ELEVATOR GROUP SUPERVISORY CONTROL USING FUZZY LOGIC

Ming Ho and Brent Robertson
Department of Electrical and Computer Engineering
University of Waterloo, Waterloo, Ontario, N2L 3G1
email: ming@crg4.uwaterloo.ca or broberts@kingcong.uwaterloo.ca

Abstract

This paper describes the use of fuzzy logic in the scheduling of a group of elevators. A new "orbital" paradigm is introduced that offers a simple unified framework under which scheduling can be completed. The principle is to choose a configuration of elevator placements upon an "orbital circle" that best fits a number of fuzzy constraints. To our knowledge, fuzzy logic has traditionally been used to support the more established techniques, such as expert knowledge-base approaches. Our system uses fuzzy logic alone to make non-predetermined control decisions. The "fuzzy orbital" scheduler was implemented and simulation results illustrate the viability of this paradigm.

1.0 Introduction

Controlling of a group of elevators in a multistory building is a complex problem. Many different approaches to solving this problem have been investigated. First the background involving the scheduling of elevators will be discussed. This is followed by the design criteria, an outline of the "orbital" paradigm, a description of the rule base, and an overview of our architecture. Simulation results are presented and finally some conclusions are suggested.

2.0 Background

Modern elevator control involves the scheduling of elevators in order to attain better performance. The measure of performance has changed in recent years. At first it was measured in terms of System Response Time and System Service Time. The former is the time from the elevator request (hall call) until its arrival and the latter is the time per floor from the destination request to the arrival at the destination floor as defined by Pang in [1]. However, recently users have requested others goals in addition to the above. Tobita et al. add the goal of minimizing the number of passengers in a car in [2].

Since the advent of microprocessor controllers in the late 1970's, numerous methods for the real-time optimum scheduling of elevators have been devised. In [3], a multidisciplinary AI-approach is described which constantly monitors and saves the traffic data, which it uses to predict future traffic and to select the most appropriate dispatch algorithm. It checks its own performance and can intelligently attempt to improve efficiency. The architecture used is similar to that in [1], namely the blackboard architecture.

In [1], an implementation of a blackboard in Prolog is described. Its goal is to minimize the waiting time, riding time, and the distance travelled by the elevators. It makes use of heuristic knowledge in order to process large amounts of data which can be incomplete and unrelated. A simple simulation is shown which may be useful for comparison.

In [4] Etessami and Hura describe a method for modelling an elevator system, the Abstract Petri Net (APN). It reports how the attributes of a real-time system can be defined into rules, which in turn can be represented by APNs. It claims flexibility, modularity, and verifiability.

Hikita and Komaya describe a new elevator group control system which improves the user's control over multiple control goals in [7]. It is noted that users have recently been interested in performance measures other than simply the waiting time; they include riding time and crowding. While the other papers describe the dispatch problem only, this paper also talks about the support system which allows a users to set the weighting on the multiple (and possibly conflicting) objectives. The reported system is actually a hybrid: it uses fuzzy logic as its user-interface to set the minimization parameters and a knowledge base to select the dispatch strategy.

Also of note is that in [6] Bates states that there is empirical evidence which shows that although elevators may run efficiently when filled to capacity, this efficiency is never realized because passengers are unwilling to enter into elevators with many people in them.

Mathematical models have been used to simulate the performance of elevators. From the work of Barney, [6] constant speed is assumed and round trip time becomes a linear function of the number of floors to travel and stops to make. In this analysis, to estimate the capacity, it is often assumed that the passenger arrive according to a Poisson pdf. It was verified by Alexandris in [6] using empirical data, that Poisson is a reasonable approximation. Barney's simulations have also shown that control strategies for sectoring are significant for down-peak and interfloor traffic.

The phenomenon of "bunching" where all cars are going in the same direction in phase, was discussed by Allaart in [6]. As a result of bunching, waiting times increase so that performance is not much better than one large capacity elevator.

In [5], Ujihara and Tsuji describe a fuzzy based system similar to that in [7]. Note that an advantage of fuzzy logic controllers for elevator group scheduling is that no assumption needs to be made about the distribution of the arrival of passengers. This approach allows the application of heuristics like expert systems and blackboards, yet with less rules.

3.0 Design Criteria

Using the information gained from the previous section, goals are defined, as well as assumptions which affect the operation.

The following goals were, in order of importance, used to design our fuzzy controller:

- Minimize the waiting time for individuals at a given floor
- Minimize the time an individual spends in an elevator
- Minimize the crowding in elevators

The common goal of minimizing the distance travelled by the elevators was not explicitly dealt with at this time.

To allow the comparison of results achieved by Pang, the same basic elevator properties were assumed:

- Up and own buttons are available at each floor, except for the top floor and the ground floor, which only have down and up buttons respectively.
- A hall call may arrive at any instance from any floor in any direction (limited by the buttons). The car call is also assumed to be recorded immediately after the passenger enters into the car.
- An elevator will not reverse direction if there is a passenger inside.
- The capacity of the elevators is six people, and when capacity is met, it may bypass otherwise acceptable hall calls.
- Each elevator travels at a constant speed of .5 floors per second.
- Serving a floor requires 4 seconds to accomplish. During this time a person may simply walk onto the elevator.
- An elevator must not bypass a car call floor.

Hence, given these specification, rules can be constructed in the effort of minimizing the stated criteria.

4.0 Orbital Paradigm

The Orbital paradigm was conceived in order to provide a framework for calculating various attributes of an elevator system configuration. A configuration is a set of decisions for each elevator to target "up" or "down" In order to understand this paradigm, we must first understand the problem it seeks to model.

In an elevator system, the scheduler can make decisions when an elevator is free due to the constraints of expected elevator behaviour. For example, if a rider enters an elevator and requests to go down, the elevator should not go up one floor to pick up another rider simply because this behaviour is not acceptable to users. Given this situation, if all elevators are running (i.e. they have riders) then the scheduler loses its positioning control. It can still allow or disallow an elevator to pick up riders along its path, but it cannot send it to the most under-serviced areas until that elevator becomes free. Thus we need a model where we can be confident that should we lose control of an elevator, that car will have a good likelihood of encountering riders (hall calls) along its way and will not compete with other cars.

At the time when a scheduler needs to make an assignment decision, it only has information at that time. Traditionally a car is assigned a floor once it becomes free. However this strategy does not allow re-assignment to be done based on new information. On the other hand, re-assignment can be dangerous as it is possible to oscillate its targeting between multiple floors. Thus we need a model which will not change targets without a very good reason. But how would we know what is a "good reason"? This is where we introduce the Orbital model and fuzzy logic to evaluate an elevator configuration.

The Orbital paradigm uses an ellipsoid to represent a movement track where cars can run only in one direction, like a circular 1-track railway. Elevators are placed on this orbit depending on its current position and direction. Hall calls are also placed on the track in the same way. *Figure 1* shows how each elevator in the physical model is mapped onto the orbital track. A free elevator can be on one side of the orbital or the other depending on which direction is chosen. *Figure 2* shows two different configurations based on the chosen directions of the elevators.

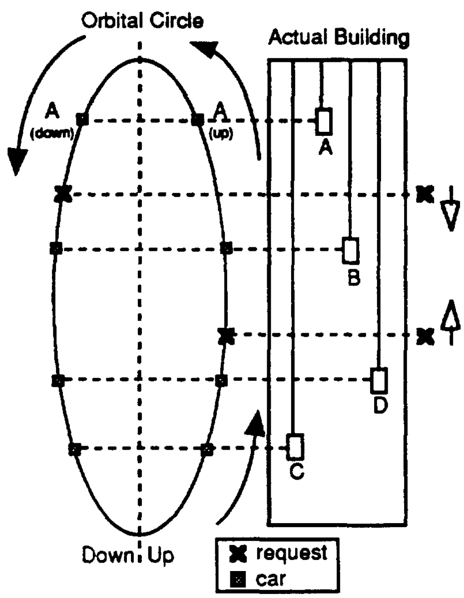

Figure 1

We can now make some measure of the "appropriateness" of a configuration. At this point we only assign a direction to each free elevator. The target can then be found by tracing the path in front of the elevator to the next hall call that has not already been targeted. The following measures have been utilized:

1) Separation of cars: this is calculated as the sum of the squares of the distance between adjacent cars along the orbital. This shows the degree to which the elevators are "bunched up", an undesirable trait.

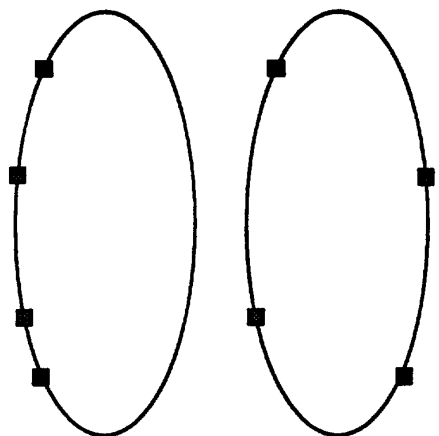

Figure 2

2) Average distance to target: this is simply the average of the absolute distance between a car and its target. This measures roughly how much time is required to reach the target.

3) Average wait: this is the average elapsed waiting time of users on the targeted floor.

There are other uses for this model:

1) Number of calls between cars: this would handle the case where hall calls are concentrated in a small area and we do not want the car targeting that area to be overloaded.

2) History information (old hall calls or a time-dependent set of hall calls such as for lunch-hour) can be superimposed on the orbital to add weighting to existing measures.

We now have some measures of traits that are desirable in the elevator configuration, but how will we combine them into a single measure? What is important in one situation may not be in another. This is where fuzzy logic is used. Each trait can be assigned a linguistic variable and the recommendations of the fuzzy rules for the various measures can be combined by defuzzification. We then have a value that can be compared with other configurations. Note that a configuration only needs a direction to be assigned. Therefore the number of possible configurations is relatively low. This would not be the case if we must consider the assignment of each elevator to each possible hall call.

Once a configuration is chosen, the actual targets for each free elevator can then be chosen based on its position on the orbital. The elevators then move towards their target. As they get closer to their target, the appropriateness of that configuration gets stronger due to the distance-to-target measure, so only serious changes in the environment would change the configuration.

5.0 Fuzzy Rule Base

A fuzzy logic controller allows the implementation of linguistic rules such as those that constitute common sense. Hence "If temperature is high then set fan to high" can be implemented. Fuzzification is the transferring the non-fuzzy input (crisp) input value to the fuzzy (linguistic) value. Through the inferencing, the input values are related to the output values for each rule. The final output (crisp) value is usually determined by taking the centre of area of the fuzzy output functions for all the rules. This is the fuzzification process.

To facilitate the debugging and to emphasize the independence of the rules, the fuzzy rules are subdivided into groups. These groups are the movement group, the stop group, and the depart group. Each group fires independently of the other, hence simplifying the optimization of each group. Another characteristic of the rules is that they are all simple, so that no AND or OR operations are needed.

5.1 Group I: Movement Rules

These rules control the assigning of free elevators to a particular direction. These rules are fired every time frame when there are free elevators. These rules are fired once for each possible configuration. Hence they return an "appropriateness" of a configuration. These fuzzy rules take as input, the sum of the square of the separation between elevators (using the "orbital" paradigm discussed above), the waiting time, and the sum of the square of the distances between elevators and hall calls. The separation is desired to be large (i.e. not bunched together), while the distance between elevators and hall calls is desired to be small (i.e. not far for the elevators to travel). In addition the waiting time is desired to be small.

5.2 Group II: Stop Rules

These rules decide if an elevator containing passengers stops at a floor with a hall call or continues to its destination. These rules only fire when the elevator reaches the floor with a hall call in the current direction of travel, there is not a car call for this floor, and the elevator is not full. The rules determine whether to service this hall call or not. These fuzzy rules take as input: how long the hall call has been outstanding; how full the elevator is; and how long the passengers have been riding the elevator. For example:

`If HallWaitTime is Long then Likelihood [of stopping] is high`

5.3 Group III: Depart Rules

The depart rules decide whether a departing elevator should leave a floor or not. These rules only fire when the elevator is ready to close the doors and leave the floor. They may delay the departure of an elevator to prevent the "bunching" problem discussed in 2.0. These fuzzy rules take as input, the last time that an elevator left this particular floor in the particular direction, how full the elevator is, and how long people have been riding in the elevator. For example, at 9:00 am up peak, if an elevator leaves the bottom floor (up) one second before, it is desirable to wait before departing so that these two elevators don't "bunch".

For example, the fuzzy function corresponding to long people have been riding in the elevator appears as:

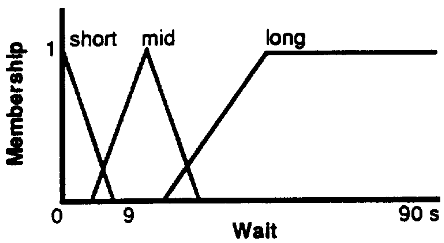

Figure 3

6.0 Results

A simulation was completed using the same setup used by Pang in [1]. Thus, the building for the simulation was 15 stories tall with 4 elevators as stated in the design criteria section. As starting conditions, the elevators are idle at the third, fourth, fifth, and tenth floors respectively. The when and where passengers arrive into the system as well as their destination is also required. This information is read into the simulation program via an ASCII simulation file.

	Blackboard System	Fuzzy Logic System
maximum SRT	8.000 s	10.000 s
average SRT	3.667 s	3.500 s
maximum SST	3.778 s	3.778 s
average SST	2.632 s	2.690 s

Figure 4

Figure 4 compares the results of the fuzzy logic simulation with the results by Pang in [1]. Except for the maximum System Response Time, all metrics for the fuzzy controller are within 5% of the results from the blackboard system. Hence the simulation results for the fuzzy controller are comparable with the simulation results from the blackboard approach.

7.0 Conclusions

Given that the simulation results from the fuzzy system were comparable to the results from the blackboard and given that the architecture is less complex than the blackboard, further simulations of the fuzzy controller are definitely recommended.

The opportunity exists to add more fuzzy rules for temporal considerations (9:00 am up peak, 5:00 pm down peak) which will improve performance. Adding fuzzy rules is much easier than adding rules to the blackboard system due to the potential for conflicting rules if proper verification is not completed and due to the system complexity. Furthermore, optimization of the fuzzy functions could improve performance of the controller.

Thus, fuzzy logic based controllers present a simple, unified framework, clearly viable for the scheduling of elevators.

8.0 References

[1] G. K. H. Pang, "Elevator Scheduling System Using the Blackboard Architecture", *IEE Proceedings*, vol. 138, no. 4, p. 337-346, 1991.

[2] T. Tobita, A. Fujino, H. Inaba, K. Yoneda, and T. Ueshima, "An Elevator Characterized Group Supervisory Control System", *Proceedings IECON '91*, vol. 3, p. 1972-1976, 1991.

[3] N. Kameli and K. Thangavelu, "Intelligent Elevator Dispatch Systems", *AI Expert*, p. 32-37, Sept. 1989.

[4] F. S. Etessami and G. S. Hura, "Abstract Petri Net Based Approach to Problem Solving in Real Time Applications", *TENCON '89*, p. 234-239, 1989.

[5] H. Ujihata and S. Tsuji, "The Revolutionary AI-2100 Elevator-Group Control System and New Intelligent Option Series", *Mitsubishi Electric Advance*, p. 5-8, Dec. 1988.

[6] G. C. Barney, *Elevator Technology*, Ellis Horwood Limited, West Sussex, England, 1986.

[7] S. Hikita and K. Komaya, "A New Elevator Group-Supervisory Control System Using Fuzzy Rule-Base", *Transactions of the Society of Instrumentation and Control Engineers*, vol. 25, no. 1, p. 99-104, 1989.