SICE Annual Conference 2007
Sept. 17-20, 2007, Kagawa University, Japan

A Branch and Bound Method to the Continuous Time Model Elevator System with Full Information*

Zhen Shen[1] and Qianchuan Zhao[2]

[1, 2] Center for Intelligent and Networked Systems (CFINS)
Department of Automation, Tsinghua University, Beijing 100084, China
([1]Tel: +86-10-62792491, E-mail: zhenshenthu@gmail.com)
([2]Tel: +86-10-62783612, E-mail: zhaoqc@tsinghua.edu.cn)

Abstract: We give a new Branch and Bound method for the scheduling of the group elevator system with full information. Full information means that not only the parameters of the elevator systems but also the arrival time, origins and destinations of all the passengers who are to be served are known beforehand. The results of the full information problem can guide the future elevator design. Our method can handle the continuous time event and is based on the concept of "trip" which means the movement of the car without changing the direction and with at least one passenger being served.

Keywords: Elevator, Branch and Bound, Trip

1. INTRODUCTION

Nowadays the elevator scheduling system usually only utilizes the current and historical information. The future is predicted based on the current and historical information but the future information is not used. Extensive research has been done in this aspect [1, 2]. In recent years, some attentions have been paid to the research on the full information problem assuming future information is available [3-5]. The importance of studying full information problem lies in the fact that the optimal solution of the full information problem is "helpful to estimate the effectiveness of the utilization of look-ahead information" and "also useful in valuating the performance of the existing rules for elevator operation", as pointed out in [5]. If we know the full information can help improve the performance a lot, in the future the elevator should be designed to take use of the future information. Moreover, it is reasonable to assume that in the future everything can be connected to the Internet and the passenger may be able to "tell" the elevator system at what time s/he arrives, just as we reserve flights and hotels. And the near future information can be collected by sensors, web cameras, etc. In [3] the authors show the near future information is valuable. In [4] and [5] the full information problem is solved by the Branch-and-Bound (B&B), with the basic assumption that the time is discrete and the events only happen at the discrete time points.

In this paper, we consider a group elevator system with M identical cars. The object is to minimize the average service time (waiting time plus transition time). We make the following assumptions. The car cannot move in the direction that is opposite to the direction the passengers in it demand. The passenger cannot de-board the car half way or change the car half way. Once a passenger boards a car, s/he must be delivered to their destination floor without passing by the destination floor and then returning to it. The speed of the car is not affected by the number of passengers in it. Inspired by the original work of [4,5], we develop a Branch-and-Bound method for a continuous time model which is more realistic to describe elevator systems. Our method is based on the concept of "trip" which means the movement of the car without changing the direction and with at least one passenger being served. We will describe this concept in detail in next section. This concept provides us the advantage of decision space reduction.

In our test, we will consider two traffic patterns, up-peak pattern and two-way pattern. "up-peak" means all the passengers appear at the lobby. "two-way" pattern is the most general pattern, in which the passenger can come from any floor and go to any other floor.

2. THE CONCEPT OF TRIP AND ITS PROPERTIES

A trip is a service course by one car for a given non-empty set of passengers satisfying conditions: 1) all passengers in the set have the same travel direction; 2) the number of passengers in the car is no greater than the car capacity at any time. If passengers in a trip are going up (down), we will call the trip an up (down) going trip. The car movement corresponding to the service of a trip is from one floor to another (there can be stops between them) without changing the direction and with at least one passenger being served. The car only stops at floors at which either some passenger arrives or departs. At the same floor, before loading passengers the car will unload the passengers. For up going trips, passengers arriving at lower floors will be loaded earlier; passengers with lower destination floors will be unloaded earlier. It is symmetric for the down going trips. Between two trips, there may be a car movement with no passenger from the end floor of the first trip to the starting floor of the second trip. Such car movement will be called 'trip link'.

Overall, to optimize the car service for a given list of passengers, it is sufficient to decide how the passengers

*This work was supported by NSFC Grant No. 60274011, NSFC Grant No. 60574067, and the NCET (No. NCET-04-0094) program of China and 111 the- 111-International-Collaboration-Project -of-China.-
.

PR0001/07/0000-0327 ¥400 © 2007 SICE

are assigned into different trips.

We have the following observations on the properties of trips which will be used in the development of our Branch and Bound Algorithm. For *n* passengers, there can be at most *n* trips. And,

[***Theorem*** **1**] Optimal performance can be achieved if we add the following first come first service (FCFS) rule to the assignment of passengers to trips: in a trip, at the same origin floor, the passengers arriving earlier must be loaded earlier.

Proof: The main idea of proving this theorem is to construct an FCFS sequence which is not worse than a non-FCFS one.

We discuss the case of two passengers first. We assume that the 1st passenger comes earlier than the 2nd passenger. For the two passengers, if the loading sequence is non-FCFS, when the 2nd passenger is boarding the car, the 1st passenger must be waiting. At this time point, the two passengers have both arrived and both want to board the car. It is feasible to exchange the loading sequence of the two passengers, and let the 1st passenger board the car at the time when the 2nd passenger boards the car in the non-FCFS case, and let the 2nd passenger board the car at the time when the 1st passenger boards the car in the non-FCFS case, we can obtain a loading sequence which is FCFS and has the same performance with the non-FCFS sequence. So, we do not need consider the case of non-FCFS loading sequence. Moreover, we compare the 'ordinary' FCFS sequence with the 'made up' FCFS sequence which is obtained from the non-FCFS mentioned above. Here 'ordinary' means the 1st passengers boards the car as soon as the car is ready to load passengers. For the 'made up' FCFS case, the 1st passenger cannot board the car before the 2nd passenger comes, even the car has been ready. Obviously, the performance of the 'made up' FCFS is not better tan 'ordinary' FCFS. And we know the non-FCFS case cannot be better than FCFS case.

Above is the case of 2 passengers. The results can be easily extended to the case in which the number of passengers is larger than 2 by decomposing the problem into a series of two passenger comparisons. For example, three passengers and the loading sequence is 3rd, 1st, 2nd, we can first compare it with 1st, 3rd, 2nd, then we know the 3rd, 1st, 2nd cannot be better than 1st, 3rd , 2nd. Then we compare 1st, 3rd, 2nd with 1st, 2nd, 3rd, we know 1st, 2nd, 3rd is better or at least has the equal performance with 1st, 3rd, 2nd. So we know that 3rd, 1st, 2nd cannot be better than 1st, 2nd, 3rd. No matter how many passengers there are, for any sequence, we can change the order of two neighboring passengers and then within finite times the sequence can be changed into a FCFS one.

So, we know that for the passengers who come from the same floor and are to be served by the same trip, we only need to consider the FCFS case. ■

3. ALGORITHM DESCRIPTION

Before presenting our Branch and Bound algorithm, we would like to state an additional property on the order of passenger service that is useful to reduce the decision space.

[***Theorem*** **2**] For the passengers who have the same origin and destination, those who arrive earlier must be loaded earlier (not necessarily in a same trip).

Proof: Similar to the proof of Theorem 1. The basic idea is to generate a feasible solution by ex change a pair of non-FCFS passengers having the s ame origin and destination pair without sacrificing performance. ■

We firstly introduce the main idea and then we give the description of the algorithm.

Let us first explain that once the set of passengers assigned to a trip is known, the movement of this trip can be determined. We take up going trip as an example, and the down going trip is similar. The starting floor of this trip is the one that is the lowest of the floors at which the passengers assigned to this trip arrive. The end floor is the highest of the floors at which the passengers want to de-board. Between the starting and end, for this trip the car will load or unload the passengers according to the demand of the passengers. And when there are more than one passenger who want to board the car at the same floor, the boarding sequence must be FCFS (Theorem 1). Just like what happens in real life, before loading passengers the car will unload passengers if there is any who has arrived at their destination.

When a car serves passengers, the typical moving of the car is that it moves up, and moves down, and up again, and so on. When serving passengers and not changing direction, it is a 'trip'. Otherwise the movement is a 'trip link'. The basic idea of our method is to decompose the whole service for the passengers to the trips. Every service sequence is made up of trips. Trip links are used to connect consecutive trips if necessary. And if we assign all the passengers to trips and the assignment is reasonable, the trips form a reasonable service sequence for the passengers.

By the trip concept, we do not need to analyze the details of boarding event and de-boarding events. Boarding and de-boarding of the same passenger must occur in one trip. If the trip concept is not introduced, we have to consider many details such as the boarding events, the de-boarding events, the directions of moving, the current positions of the cars, etc. If we do not use the "trip" concept and choose to deal with the problem on event basis, checking the feasibility will be not easy. For a coming event, if it is a boarding event, we should check direction constraint first. If the car is empty, the car can be sent to load the passenger. If the car is not empty, the car can only move in the direction the passengers in the car demand and can only load the passengers who want to in the same direction. If the car is going up/down, only the passengers whose arrival floors are not lower/higher than the current floor at which the car is can be loaded. Also capacity constraint should be considered. If the coming event is a de-boarding event, passengers in the car should be checked. Only the passengers whose destination is nearest to the current floor at which the car is can be allowed to de-board. Otherwise there is at least one passenger who goes past his/her destination without de-boarding, which conflicts the constraint. However, by “trip” concept, only direction and capacity constraints need to be checked. Passengers who want to go in the same direction can be assigned to a same trip, as long as the capacity constraint is satisfied. Modeling and describing the problem can be much easier.

Among the different assignments, there must one with the best performance, Branch and Bound method is used to find it.

We take an example of single-car up-peak. For two pa ssengers (denoted by 1st, 2nd according to the arrival tim e), the assignment to trips can be as follows.

Table 1 A simple example to illustrate the basic idea of Branch and Bound. 2 passengers. Up-peak. Single car.

Index	1	2	3	4
Sequence	1st, 2nd	1st /2nd	2nd , 1st	2nd / 1st

‘1st, 2nd’ means the two passengers are served in one trip, and the car loads the 1st passenger first, then the 2nd passenger. 1st/2nd means the car serve the two passengers by two trips. The car loads the 1st passenger and then serves him/her, and return to the lobby. Then the car loads the 2nd passenger, and serves him/her. For other service sequence, the explanations are similar. The third one which is ‘2nd, 1st’cannot be optimal according to Theorem 1. And among the other cases, there must be one with the best performance.

For the small example, we in fact enumerate all the p ossibility of assignments. But by Branch and Bound met hod, we can use a clever way to consider all the possibil ity of assignments.

Now we give the description of the Branch and Bound algorithm.

[**Algorithm**] Branch and Bound algorithm.

[Input] A set of passengers to be served. For each passenger, arrival time, origin floor and destination floor are given.

[Output] A list of trips for each car with all passengers assigned to one and only one trip of a certain car.

Step 1. Initialization. Set the lists of trips for all cars as empty and call this state the root node. Create a set U for unassigned passengers and set U as the input set of passengers. Set the best so far performance as $u=+\infty$.

Step 2. Branching. Choose one passenger p from U according to the conditions of Theorem 1 and Theorem 2. There are two ways to create a new branch. That is either

1) add a new trip containing only the passenger p to the end of the list of trips of car i; or

2) add the passenger p to the last trip of a car i as long as the adding of p does not cause the trip to be invalid (travel direction and car capacity constraints are not violated).

Step 3. Update the set U by removing passenger p. If U is empty, all new branches are called leaf nodes. Go to Step 5. If U is not empty, go to Step 4.

Step 4. Evaluate the lower bound b for each new branch as follows and prune all branches with $b \geq u$.

$$b = (S1 + S2 + S3)/n,$$

where n is the total number of the passengers. $S1$ is the total service time of the assigned passengers. $S2$ is the total “ideal service time” of the unassigned passengers. “ideal service time” of a unassigned passenger is the service time of the passenger by a dedicated car ready at the his/her origin floor. $S3$ is the time delay of serving unassigned passengers caused by serving the assigned passengers (part of waiting of the unassigned passengers). We explain $S3$ in detail as fallows. Generally speaking, the not assigned passenger cannot get a car to serve him/her as soon as he/she arrives which is the case of ‘ideal service time’. He/she must wait for the car to come, i.e. a delay for serving the passenger may exist. $S3$ is used to describe the influence of the possible delay. For a sub-problem (a branch), we assume the time for the car to finish the last trip is $t1$, and the car is at floor f. We note the fact that the passengers not assigned cannot get service before $t1$. For a passenger not assigned, we assume the time for the car to go directly from floor f to the origin of the passenger is $t2$, if $t1+t2 >$ the arrival time of the passenger, a delay exists, otherwise, the delay is not considered. $S3$ is the all the delay time for the passengers not assigned. We name $S3$ as ‘possible delay time’. $S2 + S3$ is the lower bound of the service time of the not assigned passengers.

Step 5. Evaluate the performance of the each new leaf node and update u with the best performance value of these nodes. Prune all branches with $b \geq u$.

Note: In the branching step (Step 2), we always explore the branch with the smallest value b first.

An example of the B&B algorithm for three up going passengers served by one car is shown in Fig. 1. The lobby is the origin floor of the three. Passenger 2 and 3 are with the same destination and passenger 2 arrives earlier. Each row in a rectangle represents a trip. For example, the box $\boxed{\begin{array}{c}1\\2,3\end{array}}$ means there are two trips: the first trip contains only Passenger 1 and the second trip contains Passenger 2 and 3. The number in the figure is the index of the passenger. The index is given according to the arrival time.

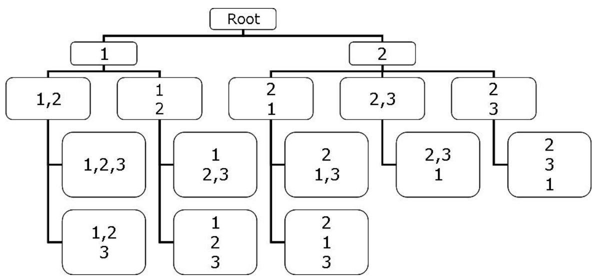

Figure 1 Illustration of B&B algorithm for 3 passengers.

4. EXPERIMENTS

We assume that the arrival of passengers is according to Poisson distribution, and the origins and destinations are uniformly distributed. At the beginning, all cars stop at the first floor. Door open time and door close time are ignored. We set the intensity as λ= 0.167/s, and the number of floors H = 10, the car speed is assumed constant, the time for the car to go across one floor T = 5s, the time for a passenger to board or de-board the car is assumed the same: τ=1s, the capacity of a car C = 8 passengers. Every testing case uses 50 samples.

Table 2 Results of experiments.

Type	M	n	G/s	SDG/s	T_{CAL}/min	SDT/min
UP	1	13	59.0	9.4	3.9	9.9
		14	60.8	10.0	14.7	25.8
	2	7	31.4	5.9	0.7	0.8
		8	30.8	5.8	12.7	18.5
TW	1	12	58.0	7.5	5.1	9.5
		13	61.1	6.6	19.2	36.6
	2	6	34.8	8.9	0.3	0.2
		7	34.5	6.9	8.0	7.1

In the table. UP = Up-Peak (passengers come from the lobby), TW = Two-way (the most general pattern). M = Number of cars. n= number of passengers. G= the mean of the shortest average service time of the problems. TCAL = time needed for calculation. SDG = standard deviation of the shortest average service time of the problems. SDT = standard deviation of the calculation time.

As an example, we use our method to evaluate the method in [6] which uses Lagrange Relaxation to assign passengers to cars and use an ordinary elevator dispatching algorithm to assign passengers to trips based on two rules. The 1st rule is the passenger who comes earlier has priority to be served earlier. The 2nd rule is that the car picks up passengers who would like to go in the same direction if the capacity constraint is not conflicted. For more details of the method, please refer to [6].

Table 3 Results given by Lagrange Relaxation method. The meanings of symbols are the same as Table 2.

Type	M	n	G/s	SDG/s	T_{CAL}/s	SDT/s
UP	1	13	84.1	18.8	0.00	0.00
		14	87.2	19.4	0.00	0.01
	2	7	40.7	11.6	6.1	0.9
		8	40.0	10.3	5.7	0.4
TW	1	12	72.8	11.3	0.01	0.01
		13	72.2	8.8	0.01	0.01
	2	6	42.6	10.7	6.3	0.2
		7	42.6	7.8	6.6	0.4

From Table 2 and Table 3, we know the performance of the optimal solution on average is about 70%~80% of the performance given by Lagrange Relaxation method. But the results are for small scale problems.

5. CONCLUSIONS

We give a new Branch and Bound method which can handle problems with continuous time events. The average service time is taken as the performance to be optimized. To show the perofmance of our algorithm, two types of arrival patterns, up-peak traffic and two-way traffic are tested. Next step. we will focuse on improving the calculation of lower bound to improve the efficiency of the Branch and Bound method .

ACKNOWLEDGMENT

The authors would like to thank Prof. Yu-Chi Ho for the helpful comments to improve the quality of a previous version of this paper, thank Dr. Jia Qingshan for the helpful discussions, and thank Mr. Sun Jin for providing the codes for Lagrange Relaxation methods.

REFERENCES

[1] G. C. Barney, Elevator Traffic Handbook, *Spon Press*, London, 2003.

[2] Pepyne, D.L., and Cassandras, C.G., Optimal Dispatching Control for Elevator Systems During Uppeak Traffic, *IEEE Transactions on Control Systems Technology*, Vol. 5, No. 6, pp. 629-643, 1997.

[3] B. Xiong, P. B. Luh, and S. C. Chang: Group Elevator Scheduling with Advanced Traffic Information for Normal Operations and Coordinated Emergency Evacuation, Proc. of the 2005 IEEE Int. Conf. on Robotics and Automation, pp. 1419 – 1424, April 2005

[4] Inamoto, t., Tamaki, H., Murao, H., and Kitamura, S., An Application of Branch-and-Bound Method to Deterministic Optimization Model of Elevator Operation Problems, *Proceedings of the 41st SICE Annual Conference*, Vol. 2, pp. 987-992, 2002.

[5] Tsutomu Inamoto, Hisashi Tamaki, Hajime Murao, Shinzo Kitamura, Deterministic Optimization Model of Elevator Operation Problems and An Application of Branch-and-Bound Method, IEEJ Trans. EIS, Vol. 123, No. 7, 2003.

[6] J. Sun, Q. C. Zhao, P. B. Luh, and M. J. Atalla, "Estimation of Optimal Elevator Scheduling Performance," *IEEE ICRA 2006*, pp. 1078-1083.