Proceedings of the 1984
Winter Simulation Conference
S. Sheppard, U. Pooch, D. Pegden (eds.)

THE ROLE OF COMPUTER SIMULATION IN THE DEVELOPMENT OF A NEW ELEVATOR PRODUCT

Bruce A. Powell
Westinghouse R&D Center
Pittsburgh, PA 15235

ABSTRACT

This paper discusses a new elevator dispatching strategy called Advanced Traffic Management (or ATM). A substantial computer simulation effort proved to be a major influence in the development of this new product for introduction to the marketplace. A simplified version of ATM will be presented, and the simulation modeling process will be outlined. The main contributions of the simulation effort were in initial product verification and in design improvements resulting in increased U.S. patent protection.

INTRODUCTION AND SUMMARY

A critical transportation system in a high rise building is the elevator system. An integral part of each bank of elevators is a group dispatching strategy which governs the movement of the elevators as they respond to passenger demands. This paper will discuss a new elevator dispatching strategy in which the results of a computer simulation study proved to be a major influence in the development of the new product for introduction to the marketplace.

The performance of the system as perceived by the passengers is inversely related to the time spent waiting for an elevator. The dispatching strategy is only one factor affecting waiting times. Waiting time is also a complex stochastic function of such variables as the number of elevators (or cars), car speed, number of floors served, and passenger traffic patterns. Dispatching strategies have evolved from the simple attendant or manual control to more sophisticated microprocessor based algorithms. Obviously, some strategies will produce better service (i.e., shorter waiting times) than others. A good strategy is a key selling point for an elevator supplier.

In the spring of 1984, Westinghouse introduced a new dispatching strategy called Adaptive Traffic Management or ATM. In its basic form, ATM is elegantly simple yet remarkably efficient. During the development of the algorithms which comprise ATM, an extensive computer simulation study was made to evaluate its effectiveness and to identify portions of the algorithm which needed improvement. Initial simulation runs revealed that the dispatching strategy as originally designed performed exceptionally well. The simulation also predicted, however, that an occasional passenger would wait for a time longer than acceptable. To correct this possibility, a significant improvement was made in the algorithm, and the simulation results of the updated strategy were used as evidence in the issuing of a U.S. patent. With this modification, the ATM dispatching strategy proved to be competitive in the marketplace.

This paper will discuss elevator dispatching strategies, beginning with a brief history. Then, the specific algorithm of the ATM system will be discussed, within company proprietary limitations. Finally, the simulation model and results of runs will be described.

SOME ELEVATOR TERMINOLOGY

Imagine an eight-story building with three elevators. When a person on Floor 5, say, pushes the button to call an elevator to go down, the dispatching strategy determines which of the three elevators is to answer the call. The word car will be used interchangably with elevator. Three other terms require definition.

Corridor Call. When a person wishes to use an elevator, he presses a button (UP or DOWN) in the corridor. The person is said to have registered a corridor call.

Car Call. When a passenger steps into the elevator, he presses a button to tell the car where he wants to go. The car is said to have a car call registered for the destination floor.

Waiting Time. The time elapsing between the registering of a corridor call and the cancelling of the call is the waiting time. The call is cancelled when the car reaches the passenger's floor of origin.

Note that the waiting time statistic is for each corridor call -- not for each passenger. If more than one passenger boards a car at a floor, the waiting time is the time that the first person (i.e., the person originating the call to the elevator) had to wait. Individual passenger waiting time is another meaningful measure of system efficiency, and it is obviously correlated to corridor call waiting. However, because passenger waiting time is difficult to measure (in a data collecting sense), the elevator industry has standardized on the waiting time associated with corridor calls.

DISPATCHING STRATEGIES FOR ELEVATORS

It is clear that the dispatching strategy must consider the complete state of the system in deciding on a car to answer any call. This system state would include information on the cars (current position, direction, load, and committed stops) and as-yet-unanswered corridor calls (i.e., floor and direction of the call). It is also clear that some ways of dispatching the cars will be better than others. While there are many ways to evaluate an elevator system, the word "better" will refer to the length of time that a randomly generated corridor call must wait for an elevator.

One of the earliest means of dispatching elevators was by attendant control. Each elevator was manned by an operator who would decide where the stops were to be made. This form of dispatching system has obvious shortcomings, not the least of which is the lack of coordination with other elevators serving the same set of floors.

The development of the automatic door led to the ability to use automated push button systems. Passengers would operate the system by pressing buttons, much as they do now, in the corridors and in the cars. Dispatching systems began with simple systems in which priority was given to car calls, thus restricting each elevator to serve one corridor call at a time. Collective control, in which calls (both car calls and corridor calls) are answered in floor sequence was an advancement in these early push button systems. Collective control is commonly used today in small installations such as two elevators serving a five floor building.

Two elevators serving a common set of floors and operating independently are not efficient. The next logical development in dispatching systems was in the area of group control, in which attempts were made to assign cars and calls by means of a specific set of rules. The rules -- which Operations Researchers call algorithms -- vary from very simple rules to extremely complicated ones. For example, one rule that makes sense is to assign a newly registered corridor call to the "closest" car. To illustrate how the rules can get messy in a hurry, consider the above rule with respect to the attribute "closest". Does this mean nearest in distance, or does it mean the elevator that would reach the floor in the least amount of time? Also, does this rule allow for a car to reverse direction to answer a corridor call? If so, can the car reverse its direction with passengers on it? Some of the common features included in many group control systems include:

- ... main floor preference
- ... recognition of evening rush hour (termed downpeak)
- ... recognition of morning rush hour (termed uppeak)
- ... special treatment of basements or parking garages
- ... priority treatment of calls that have been waiting an unacceptable amount of time

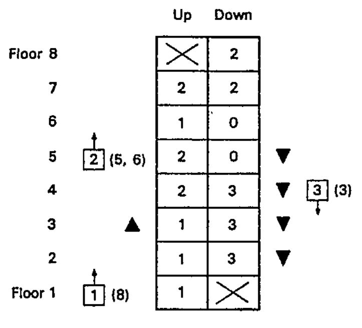
Much of the implementation of group control rules was done by hard-wired logic. Relays would be interconnected in such a manner that rules on special treatment for various passenger traffic conditions could be invoked at the appropriate times. These systems, many of which are in highrise buildings today, have the disadvantage of being nearly permanent. That is, changes in the rules made necessary by changes in the way the occupants used the elevators could be made only at the cost of rewiring. With the advent of microprocessors, such permanence has been removed. Essentially, the group of elevators can be controlled by software. Algorithms can be tailored for individual buildings, and can be made to be dynamic in time. An important side benefit of microprocessor controlled elevator systems is the capability to collect data on passenger demand patterns and on car movement patterns.

The new Westinghouse dispatching system, ATM, is a microprocessor based system. The next section of this paper will describe the rudiments of the algorithm.

THE ATM STRATEGY SIMPLIFIED

We will now discuss the new Westinghouse microprocessor based dispatching strategy called ATM. We have simplified some of the details in order to protect the proprietary nature of the product. However, the framework of the strategy is described.

Consider a hypothetical eight story building being served by three elevators. Figure 1 is a snapshot of the elevator system at a particular moment during the day. Car 1 (shown as a square with an arrow) is oriented in the up-direction at floor 1 and has a car call at floor 8. (Remember that a car call is registered by a passenger who indicates his destination by pushing a button inside the car.) Car 2 is at floor 5 and has car calls at floors 5 and 6. Car 3 is at floor 4 in the down direction with a car call at floor 3. The symbols ▲ and ▼ represent up and down corridor calls at the indicated floors.



		Up	Down	
Floor 8		╳	2	
7		2	2	
6		1	0	
5	2 (5, 6)	2	0	▼
4		2	3	▼ 3 (3)
3	▲	1	3	▼
2		1	3	▼
Floor 1	1 (8)	1	╳	

Figure 1

Snapshot of eight-floor elevator System

Indicated by the numbers in the boxes in the "UP" and "DOWN" columns are cell assignments. A cell is a <direction, floor> pair to which a car may be assigned. For example, the down cell at floor 7 (see Figure 1) is currently assigned to car 2.

Consequently, if a down corridor call is registered at floor 7, car 2 will be required to answer the call by virtue of its cell assignment. The DOWN cells for floors 6 and 5 have not been assigned to any car, and it will be shown how this perfectly normal situation can arise.

One other term is important--that of the advanced position of an elevator. The snapshot of Figure 1 might have been taken when any or all cars are moving. Since we only track the cars with reference to integer floor positions, it makes sense to keep track of the car according to the "nearest floor at which the car can stop". Therefore, had car 3 been moving, floor 4 would be its advanced position. If a car is stopped at a floor, its advanced position is equal to its actual location.

Notice that car 2 is assigned to the UP cell at floor 4. If an up corridor call were to be registered at floor 4 in the next instant, we can see that it would not be efficient to assign the call to car 2. Indeed, because the cars are moving a great deal of the time, it does not make sense to have static cell assignments. The key to the simplified ATM dispatching strategy is the reassignment of the cells at regular, short intervals.

We must therefore develop the rules by which the cells will be reassigned. The general idea is to try to spread out the cars as evenly as possible and equalize the number of committed calls. Each car will be responsible for a dynamic zone of floors in "front" of the car. The even distribution of cells and calls gives rise to two basic rules:

Rule 1. Divide the cells evenly among the cars.

Rule 2. Establish a maximum number of corridor calls assigned to each car.

There are many algorithmic ways that these rules can be implemented. We will arbitrarily say that, for the hypothetical building shown in Figure 1, Rule 1 dictates that we specify that the maximum number of cells assigned to each car shall be five. Rule 2 dictates that no car may be assigned to more than two cells at which there are corridor calls.

As the first step in the reassignment of cells, we invoke the following:

Rule 3. Remove all cell assignments except those at which there are corridor calls.

Figure 2 shows the result of the implementation of Rule 3. Also, since car 3 now violates Rule 2 (it has too many corridor calls), the DOWN cell at floor 2 must be removed.

The second step in the reassignment of cells is to determine the order in which the cars will be considered for cells. A measure of "busyness" is arbitrarily determined, one way being to order the cars in increasing level of commitment as measured by the total number of committed stops. The car that is least busy will be given first choice of assignments.

Step 3 is to assign cells to the cars that have car calls. Car 1 has a car call at floor 8; therefore, the DOWN cell for floor 8 is given to car 1. Car 2 is given the UP cells at floors 5 and 6. Figure 3 shows the status of the reassignments after step 3.

The final step is to assign the other cells. Again, cars are considered in order of busyness. Figure 4 shows the final cell assignments. For this step, car 1 is given the UP cells at floors 1, 2, and 4, in addition to the cells previously assigned to it. The cell quota is met (i.e., at most five cells) for car 1. The algorithm then moves on to car 2. The assignments are made under the quotas established by Rules 1 and 2.

It can be seen that two DOWN cells (floors 5 and 2) have not been assigned. The reason is that cars 1 and 2 have met their cell quotas, and car 3 has met its call quota. The corridor calls at floors 2 and 5 will be assigned to some car during a subsequent reassignment.

This ATM dispatching strategy allows only the assigned car to respond to a corridor call. A logical question would be "How good is this strategy?" The next section will discuss the simulation model used to test and improve the basic strategy.

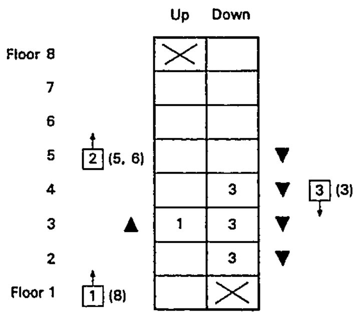


Figure 2

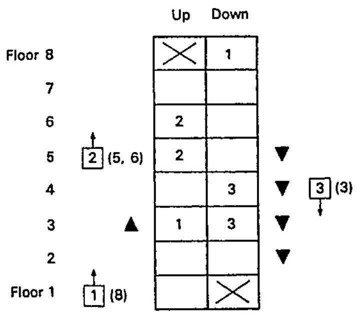


Figure 3

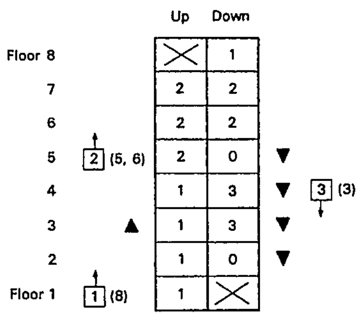


Figure 4

Final cell assignment

THE SIMULATION PROCESS

A major problem with papers dealing with applications is that the details of the model are more interesting to the person doing the modeling than to the reader. A reasonable approach, then, is to describe the problem and the general modeling approach, with many of the details omitted. The reader has already learned what the problem is and can skip to the last section to find out the results. For completeness, though, we will briefly discuss the simulation model of the ATM dispatching strategy.

In overview, the model was developed in the SIMULA language on the Sperry 1100/81 mainframe computer. The source program contains some 2500 lines of code, and requires roughly ten seconds of CPU time to make a simulation run. During the development of ATM, a great many runs were made, and several improvements were made to the algorithms.

There were two main objectives to the simulation effort. First, a general assessment of the effectiveness of ATM was needed. While in theory, the algorithms looked good, a more systematic test under simulated building conditions was desired. Second, the matter of cycle time -- the time between reassignment of cells -- was critical. Naturally, the algorithm ran in real-time, and the microprocessor required a finite amount of time to do all the calculations. It was obviously desirable that the microprocessor be able to keep up with the system. Since system effectiveness depends very much on cycle time, the simulation was asked the following question: "Given that a small cycle time is best, how long can the cycle time become before waiting times become unacceptable?" An initial cycle time of 1/2 second was thought to be computationally feasible. (As an aside, this problem has somewhat been minimized with the development of faster and faster microprocessors. However, in 1974, the issue of computability was a critical concern.)

The task of the simulation model is to act as an omniscient observer. For every passenger to the simulated elevator system, the model should be able to construct a record of important events, from which system statistics -- most importantly, waiting time -- can be determined. For each passenger, the following information was constructed, in addition to a sequence number to trace him:

(1) Floor of Origin
(2) Floor of Destination
(3) Time of Arrival
(4) Time of call registration (usually the same as, but sometimes earlier than, (3) when another passenger was first to push the corridor button)
(5) Time of Call Cancellation
(6) Time that he got off the elevator at destination

From the above numerical information, we can compute various measures of system performance. For example, the waiting time of a call is given by (5)-(4). The waiting time for the passenger is (5)-(3). The passenger's total time in the system (i.e., waiting plus travel) is given by (6)-(3).

The simulation model involves three separate stages: (i) passenger traffic generation, (ii) simulation of the dispatching strategy responding to the generated passengers, and (iii) statistical analysis of the simulation output -- mainly waiting times. Each of these stages will be treated in turn.

The model for passenger traffic generation is explained in reference (4). Basically, passenger traffic is modeled as a Poisson process, with traffic originating from each floor according to particular traffic patterns. During the morning rush hour, most of the passengers originate from the main lobby; during the evening, most of the passengers are going from upper floors to the lobby. Other important and distinct time periods are lunch hour and off peak. An origin-destination matrix is created consistent with the above assumptions. Then, individual passengers are sampled with standard Monte Carlo methods. The ARRIVER procedure schedules these passenger arrivals.

The part of the simulation dealing with the dispatching strategy contains two main features. The first is the algorithmic computation of the cell assignments according to the ATM rules. This is done in the procedure REASSIGN, which is called every δ seconds, where δ is an input parameter. The second feature of the simulation model handles the routine events of elevator operation, such as loading and unloading at a floor, interrupting the system when a new corridor call is registered, and rescheduling a stop in response to a newly-entered corridor call.

The statistical analysis portion of the model merely computes waiting times and other measures of elevator system performance. While individual passenger waiting times can be measured, most conclusions are made on the basis of corridor call waiting time. A distribution of the waiting times is calculated. The key measures of performance are average waiting time, maximum waiting time, and the percentage of calls waiting more than one minute.

A sequence of passengers is a random sample from a complicated stochastic process. From run to run, there are likely to be subtle changes in the input stream that will cause the waiting time statistics to vary. Hence, concern must be paid to replication and to run length. In most cases, a 15-minute period of simulation time is sufficient to get a good reading on waiting time statistics. In addition to making replications of the same traffic conditions, carefully planned experiments were made to simulate the various traffic conditions that the elevators would encounter. This plan recognizes the fact that while ATM (or any other dispatching strategy) works well during one traffic period (say downpeak), it might not work as well during another period (say heavy noon-time traffic).

RESULTS OF SIMULATION

After extensive runs of the simulation model for realistic building parameters and expected passenger traffic patterns, the overall result was that Westinghouse had a good, marketable product in its ATM dispatching strategy. Specific results were three:

1. Passenger waiting times were surprisingly short.

2. A cycle time for cell reassignments of up to 2.0 seconds provided acceptable service.

3. Potential weaknesses in the strategy were identified. Major modifications were made that resulted in an improved product and also in additional U.S. patents.

Each of these results will be discussed in detail below.

Waiting time is a measurable but subjective quantity. At issue is "how long (or short) a time constitutes an acceptable waiting period?" As developers of dispatching strategies, we can recognize the obvious: waiting times will vary during a typical day. As users of the elevators, we expect outstanding service at mid-morning when the traffic is light. We also expect longer waits during rush hour. Extensive use of the SIMULA model for the ATM strategy was made, in which all reasonable traffic patterns, intensities of passenger arrivals, and building configurations were tested. While a full reporting of experimental results is not appropriate for this paper, some approximate figures will contribute to the reader's appreciation for the problem.

Specifically, for a 16-story building served by six elevators, an average waiting time in the low 20-second range was observed under heavy noon-time traffic. Traffic data from another particular building (nine stories, four elevators) during heavy noon-time period produced average waits in the 12-second range. In general, we concluded that waiting times were remarkably short, considering the simplicity of the ATM dispatching strategy. Indeed, performance was comparable to that rendered by many top-of-the-line systems currently in use.

The matter of cycle time for recalculating cell assignments was a critical concern. It is clear that the faster the microprocessor can carry out the algorithmic calculations, the better the distribution of cells and calls will be. Conversely, with a long interval between cell assignments, the elevators will be responding to cells which have been allocated with "old" information. Extensive experimentation with cycle time revealed that cycle times of five seconds or more resulted in very poor service. Cycle times of two seconds provided excellent results, and smaller cycle times did not significantly reduce the waiting times.

Finally, in running some of the simulation experiments, particular sequences of events identified potential weaknesses with the strategy as originally conceived. The following paragraph provides a brief description of one of these situations.

One of the features of ATM, as can be seen from the earlier description, is that, when a corridor call is registered, the call will usually be answered by the specific elevator then assigned to the cell. In other words, a car "locks in on" a call. (This isn't always the case, due to the application of the call quotas, but it is usually true.) This feature is implemented by Rule 3 as outlined in the body of this paper. However, consider the following situation. Suppose the successive cell assignments are made at times t(n) and t(n+1). For ease of explanation, assume that a car is travelling down with advanced position Floor 4 at time t(n). Figure 5 will serve to illustrate this discussion. This car is assigned to the down cells at floors 3 and 4. At time t(n)+ ε, the position changes to Floor 3. At time t(n)+2 ε, with position at Floor 3, a down corridor call is registered at Floor 4. Now, at time t(n+1), the cells are to be reassigned, and this car has a corridor call at Floor 4, for which it "owns" the down cell. But the car has already

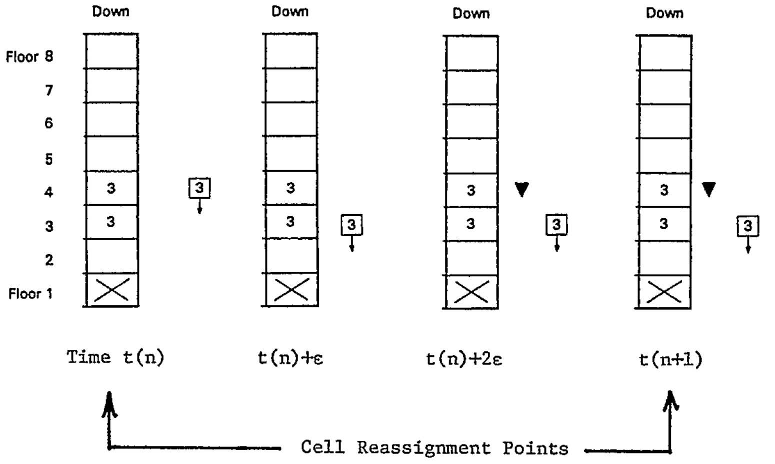


Figure 5. ATM Wipeout Modification

passed Floor 4. Rule 3 would have the car keep its assignment to the down cell at Floor 4, and the call would be answered after the car made a complete round trip of the building. With hindsight, we would say that, since the call at Floor 4 was "behind" the car, the assignment for that cell should be wiped out and the cell reassigned to a closer elevator. This particular case was identified as a direct result of the simulation effort.

Several potential problem areas were identified and corrected, the modifications were turned into patents, and the improved strategy became the ATM system which is being sold today. Numerical results of simulation runs are cited as evidence of the value of such modifications in patent documents. In fact, a U.S. patent was issued in the name of the simulation modeler (reference (5)).

To reiterate, the discussion on the details of the ATM strategy have been written carefully to preserve the proprietary nature of the product. While the skeleton of the strategy appears here, there are many important features which cannot be disclosed. The significant point to be made, however, is that the computer simulation study played a major role in the development of this elevator product.

REFERENCES

1. Barney, G.C., and S.M. dos Santos, Lift Traffic Analysis Design And Control, Peter Peregrinus Ltd., England, 1977, 317 pages.
2. Powell, B.A., "Elevator Banking For High Rise Buildings", Transportation Science, Vol. 9, No. 3, pp. 200-210, August 1975.
3. Powell, B.A., G.T. Hummert, and T.D. Moser, "Real-time Simulation of Elevators", Proc. 1978 Winter Simulation Conference, Miami Beach, Fla., Dec. 4-6, 1978, pp. 393-402.
4. Powell, B.A., "Mathematical Modeling of Elevator Systems", Chapter 2 in the book Case Studies in Mathematical Modeling, W.E. Boyce (ed.), Pitman Publishing Inc., Marshfield, Mass., 1981, pp. 18-53.
5. Powell, B.A., "Elevator System", U.S. Patent 4,046,228, Sept. 6, 1977.