Proceedings of the 2005 IEEE
International Conference on Robotics and Automation
Barcelona, Spain, April 2005

Group Elevator Scheduling with Advanced Traffic Information for Normal Operations and Coordinated Emergency Evacuation

Bo Xiong, Peter B. Luh, Shi Chung Chang

Department of Electrical and Computer Engineering
University of Connecticut
Storrs, Connecticut 06269-2157, USA

bo.xiong@uconn.edu, Peter.Luh@uconn.edu, schang@engr.uconn.edu

***Abstract* – In a building, effective operations of transportation systems including elevators, escalators, and stairs are vital. Among them, group elevator scheduling has long been recognized as an important issue for transportation efficiency. The problem, however, is difficult because of the large state space, various traffic profiles, and uncertainties. With the progress in information technology and sensor networks, one potential way is to use advanced traffic information to reduce uncertainties and optimize the performance. How to effectively utilize such information remains an open and challenging issue. This paper presents the optimized scheduling of a group of elevators with advanced traffic information for normal operations and coordinated emergency evacuation. A look-ahead time window is first introduced to model advanced information. Key characteristics of group elevator scheduling are abstracted to establish an innovative formulation. The objective function is transformed into an additive form to facilitate the decomposition of the problem into individual car subproblems. Subproblems are independently solved by using a local search method in conjunction with dynamic programming with a novel definition of stages, states, decisions, and costs to optimize single car dispatching. With surrogate optimization, local search is "good enough" to set multiplier updating directions. Individual cars are then coordinated through the updating of multipliers by using surrogate optimization for near-optimal solutions. Numerical testing results demonstrate that near-optimal solutions are obtained for problems of moderate sizes under selected traffic patterns. The results also show the value of advanced information through testing different window sizes and rescheduling intervals.**

Index Terms - Group elevator scheduling, advanced information, Lagrangian relaxation, dynamic programming, simulation.

1. INTRODUCTION

In a building, effective operations of transportation systems including elevators, escalators, and stairs are vital. Among them, group elevator scheduling has long been recognized as an important issue for transportation efficiency. The problem, however, is difficult because of hybrid system dynamics, combinatorial explosion of the state space, time-varying and uncertain passenger demand, strict operational constraints, and realtime computational requirements for online scheduling. This problem was shown to be NP-hard even for the case without car capacity restrictions (Seckinger, 1999). Recently, elevator systems with destination entry are introduced (Koehler and Ottiger, 2002; and Gale, 2002). In a destination entry system, passengers are required to register their destination floors before they are picked up such that destinations are known in advance. Furthermore, with the progress in information technology, one potential way is to use advanced traffic information from sensor networks and RFID to reduce uncertainties and optimize the performance. Optimized scheduling with advanced information shall lead to better performance as compared to those without advanced information. How to effectively utilize such information remains an open and challenging issue.

In addition, while buildings are evacuated by stairs according to current standards, stairs are inefficient because they become congested, people slow down during the long distance from top floors to the ground, and the elderly and persons with disabilities might not use stairs at all (Koshak, 2004). Elevators have been shown to be potentially invaluable in certain emergencies such as the detection of chemical or biological agents, or fires in neighboring buildings (Hakonen, 2003). Nevertheless, as emergencies are rare events, systems specifically designed for egress cannot be justified based on such merits alone, but must provide an increased value during normal operations as well.

Our problem is to provide a consistent way to model and optimize group elevator scheduling with advanced traffic information for both normal operations and coordinated emergency evacuation. While the goal is for online implementation and realtime scheduling, this paper focuses on how to utilize advanced traffic information to optimize group elevator scheduling and shows the potentials for practical applications. To model advanced information, a look-ahead time window is first introduced where traffic information within the time window is assumed available, and information outside the window is ignored. Cases with different levels of advanced information can be modeled by appropriately adjusting the length of the time window. Key characteristics of group elevator scheduling are abstracted to establish an innovative formulation. Passenger-to-car assignment constraints are established since individual cars are coupled through serving a common pool of passengers. Car capacity constraints and car dynamics are embedded within individual car simulation models. The objective function is flexible within a broad range of passenger-wise and car-wise measures, e.g., average passenger wait time or service time. The formulations for both normal operations and coordinated emergency evacuation are presented in Section 3.

The objective function is first transformed into an additive form to facilitate the decomposition of the problem into individual car subproblems. Individual car subproblems are then obtained through the relaxation of passenger-car assignment constraints. Subproblems are independently solved by using a local search method in conjunction with dynamic programming with a novel definition of stages, states, decisions, and costs to optimize single car dispatching. With surrogate optimization, local search is "good enough" to set multiplier updating directions. Individual cars are then coordinated through the iterative updating of multipliers by using surrogate optimization for near-optimal solutions (Zhao, Luh, and Wang, 1999). The methods for both normal operations and coordinated emergency evacuation are presented in Section 4.

Numerical testing results in Section 5 demonstrate that near-optimal solutions are obtained for problems of moderate sizes under selected traffic patterns. The results also show the value of advanced information through testing different window sizes and rescheduling intervals.

2. Literature Review

Many heuristic rules for online group elevator scheduling have been presented in the literature, such as the highest unanswered floor first, the longest waiting passenger first, equal load among cars, static zoning, and collective control (Barney and Santos, 1985; and Strakosch, 1998). The drawback of *collective control* is bunching, i.e., several elevators move close to each other and compete for the same hall call. In general, heuristic rules are not designed to optimize the performance, and thus cannot consistently yield good scheduling results for various the traffic profiles.

Many optimization methods were also developed for group elevator scheduling. Pure up-peak traffic pattern was analyzed to minimize average waiting time by using queuing theory (Pepyne and Cassandras, 1997, 1998). Dynamic zoning was developed to minimize the round-trip time for up-peak and down-peak traffic by using standard Lagrangian relaxation method (Chan et al, 1997). A reinforcement learning algorithm was developed based on neural networks for all the traffic patterns (Crites and Barto, 1996). While this algorithm is tractable in theory to obtain an optimal policy, it requires 60,000 offline hours of simulated elevator operation to converge for one specific down-peak scenario. Such a training effort is clearly a major drawback. Another method is based on genetic algorithms which represent possible solutions by chromosomes. A fitness function is used to evaluate the performance criterion. Because the inherent combinatorial complexity results in a large search space, it is difficult to obtain a good online solution when time available is limited (Cortes et al, 2003).

The inherent combinatorial complexity and realtime requirements led researchers to develop incremental online scheduling algorithms that consider the assignment of one passenger at a time. When a new passenger arrives, the passenger is assigned to a car whose passengers, as a whole, will be delayed least. A variant of the "Estimated Time of Arrival" (ETA) algorithm was developed based on the evaluation of average waiting time through dynamic programming (Nikovski and Brand, 2003). This incremental scheduling algorithm considers the assignment of only one passenger at a time and is not aimed to optimize the performance over a set of passengers.

Recently, online group elevator scheduling with destination entry has been investigated. The "Estimated Time to Destination" (ETD) algorithm belongs to this category (Smith and Peters, 2002). Once a new passenger arrives, this algorithm estimates the future traffic and evaluates the potential cost that each car would incur if it were to take this passenger. The car with the minimal cost is then assigned to this passenger. This algorithm cannot work well when the actual traffic is different from the estimated traffic because the potential cost is evaluated based on the estimated future traffic.

3. Problem Formulation

3.1 Overview

In this section, a look-ahead time window is first introduced to model advanced information. Passenger-to-car assignment constraints are established since individual cars are coupled through serving a common pool of passengers. Car capacity constraints and car dynamics are embedded within individual car simulation models. The objective function is flexible within a broad range of passenger-wise and car-wise measures, e.g., average passenger wait time or service time.

3.2 System Description

A building with F floors and a group of J elevators is considered. The parameters of elevators are given, including car dynamics and constraints. The car dynamics is based on Nikovski and Brand (2003) as shown in Figure 1 for an up-moving car.

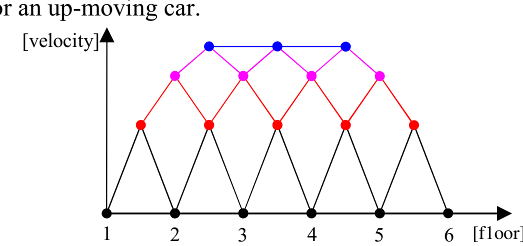

Figure 1. A discrete model of single car dynamics

In this figure, the x-coordinate is the floor index, the y-coordinate is the car velocity, and points represent the last possible locations where the car should start decelerating if it is to stop at a particular floor. The trajectory for a down-moving car is similarly described.

As presented in Section 1, advanced traffic information is assumed available, and this is modeled by a look-ahead time window of length T. Traffic information as specified by the arrival time t_i^a, the arrival floor f_i^a, and the destination floor f_i^d of each passenger i who arrives within the window is assumed given, and the information outside the window is unknown. Different levels of advanced information can thus be modeled by appropriately adjusting T. A rolling horizon scheme is then used in conjunction with time windows, and snapshot problems are re-solved periodically or as needed *without passenger re-assignment* to avoid on-sight confusion. For a snapshot problem, let S_p denote the set of I_p passengers who are leftovers from the previous snapshot problem and have been assigned but not yet delivered to their destination floors; and S_c the set of I_c passengers who are new to the current snapshot problem and are yet to be assigned and delivered. Together there are I passengers ($I = I_p + I_c$) to be delivered to their destination floors.

Constraints to be considered include constraints coupling among cars and individual car constraints. The former includes passenger-to-car assignment constraints stating that each passenger must be assigned to one and only one car, i.e.,

$$\sum_{j=1}^{J} \delta_{ij} = 1, \quad \forall i, \tag{1}$$

where δ_{ij} is a zero-one indexing variable equal to one if passenger i is assigned to car j and zero otherwise. For a snapshot problem, δ_{ij} for all $i \in I_p$ (i.e., passengers left over from the previous snapshot problem) are fixed, and only δ_{ij} for all $i \in I_c$ (i.e., new passengers) are to be optimized. Note that individual cars are coupled since they have to serve a common pool of passengers. Individual car constraints include car capacity constraints:

$$\sum_{i=1}^{I} \zeta_{ijt} \le C_j, \quad \forall j, t, \tag{2}$$

where C_j is the capacity of car j, and ζ_{ijt} is a zero-one indexing variable equal to one if passenger i is in car j at time t and zero otherwise ($\zeta_{ijt} = 1$ iff $t_i^p \le t < t_i^d$). In the above, the pickup time t_i^p and the departure time t_i^d of passenger i depend only on how individual cars are dispatched for a given assignment, and are represented by a dispatching strategy φ:

$$\{t_i^p, t_i^d\} = \varphi(\{t_{i'}^a, f_{i'}^a, f_{i'}^d, \forall i' \in S_j\}),$$
$$\text{where } S_j \equiv \{i' \mid \delta_{i'j} = 1\} \text{ and } i \in S_j. \tag{3}$$

In view that the number of variables $\{\zeta_{ijt}\}$ is very large for a discrete-time model and is infinite for a continuous-time model, and φ could be too complicated to be described, constraints (2) and (3) are not explicitly represented but are embedded in simulation models of individual cars. For single car simulation, travel time between any two floors is derived from Figure 1. Other elevator parameters such as door opening time, door closing time, and loading and unloading times per passenger are assumed given. Higher modeling fidelity can be accommodated as needed.

3.3 Objective Function

In the normal mode the scheduling shall lead to higher passenger satisfaction in terms of certain performance criteria. For simplicity of presentation, the total service time is considered here[1]. For passenger i, the service time T_i is the time interval between the arrival time and the departure time ($T_i \equiv t_i^d - t_i^a$). The objective is to minimize the total service time of I passengers, i.e.,

$$\min_{\{\delta_{ij}, \forall i \in S_c, \forall j\}} J, \qquad \text{with } J \equiv \sum_{i=1}^{I} T_i, \tag{4}$$

subject to (1), (2) and (3).

Performance metrics for the emergency mode is fundamentally different from minimizing the total service time. In the coordinated emergency evacuation, the concern is to evacuate all the passengers as soon as possible. The objective is to minimize evacuation time T_e when the last passenger departs, i.e.,

$$\min_{\{\delta_{ij}, \forall i \in S_c, \forall j\}} J, \qquad \text{with } J \equiv T_e^2, \tag{5}$$

subject to (1), (2) and (3).

The above formulation is applicable to different building configurations and traffic patterns because no specific assumption is made about them. Moreover, this formulation is flexible to incorporate different strategies of single car dispatching since car dynamics are embedded within individual car simulation models.

4. SOLUTION METHODOLOGY

4.1 Overview

As presented in Section 3, the coupling passenger-car assignment constraints (1) are linear inequality constraints, and car capacity constraints (2) and car dynamics (3) are embedded within individual car simulation models. The objective functions in (4) and (5) are therefore first transformed into additive forms to facilitate the decomposition of the problem into individual car subproblems. A decomposition and coordination approach is then developed through the relaxation of passenger-car assignment constraints (1). Subproblems are solved independently by using a local search method in conjunction with dynamic programming (DP) with a novel definition of stages, states, decisions, and costs to optimize single car dispatching. With surrogate optimization, local search is "good enough" to set multiplier updating directions. Individual cars are then coordinated through the iterative updating of multipliers by using surrogate optimization for near-optimal solutions. Duality gap, i.e., the relative difference between the primal cost and the dual cost, is used as a measure of solution optimality. The method for the normal mode is presented first.

4.2 Normal Mode

[1] Our framework is flexible to be applied to minimize the total waiting time, where the waiting time WT_i for passenger i is the time interval between the arrival time and the pickup time ($WT_i \equiv t_i^p - t_i^a$).

4.2.1 Decomposition into individual car subproblems

To decompose the problem (4) into individual car subproblems, the objective function should be additive in terms of individual cars. Therefore the objective function in (4) is rewritten by using (1):

$$J = \sum_{i=1}^{I}(T_i \sum_{j=1}^{J} \delta_{ij}) = \sum_{j=1}^{J} \sum_{i=1}^{I}(\delta_{ij} T_i) . \qquad (6)$$

With this additive form, assignment constraints (1) are relaxed by using nonnegative Lagrange multipliers $\{\lambda_i\}$:

$$L(\lambda,\delta) = \sum_{j=1}^{J} \sum_{i=1}^{I}(\delta_{ij} T_i\ ) + \sum_{i=1}^{I}\lambda_i (1 - \sum_{j=1}^{J} \delta_{ij})$$
$$= \sum_{j=1}^{J} \sum_{i=1}^{I}\left(\delta_{ij} T_i\ - \lambda_i \delta_{ij}\right) + \sum_{i=1}^{I}\lambda_i . \qquad (7)$$

By collecting all the terms related to j from (7), the subproblem for car j is obtained as

$$\min_{\{\delta_{ij}, \forall i \in S_c\}} L_j, \quad \text{with} \quad L_j \equiv \sum_{i=1}^{I}\left(\delta_{ij} T_i\ - \lambda_i \delta_{ij}\right), \qquad (8)$$

subject to capacity constraints (2) and car dynamics (3).

4.2.2 Solving individual car subproblems

Car subproblem (8) is to obtain an optimal passenger selection and an optimal dispatching of selected passengers for a given set of multipliers. In view of the complexity involved, it is difficult to obtain optimal solutions. Nevertheless, based on the surrogate subgradient method, *approximate optimization* of only one or a few subproblems under certain conditions is sufficient to generate a proper direction to update the multipliers (Zhao, Luh, and Wang, 1999). By utilizing this property, our goal is to obtain a better passenger selection with an effective dispatching of the selected passengers by using a local search method. For a given selection of passengers in the local search, dynamic programming is used to optimize the car trajectory and evaluate this passenger selection as described below.

Step 0: Initialize the assignment $\{\delta_{ij}\}$ to car j and set n = 0.

Step 1: Change the assignment of one passenger at a time over I_c passengers arriving within the current time window to generate a set of new assignments, and evaluate their $\{L_j\}$ by using DP to be presented in the next subsection.

Step 2: Identify the assignment that leads to the smallest value of L_j. Use this assignment to update $\{\delta_{ij}\}$, and set n = n + 1.

Step 3: Continue from Step 1 until a pre-specified value of n is reached or no assignment leads to a smaller value of L_j.

4.2.3 Single car dispatching model[2]

A simulation-based dynamic programming method is developed to optimize the dispatching of a car for a given selection of passengers through a novel definition of DP stages, states, decisions, and costs to reduce computational requirements. Our key idea is that for a one-way trip, if the stop floors are given, then the car trajectory is uniquely specified based on Figure 1. With this, a stage is defined to be a one-way trip of the car without changing its direction.

For a stage starting at time t_k, a DP state includes the car position f_j at t_k, the car direction d_j, and the status of the set S_k of passengers that have not yet been delivered to their destination floors at t_k (the status of passenger i includes the arrival time t_i^a, the arrival floor f_i^a, and the destination floor f_i^d). The state is thus represented by

$$X_k = (t_k, f_j, d_j, \{t_i^a, f_i^a, f_i^d \mid \forall i \in S_k\}) . \qquad (9)$$

The decisions for a state include stop floors, the reversal floor where the car changes its direction, and passengers to be delivered in the current stage (limited to those traveling between the stop floors). The decision can thus be represented by $U_k = \{u_i \mid \forall i \in S_k\}$, where u_i is a zero-one decision variable equal to one if passenger i is delivered to the destination floor in stage k and zero otherwise. For passengers already inside car j at t_k, u_i always equals one. For passengers with identical arrival and departure floors, they are picked up according to the first-come-first-serve rule.

Given X_k and U_k, the pickup time t_i^p and the departure time t_i^d of passengers delivered in stage k and the start time t_{k+1} of stage k+1 are obtained through single car simulation. Note that for each passenger, the service time is additive over his/her time delay in each stage (i.e., each one-way trip). For a passenger delivered in stage k, the delay in the stage is $t_i^d - \max(t_k, t_i^a)$. For a passenger not delivered in stage k, the delay in the stage is $t_{k+1} - \max(t_k, t_i^a)$. The delays of passengers can thus be incorporated in the following stage-wise cost:

$$g_k(X_k, U_k) = \sum_{i \in S_k, u_i = 1} \left(t_i^d - \max(t_k, t_i^a)\right)$$
$$+ \sum_{i \in S_k, u_i = 0} \left(t_{k+1} - \max(t_k, t_i^a)\right). \qquad (10)$$

With the above definitions, an optimal trajectory for single car dispatching is obtained by using forward dynamic programming.

4.2.4 Surrogate subgradient method

After a solution for a subproblem (8) is found by using local search in conjunction with DP, a surrogate subgradient is obtained. The component of the surrogate subgradient at the k-th iteration is

$$\tilde{g}_i^k = 1 - \sum_{j=1}^{J} \delta_{ij}^k . \qquad (11)$$

Multipliers are then updated as follows,

$$\lambda_i^{k+1} = \lambda_i^k + s^k \tilde{g}_i^k , \qquad (12)$$

where s^k is the stepsize.

4.3 Emergency Mode

With slight changes, the method presented above can be used to solve the problem in the emergency mode. In the emergency mode, the objective function in (5) can not be easily transformed into an additive form as in (6). Nevertheless, by requiring that the completion time T_{cj}

[2] Other single car dispatching algorithms such as three-passage policy can be incorporated (Gagov et al, 2001).

when the last passenger departs car j be less than or equal to evacuation time T_e, the objective function can be written in an additive form with the introduction of one linear inequality constraint per car:

$$T_{cj} \le T_e, \quad \forall j. \tag{13}$$

The completion time for car j T_{cj} can be easily evaluated by DP by letting the time interval during which car j travels in stage k be the stage-wise cost:

$$g_k(X_k, U_k) = t_{k+1} - t_k. \tag{14}$$

With the above additive objective function, assignment constraints (1) and the additional constraints (13) are relaxed by using nonnegative Lagrange multipliers $\{\lambda_i\}$:

$$L(\lambda,\delta) = T_e^2 + \sum_{i=1}^{I} \lambda_i (1 - \sum_{j=1}^{J} \delta_{ij}) + \sum_{j=1}^{J} \lambda_{I+j}(T_{cj} - T_e)$$

$$= \left(T_e^2 - \sum_{j=1}^{J} \lambda_{I+j} T_e\right) + \sum_{j=1}^{J}\left(\lambda_{I+j} T_{cj} - \sum_{i=1}^{I}(\lambda_i \delta_{ij})\right) + \sum_{i=1}^{I} \lambda_i. \tag{15}$$

The major difference between (15) and (7) is the introduction of an additional subproblem for evacuation time T_e. By collecting all the terms related to j from (15), the subproblem for car j is obtained as

$$\min_{\{\delta_{ij}, \forall i \in S_c\}} L_j, \qquad \text{with} \quad L_j \equiv \lambda_{I+j} T_{cj} - \sum_{i=1}^{I}(\lambda_i \delta_{ij}), \tag{16}$$

subject to (2) and (3). Local search in conjunction with DP is used to solve this subproblem. By collecting all the terms related to T_e from (15), the new subproblem is

$$\min_{\{T_e, \forall j\}} L_{J+1}, \qquad \text{with} \quad L_{J+1} \equiv T_e^2 - \sum_{j=1}^{J} \lambda_{I+j} T_e. \tag{17}$$

In view of its quadratic form, this subproblem can be easily solved. The component of the surrogate subgradient used to update $\{\lambda\}_{I+1}^{I+J}$ at the k-th iteration is

$$\tilde{g}_{I+j}^k = T_{cj}^k - T_e^k. \tag{18}$$

5. NUMERICAL RESULTS

The method presented above has been implemented in C++ on an Intel P4, 2.0GHz PC with 512MB memory[3]. A building with four cars and eight floors is considered in three examples below to demonstrate solution optimality, computation efficiency, and the value of advanced information. The parameters for the simulation are based on data provided by a software tool for elevator simulation (Elevator, 2002). In the following three examples, two window sizes (T = 30sec. and 60sec.) are tested with ten simulation runs. For the 30 second window, the rescheduling interval is 7.5 seconds. For the 60 second window, the interval is 15 seconds. The first example presents the results under various traffic patterns with different levels of advanced information for normal operations. It demonstrates that near-optimal solutions are obtained, and shows that the value of advanced information. The second example studies the performance under a heavy traffic, and shows that not much benefit can be derived from advanced information. The third one demonstrates the solution optimality and the value of advanced traffic information for coordinated emergency evacuation.

[3] For simplicity, not all possible decisions are enumerated in DP by setting an upper bound on the number of transitions for each state.

Example 1

A traffic is generated at a rate of 10 passengers per 30 seconds for a total of 100 passengers for up-peak, down-peak, and two-way patterns individually. A two-way traffic is obtained with arrival times, arrival floors, and destination floors generated from uniform distributions. For up-peak, arrival floors are fixed at the lobby while arrival times and destination floors are uniformly distributed. Similarly, down-peak traffic is generated with a fixed destination floor at the lobby.

The results are summarized in Table 1, where the mean and standard deviation of average service time, the number of multiplier updating iterations, CPU time (averaged by the number of snapshot problems), and duality gap are shown for each traffic pattern with two window sizes. The small values of duality gaps demonstrate that near-optimal solutions are obtained under these traffic patterns (< 10% in duality gap). The results also show that our method is applicable to all the three traffic patterns. Moreover, there is a clear decrease in average service time as the window size and the rescheduling interval increase for each traffic pattern. Nevertheless, the window size should not be too large, because more passengers are considered and more CPU time is thus required to obtain a good scheduling when using a larger window. Also, snapshot problems should be re-solved in a suitable rescheduling interval. If the interval is too short, the performance might be not good since only few passengers are considered to assign for each snapshot problem. In this paper, the traffic within the window is assumed to be known perfectly. This assumption might not hold in practice. If the interval is too long, passengers who are not considered in the current window might have to wait for a long time to get an assignment answer at the next rescheduling point.

Table 1. Results for various patterns with different window sizes

Scenarios		Up-peak		Down-peak		Two-way	
Time window size (s)		30	60	30	60	30	60
Rescheduling interval (s)		7.5	15	7.5	15	7.5	15
Avg. Service Time (s)	Mean	37.3	31.8	36.4	31.2	31.9	29.5
	STDEV	0.53	0.48	0.66	0.52	0.57	0.34
Decrease in Avg. Service Time		14.7%		14.2%		7.5%	
# of multiplier updating		143	176	232	291	183	252
CPU time (s)		19	25	22	28	16	24
Duality gap (%)		4.3	5.4	4.7	5.8	4.1	6.2

Example 2

This example is the same as Example 1 except that it has a heavy two-way traffic, where the arrival rate is 30 passengers per 30 seconds for a total of 100 passengers. Under such heavy traffic, a car is constantly busy in delivering passengers; therefore not much benefit can be derived from advanced information, as demonstrated in Table 2. For each window size, average service time increases compared to the previous data for two-way in Table 1.

Table 2. Results for a heavy traffic with different window sizes

The size of time window (s)		30	60
Rescheduling interval (s)		7.5	15
Avg. Service Time (s)	Mean	59.8	58.7
	STDEV	0.62	0.56
# of multiplier updating		325	464
CPU time (s)		46	65
Duality gap (%)		6.9	8.2

Example 3

In this example, a down-peak traffic pattern is used to model coordinated emergency evacuation by elevators. There are a total of 100 passengers arriving in 5 minutes, at a rate of 10 passengers per 30 seconds.

The results obtained by using our emergency mode algorithm are summarized in Table 3. The small values of duality gaps demonstrate that near-optimal solutions are obtained. In this table, evacuation time is less than 6 minutes (i.e., 360 seconds). There is only one minute between the time the last passenger arrives and the time the last passenger leaves. This means that elevators could be potential resources for evacuation, especially for high-rise buildings, if it is safe enough to use elevators under emergency in the future.

Table 3. Results with different window sizes in the emergency mode

The size of time window (s)		30	60
Rescheduling interval (s)		7.5	15
Evacuation Time (s)	Mean	341.7	330.4
	STDEV	0.99	1.98
# of multiplier updating		352	523
CPU time (s)		36	54
Duality gap (%)		7.4	8.6

6. CONCLUSIONS

Our formulation and method provide a consistent way to model and optimize group elevator scheduling with advanced traffic information for both normal operations and coordinated emergency evacuation. This formulation is applicable to different building configurations and traffic patterns because no specific assumption is made about them. Details of single car dynamics are embedded within individual car simulation models. Individual car subproblems are then obtained through the relaxation of passenger-car assignment constraints. Subproblems are independently solved by using a local search method in conjunction with dynamic programming with a novel definition of stages, states, decisions, and costs to optimize single car dispatching. With surrogate optimization, local search is "good enough" to set multiplier updating directions. Individual cars are then coordinated through the iterative updating of multipliers by using surrogate optimization for near-optimal solutions.

Numerical testing results demonstrate that near-optimal solutions are obtained for problems of moderate sizes under selected traffic patterns. The results also show the value of advanced information through testing different window sizes and rescheduling intervals.

ACKNOWLEDGMENT

This work was supported in part by the National Science Foundation grant DMI-0423607.

REFERENCES

[1] G. Barney and S. Santos, "Elevator Traffic Analysis, Design and Control," England: *IEE, Peter Peregrinus Ltd.* 1985.

[2] W. L. Chan, A. T. P. So, and K. C. Lam, "Dynamic Zoning in Elevator Traffic Control," *Elevator world*, March, 1997, pp. 136-140.

[3] P. Cortes, J. Larraneta, and L. Onieva, "A Genetic Algorithm for Controlling Elevator Group Systems," Artificial Neural Net, Problem Solving Methods, Berlín, Springer-Verlag, 2003, pp. 313-320.

[4] R. Crites and A. Barto, "Improving Elevator Performance Using Reinforcement Learning," *In Advances in Neural Information Processing Systems 8*, Cambridge, MA, MIT Press, 1996, pp. 1017-1023.

[5] Elevator, http://www.elevate.peters-research.com, Peters Research Ltd, 2002

[6] Z. Gagov, Y. C. Cho, and W. H. Kwon, "Improved Concept for Derivation of Velocity Profiles for Elevator Systems," *Proceedings of IEEE International Conference on Robotics & Automation*, Seoul, Korea, May 21-26, 2419-2423, 2001.

[7] J. Gale, "Destination Control and Tower Top Access in Belgium," *Elevator World*, Vol. 10, 2002, pp. 45-49.

[8] H. Hakonen, "Simulation of Building Traffic and Evacuation by Elevators," *Licentiate Thesis*, Department of Engineering Physics and Mathematics, Helsinki University of Technology, 2003.

[9] J. Koshak, "Elevator Evacuation in Emergency Situations," *Proceedings of Workshop on Use of Elevators in Fires and Other Emergencies*, March 2-4, 2004, Atlanta, Georgia.

[10] J. Koehler and D. Ottiger, "An AI-based Approach to Destination Control in Elevators," *AI Magazine*, Vol. 23, Issue 3, Fall 2002, pp. 59-78.

[11] D. Nikovski and M. Brand, "Decision-Theoretic Group Elevator Scheduling," *Proceedings of the 13th International Conference on Automated Planning and Scheduling ICAPS03*, Trento, Italy. June 9-13, 2003.

[12] D. L. Pepyne and C.G. Cassandras, "Optimal Dispatching Control for Elevator Systems During Up-peak Traffic," IEEE Transaction on Control Systems Technology, Vol. 5, No. 6, November, 1997.

[13] D. L. Pepyne and C.G. Cassandras, "Design and Implementation of an Adaptive Dispatching Controller for Elevator Systems in Uppeak Traffic," *IEEE Transactions on Control Systems Technology*, Vol. 6, No. 5, pp. 635-649, September, 1998.

[14] B. Seckinger, "Synthesis of Elevator Controls Based on Constraint-based Search," Master's thesis, Albert Ludwig University Freiberg, German, 1999.

[15] R. Smith and R. Peters, "ETD Algorithm with Destination Dispatch and Booster Options," *Elevator world*, July, 2002, pp. 136-142.

[16] G. R. Strakosch, "Vertical Transportation: Elevators and Escalators," New York, *John Wiley & Sons, Inc.* 1998.

[17] X. Zhao, P. B. Luh, and J. Wang, "The Surrogate Gradient Algorithm for Lagrangian Relaxation Method," *Journal of Optimization Theory and Applications*, Vol. 100, No. 3, March 1999, pp. 699-712.