevate. Figure 5 plots the reduction in average waiting time over group collective control for different arrival rates and elevators for three different buildings. For the case of 8-floors we obtain an average reduction of 8.6 %, 5.3 % for 10-floors and 3.9 % for 12-floors.

Figure 6 plots the reduction in average waiting time over ETA for different arrival rates and elevators for three different buildings. For the case of 8-floors we obtain an average reduction of 4.4 %, 3.9 % for 10-floors and 4.2 % for 12-floors. To illustrate the utility of higher-order terms, we use a small penalty for discouraging the assignment of 4 or 5 passengers to the same car and obtain further reduction in waiting time. For the case of 8-floors we obtain an average reduction of 4.6 % over ETA as shown in Figure 7.

Discussion

We show a novel method for solving the group elevator scheduling problem by formulating it as the maximization of submodular functions under a matroid constraint. Our

method consistently outperforms other industry-standard methods in a wide variety of elevator settings.

In the future, we plan to investigate alternative methods that could directly maximize non-monotonous submodular functions (Feige, Mirrokni, and Vondrak 2007). This will allow us to derive improved guarantees on the optimality of the solution. The use of explicit objective function to model several design criteria also opens up the possibility of other integer programming methods for finding globally optimal solutions.

References

- Bao, G.; Cassandras, C. G.; Djaferis, T. E.; Gandhi, A. D.; and Looze, D. P. 1994. Elevator dispatchers for down-peak traffic. *Technical report, University of Massachusetts, Department of Electrical and Computer Engineering, Amherst, Massachusetts.*
- Barney, G., and dos Santos, S. 1985. Elevator traffic analysis, design and control. *England: IEE, Peter Peregrinus Ltd.*

- Bittar, J. 1982. Relative system response elevator call assignments. *US Patent*. 4,363,381.
- Boros, E., and Hammer, P. L. 2002. Pseudo-boolean optimization. *Discrete Appl. Math.* 123(1-3):155–225.
- Cho, Y.; Gagov, Z.; and Kwon, W. 1999. Elevator group control with accurate estimation of hall call waiting times. In *In Proceedings of the 1999 International Conference on Robotics and Automation*, 447–452.
- Crites, R. H., and Barto, A. G. 1996. Improving elevator performance using reinforcement learning. In *Advances in Neural Information Processing Systems*, 1017–1023.
- Dai, D.; Zhang, J.; Xie, W.; Yin, Z.; and Zhang, Y. 2010. Elevator group control policy with destination registration based on hybrid genetic algorithms. In *International Conference on Computer Application and System Modeling*.
- Feige, U.; Mirrokni, V. S.; and Vondrak, J. 2007. Maximizing non-monotone submodular functions. In *Foundations of Computer Science (FOCS)*.
- Guestrin, C.; Krause, A.; and Singh, A. P. 2008. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research* 235–284.
- Kempe, D.; Kleinberg, J.; and Tardos, E. 2003. Maximizing the spread of influence through a social network. In *KDD*.
- Koehler, J., and Ottiger, D. 2002. An ai-based approach to destination control in elevators. *AI Magazine* 23(3):59–79.
- Koehler, J., and Schuster, K. 2000. Elevator control as a planning problem. In *AI Planning and Scheduling (AIPS)*.
- Kolmogorov, V., and Zabih, R. 2004. What energy functions can be minimized via graph cuts? *PAMI* 26(2).
- Leskovec, J.; Krause, A.; Guestrin, C.; Faloutsos, C.; VanBriesen, J.; and Glance, N. 2007. Cost-effective outbreak detection in networks. In *The ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 420–429.
- Lin, H., and Bilmes, J. 2011. Word alignment via submodular maximization over matroids. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Short Papers*, 170–175.
- Liu, M. Y.; Tuzel, O.; Ramalingam, S.; and Chellappa, R. 2013. Entropy rate clustering: Cluster analysis via maximizing a submodular function subject to a matroid constraint. *IEEE Transaction on Pattern Analysis and Machine Intelligence (TPAMI)*.
- Markon, S.; Kita, H.; and Nishikawa, Y. 1994. Adaptive optimal elevator group control by use of neural networks. *Trans. Inst. Syst. Contr. Inform. Eng.* 7:487–497.
- Nemhauser, G. L.; Wolsey, L. A.; and Fisher, M. L. 1978. An analysis of the approximations for maximizing submodular set functions. *Mathematical Programming* 265–294.
- Nikovski, D., and Brand, M. 2003. Decision-theoretic group elevator scheduling. In *International Conference on Automated Planning and Scheduling (ICAPS)*.
- Oxley, J. 1992. *Matroid Theory*. Oxford University Press.
- Pepyne, D., and Cassandras, C. 1997. Optimal dispatching control for elevator systems during uppeak traffic. *IEEE transactions on control systems technology* 5(6):629–643.
- Peters, R. 1993. Ideal lift kinematics: Formulae for the equations of motion of a lift. Technical report, Internal Brunel University/Arup Research and Development paper.
- Powell, B. A., and Williams, J. N. 1992. Elevator dispatching based on remaining response time. *US Patent* 5,146,053.
- Prasad, A.; Jegelka, S.; and Batra, D. 2014. Submodular meets structured: Finding diverse subsets in exponentially-large structured item sets. In *Neural Information Processing Systems (NIPS)*.
- Ruokokoski, M.; Ehtamo, H.; and Pardalos, P. M. 2015. Elevator dispatching problem: a mixed integer linear programming formulation and polyhedral results. *Journal of Combinatorial Optimization* 29(4).
- Schofield, A.; Stonham, T.; and Mehta, P. 1995. A machine vision system for counting people. In *In New Methods and Technologies in Planning and Construction of Intelligent Buildings*, ed. A. Lustig, 5059. Haifa, Israel: IB/IC Intelligent Building Congress.
- Seckinger, B., and Koehler, J. 1999. Online synthesis of elevator controls as a planning problem. In *Thirteenth Workshop on Planning and Configuration, Technical Report, Department of Computer Science, University of Wuerzburg*.
- Siikonen, M. 1997. Elevator group control with artificial intelligence. *Technical Report A67, Helsinki University of Technology, Systems Analysis Laboratory, Helsinki, Finland*.
- Strakosch, G. R. 1998. Vertical transportation: elevators and escalators. *New York, NY: John Wiley & Sons, Inc.*
- Ujihara, H., and Amano, M. 1994. The latest elevator group-control system. *Mitsubishi Electric Advance* 67:10–12.
- Ujihara, H., and Tsuji, S. 1988. The revolutionary ai-2000 elevator group-control system and the new intelligent option series. *Mitsubishi Electric Advance* 45:58.
- Xu, J., and Feng, T. 2016. Single elevator scheduling problem with complete information: An exact model using mixed integer linear programming. In *American control conference (ACC)*.