An Intelligent Real-Time Lift Scheduling System

by

Muna Hamdi

A doctoral thesis

Submitted in partial fulfilment of the requirements

for the award of

Doctor of Philosophy

of

Loughborough University.

Department of Electronic and Electrical Engineering

November 1999

عَلَّمَ الإِنسَانَ مَا لَمْ يَعْلَمْ

For

my parents

may their souls rest in peace

Synopsis

In modern high-rise buildings, a suitable control algorithm has to be chosen so that lifts can respond to passenger requests in such a way as to transport them quickly and efficiently to their destinations.

The aim of the current work is to assess new scheduling approaches and intelligent monitoring techniques in order to aid the design of new lift systems and to improve the performance of existing installations To achieve this, the project has been divided into three major parts Firstly, a model of passenger movements has been developed from an analysis of data gathered from installed lift systems, thereby allowing the realistic simulation of landing calls, car calls and door opening times. Secondly, a lift simulator has been produced to allow the modular comparison of alternative scheduling and monitoring approaches and to provide an accurate model of lift dynamics. Thirdly, a new intelligent lift scheduling system has been implemented.

Keywords: intelligent scheduling, real-time scheduling, heuristic search, A* search, lifts, elevators, prediction

Acknowledgement

I would like to express my debt of gratitude and appreciation to my mentor Dr. David Mulvaney for his continuous support and help while conducting this work.

The support and assistance of all my family and friends is greatly acknowledged. I would also like to thank all those who helped me during the progress of this work in the Department of Electronic and Electrical Engineering.

Publications

The following is a list of author's publications that have been produced during the time this research was carried.

- Hamdi M , Mulvaney D. J. and Sillitoe I. P W., *An intelligent real time lift scheduling system,* 14th Workshop of the UK Planning and Scheduling Special Interest Group, Coventry, UK, November, 1995.
- Hamdi M. and Mulvaney D. J., *Visual interactive lift simulator*, Elevcon 96, Barcelona, Spain, October, 1996.
- Hamdi M. and Mulvaney D. J., *Intelligent real time scheduling and its application to lift systems*, IASTED International Conference on Artificial Intelligence and Soft Computing, Banff, Canada, July, 1997
- Hamdi M and Mulvaney D. J., *Simulating of lift systems and modelling of passenger movements*, The International Journal of Elevator Engineers, Vol. 2, 1998, pp. 1-18.

Content

CHAPTER 3

CHAPTER 4

CHAPTER 5

CHAPTER 6

CHAPTER 9

Abbreviations

AI	Artificial Intelligence
A*	Astar heuristic search
ACA	Adaptive call allocation
CC	Contract Capacity
DO	Door Open
DC	Door Closed
DD	Door Delay
DS	Dynamic Sectoring
eta	estimated time of arrival
ETQ	Express Traffic Quantum
FS0	Fixed Sectoring common-sector system
	Fixed Sectoring priority timed system
GA	Genetic Algorithms
HC	Handling Capacity
I	Interval until next passenger arrival
INT	the average time between successive lift car arrivals at the main floor with cars loaded to any level [Barney and dos Santos 1985]
IDA*	Iterative Deepening A*
L	number of Lifts
LC	Landing Call

NN	Neural Networks
NP_complete	Nondeterministic Polynomial. The word complete is used in the sense of "most extreme" [Russell and Norvig 1995].
PASS	passengers
RL	Reinforcement Learning
RSR	Relative System Response
RTA*	Real-Time A*
RTT	Round Trip Time
T	Time interval
UPHC	Up Peak Handling Capacity
UPINT	Up Peak Interval
WT	Waiting Time
PCC	Predicted Car Calls

Symbols

λ	mean arrival rate
☺	a lift
*	a car call
[→←]	door closing
[←→]	door opening
↑	up
↓	down
_	dwell: lift's door open with no direction
f	cost evaluation function.
g	cost of getting from the initial state to the current state of the node.
h	estimate of the additional cost of getting from the current state of the node to the goal state.
LC1→LC2	answering landing call 2 after landing call 1

Chapter 1

Introduction

Scientific research is dedicated mainly to enhance the quality of life, by providing methods and technologies to assist humans in their efforts to achieve better performance. One of the main growing areas of research and development is the use of Artificial Intelligence (AI) in the field of automation to allow the manipulation of large amounts of data, to improve decision making, to improve the quality of product, to enhance cost effectiveness, time efficiency and to improve reliability. AI can be defined as the use of computational power to analyse, classify, process, learn, reason and search for optimum or near optimum solutions to complex problems. Intelligent methods allow us to predict future events based on the knowledge gained from the experience of past events.

High rise lift scheduling systems are expected to respond to passenger demands in such a way as to transport them quickly and efficiently to their destinations without excessive waiting In a competitive market, all the major manufacturers have investigated the use of AI methods to improve the performance of their lift schedulers. The lift industry is often considered to be conservative due to the requirements of meeting stringent safety standards and the commercial need for reliability [Ovaska 1991]. There is also resistance to change due to the prohibitive cost overheads involved in the testing, installation and maintenance of new scheduling systems. Consequently, although manufacturers have studied many intelligent scheduling methods, few have been put into production

Modern high rise buildings often have more than one group of lifts (typically comprising 4 or 6 lifts), with each group being controlled separately. A passenger requests a lift by pressing the lift-calling button at the landing, thereby registering a landing call In the lift, the passenger presses the destination floor button to register a car call The lift traffic intensity and pattern depends on the building function and will vary during the day. For example in an office building occupied by one or more companies, employees arriving for work at the terminal floor will generally produce a considerable demand at that floor resulting in the 'up peak' traffic. Following this period and up to lunch time there is generally relatively uniform movement between all floors, forming the 'interfloor' traffic. At lunch time there will normally be more demand to travel to the restaurant floor (if any) and the terminal floor At the end of the day people leave the building from all floors in the direction of the terminal floor forming the 'down peak' traffic. During the day lift traffic patterns can suddenly change due to variety of reasons (such as conference meetings), which can result in an unpredictable sudden change in demand to or from one or more floors, further details are given in (Section 2 1).

There are a number of characteristics of lift scheduling which make it a *hard* problem [Mulvaney and Sillitoe 1992].

- The general solution to optimal lift scheduling falls into the category of problems mathematicians term *NP-complete*♦ , namely that the time taken to solve the problem will increase exponentially with problem complexity This can be illustrated by considering a building with n lifts and p landing calls. As there are n^P possible scheduling solutions, an extra landing call will give rise to a substantial increase in the number of schedules to be considered. Consequently, it cannot be guaranteed that an optimal solution will be found within the critical time required to produce a schedule. It is the task of the lift control system to select which of the assignments is the most suitable in order to achieve specific objectives (such as minimising waiting time)
- There is no sufficiently complete model of the relationships between the characteristics of lift systems (such as landing calls and car calls) to allow either a closed analytical solution or an approach which would reduce the complexity of the optimal solution.
- System safety and reliability is of high importance and should be taken into consideration when designing new lift systems.

♦ *Belonging to the class of nondeterministic polynomial problems [Russell and Norvig 1995]*

- The scheduler must be sufficiently flexible to allow tailoring to suit individual requirement such as building design and customer demands
- Incomplete knowledge is supplied to the scheduler such as when the next landing call is going to occur, how many passengers are behind a call, how long it will take them to get into a lift and which destination each is going to choose. The quality of the data presented to the scheduler is of upmost importance to its performance

One of the most difficult problems in lift system design is accounting for the unpredictability of the traffic patterns. The missing knowledge makes the task of providing optimum real time decisions by the scheduler impossible in practice. In general, the more we know about the building and lift environment and the more accurate the data supplied to the scheduler, the better will be its performance An intelligent lift scheduler uses the assistance of a prediction system which analyses the daily traffic data of the lift system and compares it with previous analysed data in order to identify general trends in the traffic density and distribution. These expected values can then be used to aid the production of scheduling decisions At times of peak demands in a group of lifts, an intelligent co-ordinated scheduling lift system is necessary to minimise the passenger waiting time and to prevent long queues developing [Hamdi and Mulvaney 1998].

It is important to identify areas where the application of AI methods are suitable and to avoid making the system more complex. Five principal areas have been identified as follows

- **Lift system status and knowledge base** containing general information and heuristics relating to lift systems and lift scheduling, such as the relative priorities of different calls, lift dynamics, the rules for strategy selection and the stability of solutions.
- **Prediction** of lift system input data such as landing calls, car calls, loading and unloading times.
- **Control policy selection** The use of input data (and predicted input data) in the selection of an appropriate policy, such as normal operation or up peak.

- **Strategy selection** The use of client-specified objectives in the selection of an appropriate control strategy. For example, choosing minimising energy consumption as an optimisation factor during low intensity traffic, further details are given in Section 3.2.
- **Scheduling** Adopting a new scheduling technique may allow easier configuration to suit customer requirements. The real-time scheduler may need other control policies to complement its work such as parking and up peak policies

The first four of these areas are the modules which constitute the **intelligent monitoring** of the lift system. Intelligent monitoring is discussed in Chapters 3, 4 and 8.

1.1 Aim and objectives

The present work aims to improve the performance of existing lift system installations by reducing the average waiting time of passengers. This is to be achieved by developing a system which provides a suitable combination of intelligent scheduling and monitoring The intelligent lift scheduling system should exhibit the following characteristics

- Safety and reliability have the highest priority in the lift system and must be taken into account when choosing a lift control system.
- The assignment of lifts to landing calls must be given in real-time. For the building example used in this work the scheduling cycle must be no more than 250 ms, Chapters 2 and 6.
- The system must be adaptable to suit different building configurations without incurring a prohibitive cost overhead.
- The solution must be able to adapt to different traffic patterns and intensities.
- The system must make the most of the available knowledge by providing suitably high quality data (actual and predicted) to the real-time scheduler, for example lift system status, predicted car calls, operation rules and constraints to improve the system performance.

- The need for human intervention should be kept to a minimum. For example a lift operator or designer may need to set or specify predetermined rules such as 'if more than four car calls are registered at the terminal floor then switch up peak on'.

In the current work, the Kodak lift control system using the Express Traffic Quantum (ETQ) scheduler developed by Express Evans Ltd. is taken as a case study, since actual data recorded from the lift system have been made available This allows a direct comparison between the existing lift system and the new intelligent lift scheduler developed in this work.

Heuristic search is a popular method for real-time scheduling (Chapter 3), where the heuristics are used to limit the search and control its execution time. In a lift scheduler it is very important to be able to predict accurately how long a search will take and, if incomplete, whether the partial schedule would be adequate. Generally, the deeper the search the higher the quality of the schedule produced, but the longer is the computation time. Dynamic scheduling algorithms perform sequencing and resource allocation on-line in the hope of making use of comprehensive and up-to-date information about the tasks and the environment. Reducing the complexity of the search may diminish the quality of the schedules produced in terms of guaranteeing deadline compliance [Hamidzadeh and Atif 1996] Dynamic real-time lift schedulers must be able to achieve a desired quality and typically this is the minimisation of average passenger waiting time. Barney and dos Santos [1985] recommended that a passenger in an office building should not be expected to wait longer than 15s To provide a suitable compromise, one technique is to produce a partial schedule for high priority tasks first, in which case it is important to consider how best to allocate processing time between the tasks [Hamidzadeh and Atif 1996]. Mouaddib and Gallone [1996] used a progressive scheduling technique to generate quickly an initial imprecise answer, which is then successively refined.

To meet the aim and considering the above characteristics, the objectives of the current work can now be stated.

- To analyse the data available from the Kodak building, and to understand both the lift system and passenger traffic behaviours in order to develop passenger arrival and movement models to be used for traffic simulation and prediction.
- To implement a lift simulator to test the feasibility of the technique and to assess its performance when working with the existing lift control system. The lift simulator is also important for the understanding of the lift system and traffic flow.
- To introduce and develop a real-time intelligent lift scheduler using a heuristic technique.
- To implement an intelligent real-time lift control system which consists of two main parts, namely a real-time scheduler and a monitoring system. It is a single strategy system since only one optimisation factor is considered and that is minimising the average waiting time. This was chosen according to the requirement of Express Evans Ltd, since minimising passenger waiting time is considered by the company as the highest priority at all times.

1.2 Structure of the thesis

This thesis is the documentation of the research carried out to design and implements an intelligent lift scheduling system It is divided into nine chapters including the current chapter. In the following, the structure of the thesis and the arrangement of its chapters in the order they appear along with the issue of discussion in each chapter are briefly described.

Chapter 2 alternative conventional lift scheduling algorithms are reviewed including the ETQ lift scheduler. The chapter discusses the important characteristics of a lift control system, the drawbacks of conventional lift control systems and the points that need to be addressed when designing an intelligent lift control system.

Chapter 3 gives an overview of scheduling and relevant AI methods. The need for intelligent real-time lift scheduling systems is discussed and examples of a number of intelligent lift systems are introduced, including their advantages and disadvantages.

Chapter 4 the intelligent real-time lift scheduling system implemented in this work is introduced, including the main parts of the system consisting of a lift system simulator, a model of passenger movements, the real-time scheduler and the lift monitoring system. A discussion of the use of real-time heuristic search is given and suitable search techniques for the lift scheduling problem are suggested.

Chapter 5 presents the lift data analysis performed on the actual data extracted from the Kodak building, including an analysis of the distribution of passenger arrival A model of building traffic and passenger movement is developed, implemented and tested using the lift simulator.

In **Chapter 6** the implementation of the lift simulator is described and test results using the simulated ETQ scheduler are presented

In **Chapter 7** the intelligent real-time lift scheduler design, development and testing is described. The chapter includes the definition of the lift scheduling problem and an analysis of the heuristic search problem and its solution.

Chapter 8 gives the implementation of the lift monitoring system which is combined with the real-time scheduler to produce an intelligent lift scheduling system which is tested using the lift simulator. The intelligent lift scheduling system performance is compared with that of the ETQ scheduling system. Several simulation runs results are shown and new adaptive parking and up peak detection policies are introduced and implemented.

Chapter 9 reviews and discusses the contribution of the work presented in this thesis, and identifies areas for future work.

Chapter 2

Description of Lift systems

A supervisory lift control system consists of two main parts, namely the low-level control such as door movement and the high-level control such as the lift scheduler which may be supported by an off-line traffic monitoring and prediction system The work of this thesis concentrates on the high-level control part of the lift control system. This chapter gives a definition of lift system traffic and peak traffic periods in Section 2.1, an overview of the lift system in Section 2.2 and the conventional techniques that are used to optimise lift operation and minimise passenger waiting time, especially at peak traffic periods in Section 2 3. Section 2.4 gives four examples of supervisory control systems Section 2.5 describes in greater detail the Express Traffic Quantum (ETQ) lift scheduler used by Express Evans Ltd. ETQ is used as a benchmark against which the performance of the intelligent scheduling system presented in this thesis is compared.

In modern high rise buildings, a lift system is expected to respond to passenger demands in such a way as to quickly and efficiently transport passengers to their destination without excessive passenger waiting times occurring or queues developing, especially at peak times. An average waiting time of no more than 15 seconds is typically specified for a busy office building [Barney and dos Santos 1985] To achieve such a performance, the lift system has to calculate a schedule as rapidly as possible; for example the calculation of a schedule takes approximately half a second in Kone lift systems [Siikonen 1997]. Due to the complexity of the problem, the computational cycle may also lengthen if the number of floors served is increased; for example in a 16 floor building the ETQ scheduler needs 250 ms to calculate the schedule while the same scheduler needs 450 ms to determine a schedule when serving a 32 floor building (Express Evans Ltd personal communication).

In a lift system, the assignment of a landing call only becomes irreversible once the lift begins to slow down in order to stop at a floor Before this, the lift scheduler is normally at liberty to reassign the landing call to another lift should the scheduler decide that the overall waiting time for the building, as a whole would be reduced.

If there are n lifts in the group and the number of landing calls registered is p, then there are n^p call assignment possibilities. The task of the lift scheduler is to decide which among these possible solutions is the one most suitable for achieving the specified objective, typically one of minimising: waiting time at the floor, the duration of the ride, the number of intermediate stops, or energy use [Ovaska 1991]. Attempting to consider all possibilities will often take more time than is available to the lift scheduler. Hence, the lift scheduler has to shorten the scheduling time by using some rules that will reduce the number of solutions that need to be considered. Often, the resulting schedule will be suitable for the purpose, but not necessarily optimal.

Traditionally, each lift has a *lift next direction* indicator above its door in the lobby that illuminates once the lift is irrecoverably committed to the landing call. Some Japanese-manufactured lift systems commit a lift to a landing call immediately, a lift is first assigned to it and the lift indicators at the lobby signal this. This method attempts to reduce the 'psychological waiting time' for the passenger at the expense of lengthening the actual waiting time, since the optimal assignment will normally change in the time the lift takes to arrive at the landing [Siikonen 1997].

The type of building, the traffic pattern and its intensity all have significant effects on the lift scheduler design. The following section explains the methods used to adapt the lift scheduler to changing passenger demands.

2.1 Lift system traffic patterns

This section discusses the different traffic flows that can occur in a lift system. These depend on the type of building and its occupancy, for an example, in an office building the occupants may be subject to strict times for starting, breaks and leaving. During the day, different traffic patterns are likely to be morning 'up peak', evening 'down peak', mid-day four-way traffic and random (balanced) interfloor traffic. At the lift system design stage when the number of lift cars is chosen, their capacity would need to be adequate to handle the anticipated building traffic and the lift real-time control system has to be chosen so as to be able to respond to traffic demands. Several lift control strategies have been implemented to optimise the lift system performance according to lift traffic [Strakosch 1983, Barney and dos Santos 1985, Siikonen 1997].

The main traffic streams are defined below.

- **Up peak traffic** An up peak traffic condition exists when the dominant or only traffic flow is in an upward direction with the majority of passengers entering the lift system at the main terminal of the building. The lift control system performance is usually measured during the up peak period, as at this time the lift system experiences the greatest change in traffic demand with passengers arriving in the morning to start work or returning following a lunch break [Strakosch 1983, Barney and dos Santos 1985, Siikonen 1997]
- **Down peak traffic** A down peak traffic condition exist when the dominant or only traffic flow is in the downward direction with all or the majority of passengers leaving the lift system at the main terminal of the building [Strakosch 1983, Barney and dos Santos 1985, Siikonen 1997].
- **Two and four way traffic** Two way traffic condition exist when the dominant or only traffic flow is to and from one specified floor, which is not the main terminal Four way traffic conditions exist when the dominant or only traffic flow is to and from two specified floors, one which may be the main terminal [Strakosch 1983, Barney and dos Santos 1985].
- **Interfloor traffic** Random interfloor traffic can be said to exist when no discernible pattern of calls can be detected [Barney and dos Santos 1985, Siikonen 1997].

2.2 Overview of different lift control techniques and services

In this section, the factors that are involved in the lift system design and tuning are discussed and the use of different control policies and their effects on lift system performance are explained. The lift policies are mainly concerned with providing a control algorithm which may consist of more than one policy to deal with different traffic demands. Dividing the building into zones and distributing lifts between those zones is one way of simplifying control and improving the performance of lift systems. The divisions are typically selected depending either on floor population, traffic intensity or the number of landing calls in a zone.

For an existing building the traffic patterns and floor utilisation information are available and therefore, compared to the design stage, it is easier to define which type of control algorithm and which policies are the most appropriate. Prior knowledge of traffic intensity patterns would help the lift control system to adapt its policy accordingly. Early lift scheduling methods included little or no prediction either of the passenger arrival rate or of the lift system future state. However, improvements in computer processing speed and memory capacity allow modern lift control systems to incorporate traffic prediction and detection in order to enhance the lift system performance

At the design stage the lift designer mainly depends on information available about the building and assumptions regarding the number of passengers which will be using the building, their probable traffic patterns and their intensity. Statistical analysis is often used to estimate the demands the use of a building will make on the lift system, based mainly on analytical techniques applied to up peak traffic [Barney and dos Santos 1985]. A number of parameters relating to building information is required in the design process, some of which are known precisely, such as the number of floors, purpose of the building, the relation between floors (for example, a restaurant may be located at a certain floor, a building may house several companies with each company having a certain number of floors and a certain population). Other values may not be strictly known, such as the times at which different companies start work and whether they work flexible hours. At this stage, the lift designer may have to determine the optimum control algorithm based on the information available for the building which gives the best performance in terms of *quantity* (number of lifts required and their capacity) and *quality* of service (average passenger waiting time or journey time). Note that ***passenger journey time*** is defined as the sum of the passenger waiting time plus transit time [Barney and dos Santos 1985, Siikonen 1997]. Other factors affecting the calculation include whether or not:

- floors are evenly populated;
- interfloor heights are constant;
- various delays (passenger disturbance, despatch intervals, loading intervals) can be considered negligible;
- passengers arrive uniformly with time;
- the number of people each lift transports on each trip meets the loading assumptions (80% or more of the lift capacity is used at peak time).

Depending on the complexity of the problem and the adopted solution, lift supervisory systems can be simple program or be complex multi-tasking systems in which a number of control algorithms is required to cover such conditions as up peak, down peak, heavy floor demand, balanced traffic, off-peak, basement service and night service [Barney and dos Santos 1985].

During up peak the average time the lifts takes to serve all car calls and return to the terminal floor is often estimated using probability theory. Typically, in conventional lift systems, the total number of passengers a lift system can transport during that five minute period of heaviest demand and the duration of the interval between consecutive departures from the terminal floor (calculated using average round trip time in Section 2.2.3) are used as performance measures [Barney and dos Santos 1985, Siikonen 1997].

Lift simulators are often used to assess the performance of the lift system under a variety of different operating conditions, such as changes to the number of entrance floors Due to the NP-complete nature of the scheduling problem, Chapter 1, the impact of the lift supervisory control system on passenger service or journey time under different traffic situations cannot be calculated analytically [Siikonen 1997].

2.2.1 Parking

Under light to medium traffic conditions, a lift has frequently no calls to answer. The lift is free for allocation if no further demands exist, and it can then be parked at a convenient floor or sector in the zone, where a zone is a group of adjacent floors

Parking is mainly intended to distribute the lifts around the building, in anticipation of traffic demand and a proper parking policy is essential for good lift system performance An example of a parking policy is given below [Barney and dos Santos 1985, Siikonen 1997]

- The building is divided into a number of parking zones, each zone grouping a number of adjacent floors; the main terminal constitutes a parking zone in itself
- A lift with no passengers and no calls allocated is considered a free lift. After the lift has been free for a defined time, it may be moved to a particular floor as part of the parking strategy.
- If the main terminal is vacant, the nearest free lift is 'expressed' down.
- The distribution of free lifts takes place when the algorithm detects more than one free lift inside the same parking sector.
- During up peak, the parking algorithm may be cancelled.

In Siikonen [1997], during light traffic statistical passenger traffic forecasts are used for parking lifts at those floors where passengers are expected to arrive next The statistical forecasts are searched as long as the number of arriving passengers exceeds a certain percentage of the up peak handling capacity. A building is divided in to sectors with equal arrival rates, and sectors are given priorities, where the priority of the sector is found by dividing the passenger arrival rate in the sector by the number of floors inside the sector. The sector with the highest arrival rate is served first and a lift that remains free for a predefined time is returned to the busiest floor in the sector. The parking operation is cancelled if new landing calls are registered and the reason for lift assignment need to be considered, and this is probably used to limit additional trips in periods when the lift is likely to be assigned.

2.2.2 "Next lift" procedure and terminal floor

The terminal floor is often a subject of preferential service. In a priority system, a free lift is usually allocated to the terminal floor as its first operation, although it usually allows the lift to pick up high priority calls on its way down to the main terminal.

Usually one of the lifts parked at the terminal floor keeps its door open as an indication that it will be the next lift to leave and it is not available to serve other floors until a specified period of time has elapsed [Barney and dos Santos 1985].

2.2.3 Subzoning and up peak service

The up peak service is usually triggered when the traffic leaving the terminal floor exceeds a predetermined rate The up peak service is responsible for supplying the terminal floor with more than one lift car in response to a single landing call. The ***up peak percentage arrival rate*** depends on the maximum number of passengers who arrive at the main terminal of a building for transportation to the upper floors over the most intense period The up peak percentage arrival rate is expressed as a percentage of the total building population, which can vary from 5% to 25% depending on the type of building. Most algorithms start an up peak service on the detection of a lift load at the main terminal floor in excess of a predetermined weight, but also taking into account the number of car calls to avoid the effect of temporary traffic fluctuations. Normally a false landing call is set up at the main terminal floor to ensure that a lift is available as soon as each lift departs and the up peak algorithm returns the lifts to the main terminal as soon as they have answered their last call. The "next lift" policy is that, if there is more than one lift at the terminal floor, then the second lift will keep its doors closed until the weight in the first lift exceeds some predetermined value [Barney and dos Santos 1985].

The most common form of lift control system for a group of lifts divides the floors of the building into zones. A ***zone*** is a number of floors usually adjacent in a building which are serviced by a group (or groups) of lifts Zones may be divided into ***subzones***, with approximately the same number of subzones as there are lifts in the group [Thangavelu Aug. 31, 1993]. The occupants of each subzone may be associated with each other and may be expected to generate some interfloor movement within the zone. Subzoning is usually used in high rise buildings over 20-30 stories, such as in the 100 storey New York World Trade Centre and the Chicago Sears Tower. Both of these buildings have three zones with one main entrance and two sky lobbies Each zone has four groups of lifts. low rise, medium low-rise, medium high rise and high rise [Siikonen 1997].

Barney and dos Santos [1985] explain that in a building or a zone with large number of floors (for example 18 floors), subzoning may improve ***up peak traffic handling capacity (UPHC),*** which is defined as the total number of passengers a lift system can handle during the five minute period of heaviest traffic. For example a zone may be divided into two *subzones,* which do not depend on landing calls but only on the passenger destination. One subgroup of lifts is allocated to serve the traffic terminating at the upper subzone and the remaining lifts serve the lower subzone. An indicator at the terminal floor is used to announce the available destinations of each subgroup of lifts serving the subzone. By reducing the number of available destinations and hence the number of stops, the return trip to the terminal floor can be reduced, thereby increasing each lift's handling capacity. This is demonstrated in the argument which follows The ***round trip time (RTT)*** is the time in seconds for a single lift trip around a building from the time the lift doors open at the main terminal until the doors reopen when the lift has returned to the main terminal floor.

$$\textbf{\textit{5 min HC for a single car = number of passengers per trip}} \times \frac{\textbf{\textit{300}}}{\textbf{\textit{RTT}}} \tag{2.1}$$

Where the period of 5 minutes for the handling capacity definition has achieved general acceptance, as the time during which conditions in the lift system remain reasonably fixed. The number of passengers in a lift during a journey leaving the terminal floor at up peak is given by

$$\textbf{\textit{Number of passengers per trip = average load = 80\% lift contract capacity (CC)}} \tag{2.2}$$

and, HC can now be written

$$HC = \frac{80}{100} \times CC \times \frac{300}{RTT} \qquad (2.3)$$

The ***up peak handling capacity (UPHC)***is the handling capacity of all L lifts in the group

$$UPHC = L \times HC \qquad (2.4)$$

Reducing the number of lifts usually implies longer passenger waiting periods Hence, the ***up peak interval (UPINT)***, which is the average interval between lift arrivals at the main terminal to each subzone, increases because fewer lifts are serving the subzone. However, under heavy traffic condition, subzoning frequently reduces the average passenger waiting time, allowing the lift system to cope with traffic rates that would normally cause saturation in the lift system. Moreover, reducing RTT means reducing passenger average trip time, as shown by the relationship

$$UPPINT = \frac{RTT}{L} \qquad (2.5)$$

UPPINT and UPHC are generally used as metrics for evaluating performance. Statistical relationships between the number of floors in a building and the number of passengers carried during a lift trip are used to define the expected number of lift stops and the mean highest reversal floor for different lift configurations, allowing the lift designer to calculate the expected RTT for a single lift under up peak condition [Barney and dos Santos 1985, Barney 1992]

The efficient partitioning of the building into zones and the allocation of lifts to each subzone requires the prediction of traffic patterns and adaptation to their changes. The introduction of prediction (perhaps using artificial intelligence techniques) could provide a dynamic zoning system which modifies zones according to changes in passenger demands.

When zoning is not used, reducing the number of stops made by each lift may mean that the lift is below its normal average load Making the most of the available lift capacity often involves using fewer lifts, implying that less of the building space is used for transportation In such a case a balance has to be reached between the need for fewer lifts and the number of stops made.

During the up peak period lift systems have the following drawbacks [Barney and dos Santos 1985].

- A system relying on lobby preference to bring lifts to the main terminal suffers from the extra time to bring a lift down from elsewhere in the building. Prediction could be used for early detection of known traffic conditions, and an example of such a system is introduced in the next section
- If a system does not employ a loading interval approximately equal to the time to half fill a lift, lifts will leave the main terminal with less than their average capacity.
- A system which does not possess the "next lift" feature (Section 2.2.2) will allow more than one lift to service landing calls, making inefficient use of the available capacity.

Under the above conditions, lift availability, capacity and available destinations are not efficiently matched to the immediate needs of the passengers resulting in an increase in the passenger waiting time. The common usage of a constant dispatching interval does not maximise load nor minimise the number of stops that are made before the lift returns to the terminal floor to receive more passengers. One solution to this problem is *variable lift up peak dispatching interval*, meaning that a lift might depart only after a convenient interval between it and other lifts has elapsed. Another solution is to use the *adaptive assignment of lift car calls*, in which lifts are assigned floors, based on car calls that are entered from a central location (which adds to the cost overhead) and passengers are directed to the lifts [Thangavelu Aug. 31 1993].

2.2.4 Sectoring

Sectoring is a method used to group a number of landing calls together for lift car allocation or parking purposes. A lift is assigned to service a particular demand sector and prevents other lifts from responding to landing calls within that sector [Barney and dos Santos 1985]. A building or a zone can be divided to sectors, note that in subzoning a subzone is also served by one lift, however a subzone does not depend on landing calls but on the passenger destination, section 2.2.3. As in zoning, the idea of sectoring is that each lift serves fewer floors, thereby minimising RTT. The aim is to distribute lifts evenly around the building or the zone allocated to them and this can be partly achieved by grouping those landing calls to be answered by individual lifts. The assignment of lifts to sectors depends on the number and the position of available lifts and on the sector priority levels. The sector with the highest priority is usually assigned first and if more than one lift is free, the nearest lift is allocated to the sector. Each sector is timed as soon as a landing call is issued to register waiting time The timing is used to measure the priority level for the sector so that the longer the waiting time the higher is the priority. The time interval a sector would take to reach a certain priority level may vary to allow some sectors to have an overriding priority and to provide a preferential service to floors. As a sector reaches its highest priority, the nearest lift travelling towards the sector bypasses all landing calls to service the sector.

There are two type of sectoring, namely static and dynamic, and these are explained in detail below

Static sectoring

In static sectoring, a fixed number of landing calls is grouped together There are two types of static sectoring, namely common and directional

In *common sectoring* [Barney and dos Santos 1985], a fixed sector is defined for both up and down landing calls originating from a number of contiguous landings The terminal floor is usually a sector and the number of sectors defined normally depends on the number of lifts. Lift allocation is carried out as follows.

- A lift is allocated to a sector if it is present in that sector and the sector is not committed to another lift
- Fully loaded lifts are not considered for allocation.

- An assigned lift operates under the *directional collective principle* within the limits of its range of activity, which may cover unoccupied sectors above the lift, or in some cases below it. A sector becomes unoccupied when a lift leaves the sector or is fully loaded. Any lift movement is then made in response to registered car calls, landing calls registered inside the sector, landing calls adjacent to the sector in which the lift is assigned and landing calls in the contiguous unoccupied sectors above the lift if it is moving in the up direction, below if the lift is moving in the down direction.
- A lift is allowed to answer landing calls along its way even if it is not assigned.
- The assignment of a lift to a sector ceases when the lift leaves the sector or becomes fully loaded within the sector.

Directional sectoring [Barney and dos Santos 1985] uses a fixed sector that includes a number of contiguous landing calls defined for one direction only. The lift will respond as follows.

- A lift travels express to the assigned sector and answers the landing calls within the sector
- As it leaves the sector it answers landing calls along its way.
- A lift becomes free when it answers the last car call.
- If there are any passengers at the lift's last car call floor, they are discouraged from entering the lift by illuminating "DO NOT ENTER" sign and dimming lights in the car.
- As the up direction sectors have higher traffic demands, the terminal floor is often a sector itself, and there are fewer floors in adjacent up sectors than in the sectors further up the building Similarly for down direction sectors, the upper sectors contain fewer floors than do the lower sectors.

Dynamic sectoring

The building is divided to sectors whose size and number are not defined during design, but are altered according to the instantaneous state, position and direction of travel of the lifts. Like zoning, this method has the disadvantage of needing an extra display indicator for each lift at each floor to announce the lift destinations [Barney and dos Santos 1985].

2.2.5 Service shutdown

Under conditions of low demand, a control system should proceed with a shutdown procedure to save wear, tear and power. To achieve this a common procedure is to switch off the motor generator if it has remained inactive for a predetermined period of time, and as demand increases the lift can be automatically returned to service [Barney and dos Santos 1985].

2.2.6 Basement service

It was observed [Barney and dos Santos 1985] that basement landing calls demands significantly degrade the lift system performance, especially during up peak and down peak. Most systems provide only a restricted service or low priority to and from basement and sub-basement floors, or in some cases a separate vertical transportation system is provided.

2.3 Lift control algorithms

This section gives the objectives, rules and constraints involved in the design of a lift control system with some examples of control algorithms and their advantages and disadvantages

A lift supervisory system is required to achieve both control and management of a single group or a number of groups of lifts. An example of a low-level conventional method of landing call allocation is the ***simplex collective*** lift control system, which simply commands individual lift cars to answer landing calls in their direction of travel, to stop or start and to open and close the doors [Barney and dos Santos 1985] The simplex collective method generally results in calls being answered in a different order to their registration. A higher level of traffic control has the function of co-ordinating the activity of group of lifts cars, by means of a set of logical rules defined by the lift designer with the aim of:

- providing a service of the same priority to each floor in a building;
- minimising passenger waiting time at each landing,
- minimising journey time,
- adequately servicing as many passengers as possible.

In general, certain constrains regarding lift system operation must be implemented in order to meet the expectations of passengers and these generally include the following.

- A lift may not pass a floor at which a passenger wishes to alight.
- A lift may not reverse its direction of travel while carrying passengers.
- A passenger may not enter a lift travelling in the reverse direction of travel. Car calls in the opposite direction of the lift are usually postponed until the lift answers all assigned calls in its way of travel
- A lift should not stop at a floor where no passengers enter or leave the lift.

2.3.1 Fixed sectoring priority timed system

In fixed sectoring priority timed system named as FS4 by Barney and dos Santos [1985], the directional form of static sectoring is used, in which each sector is timed as soon as a landing call is registered within its limits. The features of the algorithm are outlined below.

- Preferential service is provided to the terminal floor.
- A special algorithm is provided for up peak traffic where a false landing call is issued at the terminal floor when a lift departs the terminal floor with more than 50-60% load. If there are no lifts at the terminal floor then the first available lift is expressed down bypassing all landing calls except landing calls with the highest priority. Occasional down calls registered during the up peak are most probably only served when they reach highest priority; where their priority increases with time that has elapsed since the landing call was registered

- There is no special algorithm for down peak traffic. If all down travelling lifts become fully loaded, all the up peak sectors are reduced to a minimum priority.
- Light traffic, shutdown, next lift and energy saving procedures are also provided.
- A landing call with the highest priority is served immediately by the nearest lift travelling towards it. This feature is useful for sudden traffic demands when a lift becomes full before answering all landing calls at its sector, the sector priority is advanced then to highest priority

The advantages were as follows.

- Good up peak performance
- Very good down peak performance.

The disadvantages were as follows.

- Fair interfloor traffic performance but not as good as that obtained from dynamic sectoring. This is mainly due to the absence of parking policy as the allocation procedure only distributes lifts when landing calls exist. A uniform performance is obtained for all floors, but average waiting times tend to become extended.
- Should the original traffic flow vary considerably from the original specifications, it is not easy to adjust the control system in the field to the new conditions.

2.3.2 Fixed sectoring common-sector system

This algorithm named as FS0 by Barney and dos Santos [1985] has the following features

- A parking policy is provided. A free lift moves to the nearest vacant sector in the lift direction, and, if this is not available, then the movement is to the nearest vacant sector in the opposite direction. The highest priority is given to the terminal floor.
- To set the sector priorities, the number of landing calls per sector is compared with the average number of landing calls per sector. Extra lifts are brought to the highest priority sector, by-passing the landing calls at other sectors

- During up peak, only car calls are answered until returning to the main terminal
- On detection of down peak, the allocation of lifts to the main terminal is cancelled. No special algorithm is provided.

The advantages were as follows

- Lifts are distributed evenly in the building.
- Good performance is obtained under balanced interfloor traffic, up peak and for unbalanced interfloor traffic.

The disadvantages were as follows.

- No proper procedure is provided to respond to sudden heavy demands at a floor.
- Poor service is provided to lower floors in the down peak owing to the recycling of the lifts to unoccupied sectors.

2.3.3 Dynamic sectoring system (DS)

Dynamic sectoring [Barney and dos Santos 1985] is suitable for light to heavy balanced interfloor traffic. DS is complemented by a number of other control algorithms to cater for unbalanced traffic conditions, as detailed below.

- A free lift control algorithm is introduced with the intention of inserting free lifts ahead of lifts located to high demand sector. The algorithm continuously monitors the number of car and landing calls within the normal dynamic sectors by counting the calls from the current lift position and comparing this with a predefined value A free lift is also sent if a landing call is issued within 2 seconds of a fully loaded lift departing from that floor. The insertion of a free lift into a heavy traffic section instead of attempting a redistribution of lifts makes this control algorithm highly efficient The free lift control algorithm works in parallel with the balanced interfloor traffic algorithm, where a free car allocated to a high demand sector is taken out of the basic control algorithm

- Up and down peak traffic periods are detected using a forward-backward counter which increments each time the lift load of a departing car exceeds a predetermined value and decrements each time a lift departs with a load below this value. Decrements can also occur after a period of inactivity. The peak period algorithm is turned on or off when the counter reaches a turn-on level or turn-off level.

- The building is divided into fixed parking sectors, and a free car is parked by issuing a dummy landing call for each of the unoccupied sectors. The main terminal is granted preferential service.

- The up peak algorithm returns all lifts to the main terminal as soon as no car calls are registered At the terminal floor, only one lift opens its door for loading and starts moving after an adjustable time period or when the load exceeds a fixed percentage of its maximum.

- An up peak subzoning algorithm is provided as an option where the building is divided into subzones based on passenger destinations

- A down peak algorithm is provided, in which down landing calls are grouped together and lifts are cycled between them, each lift being allocated a group which it services according to a collective principle Once a lift has dealt with all calls in its group, then, if it is not fully loaded, it is assigned to the nearest free group If the lift becomes fully loaded before answering all the calls within the group, then the group is denoted to be free, and eventually regrouping takes place. During one service cycle a maximum of one lift is sent to any one landing The service of up calls during down peak can be either suspended or restricted, in a manner similar to the handling of down traffic during up peak.

A very good performance is achieved under balanced or unbalanced interfloor traffic conditions. However, the down peak algorithm does not perform particularly well, with too many landing calls being bypassed, leading to an increase in lift travel time. If the lift is full and passengers are left, they must wait for a complete cycle of service before being collected as a floor is only served once during a service cycle.

2.4 Examples of supervisory group control

Most supervisory control systems employ algorithms and control policies that work well in up peak traffic conditions but perform not so well in other traffic conditions such as down peak and normal traffic conditions. For down peak, Barney and dos Santos [1985] produced the following findings.

- Dynamic sectoring and simplex collective principle control policies both suffer from poor service to lower floors resulting from lifts becoming full at the higher floors. Proper lift cycling is required to provide a uniform service The FS4 algorithm has tackled this problem, and despite the absence of a down peak function, the service to lower levels does not depend on spare lift capacity. This is achieved by detecting heavy down peak traffic and providing a restricted service to low intensity up traffic while retaining no lifts at the main terminal. Floors with the highest priority are given immediate service.

- The quantity of service does not dependent on the control policy, but its performance, in terms of waiting time, is heavily dependent on the policy. Consequently, for most lift systems, in order to make full use of the down peak handling capacity, it is necessary to allow for high average waiting times.

During balanced interfloor traffic [Barney and dos Santos 1985] the performance of fixed sectoring lift systems degrades severely with demand and is dependent on the type of supervisory control system implemented On the assumption of an evenly populated building, waiting times can become extended if this assumption is changed A dynamic sectoring system is generally more efficient under such conditions than are fixed sectoring systems.

Chan and So [1996] used dynamic zoning to improve the control system performance during both up peak and down peak traffic, but switched off the zoning when the traffic condition return to normal. They found that dynamic zoning did not improve lift system performance under balanced traffic conditions in contrast to dynamic sectoring.

It is apparent that an algorithm that is suitable for one traffic condition may not be suitable for other conditions A mixture of policies could be adopted [Barney and dos Santos 1985] such as:

- an up peak algorithm with fixed up peak subzoning;
- a fixed sectoring priority timed down peak algorithm;
- a dynamic sectoring algorithm to cater for balanced interfloor traffic;
- insert free lifts ahead of lifts serving heavily loaded dynamic sectors,
- provide a parking algorithm.

Observations made by [Barney and dos Santos 1985] by inspection of data logged from commercial building has demonstrated that the following conditions occur.

- Both up peak and down peak can last longer than predicted.
- During up peak, significant down traffic may exist and conversely, during down peak significant up traffic may exist.
- Algorithm switching caused by an incorrect detection of a specific traffic condition can seriously degrade performance.

An intelligent detection system may be able to predict the start and end of a peak period by monitoring traffic patterns and a single allocation algorithm which adapts to changes in traffic conditions

The general characteristics of the algorithms discussed above are as follows.

- The systems are multi-algorithmic, meaning that different sets of rules are used when handling different traffic patterns. The default algorithm is that for interfloor traffic, otherwise an up peak algorithm is used Down peak is served by an interfloor algorithm in both fixed sectoring systems and incorporates the cancellation of the allocation of lifts to the main terminal.
- Where different algorithms are used to handle up peak or down peak traffic, the change-over between algorithms is usually abrupt This can cause disruptions to the

current traffic, although the severity of the effect depends on the intensity of the current traffic. When detection is incorrect, the effect is most severe Having an intelligent early detection system and the use of a single control algorithm which can adapt to different traffic conditions can provide a smoother transition, or eliminate the requirement altogether.

- The up peak control algorithms are similar in all cases discussed.
- Partly loaded lifts serve landing calls in the path being followed.
- Both of the fixed sectoring systems lose further control of lifts once allocated, while free or uncommitted lifts are allocated to demand sectors. Once the lift answers a landing call in its allocated demand sector it continues to answer all other landing and subsequent car calls in simplex mode Thus committed lifts work virtually independently of one another, defeating the objective of group control.
- Static sectoring algorithms generally allocate a lift to a sector without checking whether there is better choice of lift. A loaded lift is allowed to serve the sector ahead of it, without checking if another lift has already been allocated. This affects the fairness of assignment and causes longer passenger waiting times.
- The common sectoring FS0 and dynamic sectoring systems have no concept of passenger waiting time. Thus it is not possible to give urgent attention to landing calls which have been waiting for a long time.

The following sections include discussions of some of the more common lift supervisory systems in use, namely the intelligent up peak contiguous floor lift system, up peak and down peak prediction, landing calls queue scheduler and weighted relative system response.

2.4.1 Intelligent up peak contiguous floor channelling lift system

The up peak channelling system with optimised preferential service to high intensity traffic floors was developed by Otis [Thangavelu Feb. 2 1993] and uses dynamic sectoring with the help of an intelligent prediction system. This scheme was introduced to solve the up peak problems discussed in section 2.2. An example of a building with four lifts is given, three of them dedicated to sectoring, while the fourth is free to answer other calls at any of the floors in the building. An intelligent traffic predictor is used to estimate the traffic for the next five minutes and is able to assign dynamically the floors to sectors, and these need not be of the same volume. Their estimation is made using traffic measured during a small number of previous time intervals on the same day and the current sector is indicated on the lift display at the terminal floor. During each five-minute interval, the traffic volume is determined and compared with the determined average traffic per sector. The frequency of service of a lift to each sector is variable, based on this comparison Thus, sectors having more traffic are serviced more often relative to sectors having less traffic.

Such intelligently assigned sectoring is aimed at reducing the length of passengers queues, the waiting times and the trip times at the terminal floor by achieving a more uniform loading of lifts which also increases the handling capacity of the lift system.

2.4.2 Prediction of the start and the end of up peak and down peak traffic periods

Thanngavelu [Aug. 31 1993] describes a method employed by Otis for selecting suitable algorithms for up peak, down peak and normal traffic periods. The most common approach is to assume that the start of a peak period occurs when the time interval between two consecutive lifts either leaving the terminal floor with more than a specified load or arriving at the terminal floor with more than a specified load, is sufficiently short Such a scheme often causes delays at the onset of a peak period in the dispatch of empty lifts from the upper floors to the terminal floor during the up peak period and from the terminal floor to upper floors during the down peak period. This results in long passenger queues and increases waiting time at the terminal floor at the start of the up peak period and at upper floors at the start of the down peak period. In addition, the forming of sectors for up peak channelling and zones for down peak operation is delayed Similarly, the end of the up peak period is normally assumed to occur when no lift leaves the terminal floor with more than a specified load during a certain time interval. The end of a down peak period occurs when no lift arrives at the terminal floor within a specified interval and with more than a specified load. However, such an empirical scheme may well deactivate the peak period dispatch strategy too early resulting in poor service to interfloor traffic.

An AI based method was introduced by Otis to predict the start and the end of the peak periods. The terminal floor traffic data, passenger boarding and de-boarding, and the number of arrivals and departures at the terminal floor are collected during the day and are used to predict the peak-period times. The times when the loads of consecutive lifts leaving the terminal floor breach a certain level are recorded for a number of days. The recorded data are smoothed, combined with the current day's data and used to predict the arrivals and hence the start and end of peak periods. The aim of the strategy is to reduce the passenger queue lengths and waiting times which could have otherwise been caused by the delayed start of the peak period or by its premature termination.

2.4.3 Landing calls queue scheduler

This section describes a lift scheduler that is suggested by Barney and dos Santos [1985] They claim that a fair service for each landing call can be obtained by calculating all the possible paths for each lift and then finding the optimum path for the landing call taking into consideration which call has been waiting the longest. The priority is inherent in the first-in, first-out queuing method used. This is augmented by the allocation of higher priority to those landing calls registered for longer than a specified threshold time, in which a lift passing a landing call in its direction can only be answered on condition that there is no other call which might have to wait for more than the threshold value. A landing call is taken from the head of the queue and allocated to a suitable lift using the concept of minimum cost, where a suitable cost function can be based on one or both of the following.

Quantity of service This is a measure of lift capacity consumed to serve a specific set of calls and is indicated by the total journey times of all the lifts.

Quality of service This is indicated by the average value of either the passenger waiting time or the passenger journey time.

The minimisation of the total lift journey time implies using minimum system capacity, which is equivalent to using the smallest possible number of lifts, section 2.2.3. The result of this policy would be longer passenger waiting times. This criterion alone is thus not suitable as a cost function and the cost of an allocation of a new landing call to a particular lift is instead given by the incremental value of the expected extra system response time due to the new allocation. In the special case when all lifts are uncommitted to any allocations and a new landing call occurs, then the cost can be represented by the system response time of the nearest lift to the call The allocation procedure thus has an in-built capability to assign a landing call to the nearest lift

During balanced interfloor traffic, a lift serving a landing call will in general make at least one further stop (car call) before becoming free. The number of stops made by lifts per unit time indicates the level of loading and any reduction in the number of stops will clearly conserve system capacity. This can be achieved by serving a landing call where a car call already exists. If the expected value of landing call waiting time is more than a threshold time, then the landing call should be allocated to the particular lift even if other allocations are smaller. This has the effect of conserving lift system capacity to serve other demands better. The threshold time is not a constant but varies with demand to enable an even service.

Unbalanced interfloor traffic occurs when consistently more passengers arrive at a particular floor. An even distribution of service is still required and the handling of the unbalanced traffic requires both detection and correction. The number of passengers boarding at a particular floor can be detected as an excess load in the lift on arrival at a floor. Then, the heavy duty floor is moved to the head of the landing call queue, where it will receive immediate treatment Should two heavy duty floors exist simultaneously, then their priorities would be determined on the basis of their respective waiting times. The level of the heavy duty floor must be reduced or removed as the unbalanced traffic disappears. Priority floors and preferential service floors are given a lower time limit on the waiting time. The landing call queue is arranged so that the various categories are ordered, from the head of the queue as follows:

1 Priority call (Higher level),

2 Heavy duty call (Level 2),

3 Priority call (Lower Level),

4 Heavy duty call (Level 1),

5 Long waiting call,

6 Normal call.

Siikonen [1997] described a similar scheduler which was used in the Kone TMS9000 system. Kone employed an AI system to learn the passenger flow in the building to help detect and optimise passenger waiting time. The predicted number of passengers is used to favour landing calls with more passengers waiting to landing calls with only a few waiting passengers. During heavy traffic peaks extra lifts are sent to popular floors according to the forecast pattern while during light traffic lifts are parked at floors with the most probable traffic. All landing calls are allocated continuously according to the enhanced spacing principle where lifts are kept evenly spaced and landing calls are allocated starting with the landing calls with longest queues and waiting times, and bypassing landing calls with short queues and waiting times. The passenger traffic flow detection method is further explained in Section 3 2.1.

Thangavelu [1989] described the Otis queue based lift scheduler which used traffic prediction to estimate the number of passengers waiting at a floor. Lifts are assigned to floor with long queues on priority basis, thereby decreasing the queue length and waiting time at the terminal floor and upper floors During up peak, the terminal floor has the highest priority while during down peak multiple queue levels and the percentage of waiting time limits are used. The passenger prediction technique is further explained in Section 3.2.1

2.4.4 Weighted relative system response

Another control algorithm similar to the landing call queue scheduler is the use of weighted relative system response (RSR) lift car assignment with variable bonuses and penalties described by Otis [Thangavelu 1991]. The landing calls are assigned to the lift as they are received, using initial values of bonuses and penalties to compute relative system response factors. The penalties and bonuses are selected so as to give preference to the landing calls that have remained registered for a long time, relative to the previous selected period's average waiting time of the landing calls. Lift travel time to a landing call is expressed in terms of various time related penalties (such as the number of stops or the door dwell time delay). These penalties are added together with other penalties that reflect undesirable choices for the assignment. Bonuses are given to desirable actions (such as assigning a landing call to a lift already travelling to a car call at that floor) and these are subtracted from the sum of penalties resulting in the RSR value. These values are calculated for each lift and the lift with the minimum RSR value is assigned to the landing call. When the landing call registration time is small compared to the selected time period's average waiting time, the bonuses and penalties are adjusted accordingly so as to give lower priority than those landing calls with large registration times.

This scheme is able to assign calls on a relative basis, rather than on an absolute basis, taking into account instantaneous system operating characteristics.

However the above scheme has the following disadvantages:

- landing calls are all treated equally without regard to the number of people waiting behind the landing call;
- lifts are all treated equally without regard to the current lift load and the expected lift load, unless the lift is fully loaded.

If as a result of these actions, the lift assigned becomes full, it may then be deassigned certain landing calls and allowed to answer its remaining assignments. To help solve this problem an AI based crowd sensing system was introduced for lift car assignment. This involved the prediction of traffic levels at all floors of building at all times of the working day based on historic and real time traffic predictions, Section 3.2.1 This information is used to predict the number of people waiting behind the call and the number of people expected to be boarding and de-boarding at lift stops. Any mismatch between the predicted spare capacity and the number of people waiting at the landing call then is used to honour or dishonour the lift in answering the landing call. With more people in the lift it will need to stop at a greater number of car calls, so generally increasing the waiting time. In the Otis system, a lift load penalty is applied that varies proportionally with the number of people in the lift, but at a lower rate than the function relating the penalty to the number of people waiting behind the landing call The resulting RSR value is affected by the lift load at the landing call floor, the number of people waiting at the landing call floor and the number of people boarding and alighting the lift on each round trip. Lift load and lift stops are distributed equitably, so as to minimise the service time and the waiting time of passengers and to improve handling capacity.

2.5 Express Traffic Quantum (ETQ)

The ETQ scheduler [Express Evans Ltd. Reports 1988-1994, Mulvaney and Sillitoe 1992] developed by Express Evans Ltd. is based on empirical observations. Many years of development and an intimate knowledge of lift systems have led to continual improvements in the scheduling algorithms. The lift supervisory system includes one scheduler for all types of traffic in addition to a up peak and parking policy. One parking policy is used except for up peak traffic where a special parking policy is adopted which gives a higher priority to the terminal floor

ETQ employs two separate operations, namely a quantitative algorithm (QA) and an expert algorithm (EA). Tables 2 1 and 2.2 outline the operations of the two algorithms. The ETQ algorithm provides continuous and comprehensive evaluations of all possible service options, followed by the selection of the final solution which is based on the lowest waiting time, and the lowest *match figure* which is used to indicate the closeness of each of the solutions to the previously selected solution. The higher the match figure the less switching and the more stable the system will be. Only one landing call is transmitted to a lift at a time, even if more than one landing call is allocated to it, allowing later reallocation

Up peak is detected when the lift load exceeds a predefined limit or when car calls in a lift leaving the terminal floor exceed a predefined number. During up peak lift parking is concentrated on the lower half of the building, and two lifts are provided at the terminal floor. The parking policy is triggered following an interval of inactivity. A building is divided into bands depending on their priorities and the number of lifts in service. A lift parking operation will be cancelled if it receives a true call

The assumptions made in the ETQ control system work well in practice, as proved by the many successful installations. However, the enhancement of the accuracy of those assumptions might achieve better performance, and some examples are listed below.

- The assumption of a fixed travel time between floors will produce errors in the estimates of expected time of arrival values.
- No use is made of the current speed of the lift (it may be stationary, have reached maximum speed in either direction, or be accelerating or decelerating).
- Loading and unloading times are estimated as fixed values. It is likely that the time to load or unload will depend on demand, with the respective times being of longer duration during the up peak or down peak periods. There is also likely to be a relationship dependent on the load status of the lift. To improve such estimates, an appropriate prediction system needs to be developed.
- It is assumed that for each landing call answered, two seconds per floor should be added to the travel time as far as the terminal floor. The estimation of these data could be improved by better prediction of the form of new car calls which result from landing calls.
- The status of the lift doors is not currently taken into account. Knowing the actual position of the doors will allow improved estimates of eta values to be produced. For example, if the doors are opening, eta values will be longer than those when the doors are closing
- Stability is measured in terms of landing call assignments, yet the decision whether to maintain the same assignment is probably better represented as a higher-level function The advantage of this approach is probably best illustrated by example Assuming that a further high-level function is that car calls are answered in preference to landing calls, the scheduling system may well be able to defer many cases of instability until current priorities have been satisfied. This highlights a major problem of systems such

as ETQ which have knowledge distributed throughout the code, namely that decisions are taken in isolation in disparate parts of the system. The solution is to combine such knowledge into a single, separate module so that global and better-informed decisions can be made. Such global knowledge bases also have maintenance benefits and can be more easily configured to suit individual customer requirements.

Each individual improvement of data quality may make only a minor effect on performance, and experimentation is needed to assess the improvements in performance which are available.

The operation of ETQ is known as opportunistic, since scheduling is repeated continually in order to react to changes in events as they occur (given that the cycle time is small in comparison with the critical scheduling time) and, apart from the match figure, the current system operates on current knowledge alone and has no record of past events. Opportunism is often employed in problems where little predictive information is available and in such circumstances has a number of advantages over those systems based upon a priori assumptions as these may become increasingly invalid as time progresses.

2.6 Conclusion

In high rise buildings, traffic density and pattern are the main processes that influence the lift system Traffic patterns depend on building type and occupancy, and so it is important that the lift supervisory system is designed in a way that it is sufficiently flexible to adapt to changes in operation that deviate from the original design assumptions, as well as being able to adapt daily to passenger traffic demands.

To maintain control and an even distribution of lifts in the building, a commonly used technique is to divide the building into zones and sectors and distribute lifts among them Each sector is served by one lift and the lift scheduler is replaced by a simple collective control policy. A second technique uses a scheduler which searches for the best lift assignment that can be found within the computational interval allowed and the scheduler is called on a regular basis to allow the assignment to adapt to traffic fluctuations. Other control policies and services may be added to complement the main lift control algorithm.

Lift schedulers may be used in conjunction with zoning, especially in high rise buildings with a number of floors above thirty. Each zone is served by group of lifts controlled by a seperate lift scheduler

Some supervisory lift control systems are multi-algorithmic, in that a variety of algorithms are available; one for each identified traffic condition. In order to exhibit a smooth performance under all traffic conditions, an early traffic detection and prediction system is needed to identify when the control algorithm or policy should be changed. This gives an added complexity to the lift supervisory control making its maintenance more difficult.

Other lift systems have more than one control strategy depending on the optimising factors. In practice, these factors may conflict, such as minimising passenger waiting time and energy consumption Different control values may be used to balance assignments according to certain traffic condition or strategies such as the use of weights, bonuses, penalties, priorities and threshold values These values have to be set by the lift designer, the operator or even by another control policy.

The above leads the author to conclude that in order to implement an adaptive lift control system with the required performance, a simpler approach should be taken. The development of one-off systems is made simpler if the lift scheduling system can be easily configured to suit the building design, can be adapted to specified priorities for lift scheduling and can adopt particular scheduling strategies depending on traffic conditions. Consequently, the issue of flexibility must be an important factor in the adoption and implementation of new techniques. As will be shown in the next chapter, modern intelligent lift control systems comprise two main parts

1. The monitoring system which is responsible for obtaining the knowledge required by the lift control system, for prediction and statistical calculations such as passenger arrival rate. The knowledge is used in the detection of peak traffic periods, and the for control policy selection which depends on traffic prediction and strategy followed when there is more than one optimisation goal, Chapter 1.

2. The real-time lift scheduling system which is responsible for allocating landing calls to lifts based on data available from the lift system and information provided by the monitoring system.

Quantitative algorithm	
input data	**output data**
• lift position • lift direction • contract speed • car calls • landing calls • lift status (in or out of service) • lift assignment of previous solution • landings served by each car	For each of the 16 final solutions: • assigned landing calls • etas* for each landing call • waiting time for each landing call • average waiting time • longest waiting time • number of lifts used • energy used • match to previous solution • lift distribution
assumptions	
• lift speed is constant at the contract speed • loading time is constant • unloading time is constant • following assignment to a landing call, the lift travels to the terminal floor • following assignment to a landing call, the lift is subject to an additional 2 seconds delay per floor as far as the terminal floor	
operation	

for each Lift
    generate a free Path
    for each landing call (real or anticipated)
        generate a Path consisting of an eta list
    endfor
endfor
 	
for each fixed pairing of Paths (AH,BG,..)
    produce all Combined Paths keeping only the 4 highest ranked by Merit Figure
endfor
 	
for each fixed pairing of Combined Paths (AHDE, BGCF)
    produce all Combined Paths keeping only the 4 highest ranked by Merit Figure
endfor
 	
keep 16 Combined Paths as ranked by Merit Figure	

Table 2.1 ETQ quantitative algorithm

* *eta: estimated times of arrival*

Expert algorithm	
input data	**output data**
The outputs of the quantitative algorithm	A single final solution. • assigned landing calls • etas for each landing call • waiting time for each landing call • average waiting time • longest waiting time • number of lifts used
assumptions	
• stability is measured by the number of landing call assignments which are the same as that in the previous final solution	
operation	
select the final solution from the 16 based upon lowest wait, lowest average waiting time and highest Match Figure.	

Table 2.2 ETQ expert algorithm

Chapter 3

Survey of intelligent real-time lift scheduling systems

This chapter gives an overview of scheduling and intelligent scheduling techniques with particular application to real-time systems. The majority of published work concentrates on systems in which a separate off-line *monitoring* system provides the knowledge that aids and reduces the complexity of the operations of the real-time *scheduling* system.

Scheduling can be defined as an optimisation process whereby limited resources are allocated over time among both parallel and sequential activities, the goal being to minimise service time and maximise machine use while satisfying constraints [Uckun *et al* 1993, Zweben and Fox 1994]. There are two types of scheduling, namely *static* and *dynamic* Static scheduling algorithms have a complete knowledge regarding the task set and its constraints. Dynamic schedulers have a complete knowledge only of the currently active set of tasks, but new arrivals may occur in the future, not known to the algorithm at the time it is scheduling the current set, and therefore the schedule will normally change with time [Stankovic *et al.* 1994].

Scheduling methodologies fall into two categories

- Constructive methods that incrementally extend a partial solution until it is complete.
- Opportunistic scheduling in which rescheduling or schedule repair is performed on a complete schedule when a change of events occurs that affect the current solution

The majority of the artificial intelligence (AI) approaches to scheduling have been constraint-based search methods, in which constraints are used to prune the search for a solution. Zweben and Fox [1994] discuss alternative approaches to rescheduling such as backtracking and iterative repair, where repair operations performed on a full schedule will allow global constraints and optimisation criteria to be evaluated cheaply

In a dynamic real-time system the assignment of all resources has to be completed in a critical time interval The assignment of each resource is affected by both current and future resource assignments and heuristics, and therefore a complete schedule has to be produced in each interval Dynamic scheduling is required for real-time systems, such as lift systems, but very few theoretical results are known [Stankovic *et al.* 1994] A *real-time dynamic scheduling system* is event driven (for example a new schedule is calculated each time a landing call is issued), typically aided by an off-line prediction system, which provides the real-time scheduler with the expected dynamic environment conditions.

In the following sections a real-time lift scheduling system is taken as a case study The lift scheduler needs to operate in continuous state space and in continuous time but is normally implemented as a discrete, event driven dynamic system. Moreover, the state of the problem is not fully observable (for example the number of passengers is not accurately known) and is non-stationary due to changing passenger arrival rates [Crites and Barto 1996] The aims of this chapter are to introduce the relevant AI methods, to assess the scheduling policies and intelligent monitoring techniques found in the literature and to discuss their relative merits, Section 3 1. In addition, the intelligent lift scheduling systems found in the literature are discussed and compared, Section 3 2.

3.1 AI methods relevant to scheduling

This section briefly explains the main AI methods used in the intelligent scheduling systems, namely search techniques, real-time search techniques, reinforcement learning, neural networks and fuzzy logic.

3.1.1 Search techniques

Search is a traditional AI method for problem solving. The three classical types of problem that have been tackled by search algorithms are path finding problems, constraint satisfaction and two player games. Scheduling problems usually fall into the first two types. In general, a problem space consists of two components, namely a set of states which represent the problem *world* and a collection of actions that map one state of the world to another. The task of the path finding model is to find a suitable sequence of operations that map the initial state to the goal state The efficiency of search algorithms can be measured using the cost of the solution (for example passenger waiting time), the time required for the search, or the memory required for the search One way to reduce the complexity of a search is to provide additional knowledge about the problem being tackled, termed *domain-specific knowledge* or *heuristics* [Korf 1988, Rich and Knight 1991]. One popular algorithm which can incorporate simple heuristics is A* search The A* search is said to be *admissible*, that is, it is guaranteed to find a minimal path to a solution whenever such a path exists, where the determination of the minimal path is determined by the values returned by a heuristic *state evaluation function f*. The state evaluation function is the sum of two values, the cost of having reaching the current point in the search tree *g* and the cost of how far the current point is from the final goal *h*. The latter cost can normally only be estimated, but if it is never overestimated the solution will be optimal in terms of *g* [Mulvaney and Sillitoe 1992, Nilsson 1998] The estimation is done using a heuristic evaluator, which is generally a function that is inexpensive to compute and estimates the relative merit of different states relative to the goal [Korf 1990]

Korf [1988] presented research results regarding A* performance in term of the time required to find a solution which indicated that A* is the fastest search algorithm for finding optimal solutions, for a given non-overestimating heuristic function. A disadvantage is that in the attempt to find the optimal solution, the time taken to determine it will be an exponential function of the number of inputs (such as the number of landing calls) to the search system. To minimise the effects of this problem, various solutions have been suggested, one of which is to consider sub-optimal solutions as the complexity of the problem increases [Korf 1988, Rich and Knight 1991].

Genetic algorithms (GAs) are search techniques based on the mechanics of natural selection. The central theme of research into genetic algorithms is *robustness*, its protagonists claim that results are both more efficient (an investigation of the entire search space is not necessary) and more effective (results can be produced more quickly than those for other search techniques) [Mulvaney and Sillitoe 1992].

The pure implementation of the GA merely compares the quality of candidate solutions, requiring no knowledge of the problem itself to guide the search and no estimation of the distance to the desired goal. However, a number of researchers has developed modified genetic searches which either use heuristics to determine promising initial states or estimate distances to guide the direction of search.

The operation of the *pure* genetic algorithm cycle is outlined below.

- Select at random a set of initial states from which searching is to begin.
- Calculate a *state evaluation function* for each of the states. According to the values returned by the state evaluation function, the more promising states are kept and the less promising are destroyed.
- The remaining states are individually split and collectively combined to form new states for inclusion in the next cycle.
- Additional new states are generated at random by mutation for inclusion in the next cycle.

Montana *et al.*[1998] used GAs for 'complex real-time scheduling'. The scheduler described operated on a continually changing field service problem which was sampled at a period of 10 minutes The scheduler automatically assigned service calls to field engineers and conflicting preferences often resulted in one or more service calls not being satisfied. The scheduler is able to balance conflicting demands and alert the human operator to fix the conflict or override decisions. The authors demonstrated the success of the approach by presenting early results which show 60% to 100% resource efficiency improvements

3.1.2 Real-time search techniques

Real time search normally uses a limited *search horizon*, that is, only a certain number of alternative solutions are considered in order to ensure the search ends within a predetermined time limit. Such an approach will be sub-optimal, as the action will be performed based on incomplete information. The two most widely found real-time search techniques are RTA* and alpha-pruning.

RTA* is used for interleaving planning and execution [Korf 1990]. RTA* commits to real world action every k seconds, where the value of k depends on the depth of the search horizon Each time RTA* carries out an action, it restarts the search from that point For each move, the $f=g+h$ value of each neighbour of the current state is determined, and the problem solver moves to the state with the minimum value. The second best value, that is, the best among the remaining alternatives, is stored with the previous state. This represents the h value of the previous state from the perspective of the new current state. These operations are repeated until a goal is reached. To determine the h value of the neighbouring state, if it has previously been visited, the stored value is used, otherwise the *heuristic evaluator* is called. The heuristic evaluator normally employs a *minimum lookahead algorithm* in conjunction with *alpha-pruning*. The minimum lookahead algorithm searches forward from the current state to fixed depth determined by the informational or the computational resources available. Finally, a single move is made in the direction of the immediate child of the current state with the minimum value [Korf 1988, Rich and Knight 1991].

Alpha-pruning uses a *plausible move generator* to select possible moves. The plausible move generator is a heuristic procedure which expands those nodes most likely to succeed and each plausible option is evaluated by another heuristic function, known as the *state evaluation function*, in order to rank them. The values calculated by the state evaluation function are propagated up the search tree and the minimum merit is substituted for the parent merit figure. At this point, previously unexpanded nodes may become more attractive and so expansion of these nodes is begun in preference to the current direction. Alpha-pruning extends this approach by abandoning partial solutions which are clearly worse than already-known solutions and so increasing the efficiency of the search [Mulvaney and Sillitoe 1992]. If α is set to the cost of the first frontier node, then pruning is performed on all generated frontier nodes (and their children) with costs no less than the value of α. Such pruning requires that the cost function does not decrease along any path away from the initial state [Korf 1988].

The quality of the heuristic value effects the speed in which a solution can be found. The algorithm, however, can be used effectively even in the absence of a heuristic function, for example, when $h=0$. Minimin lookahead search with alpha pruning is an algorithm for evaluating the immediate children of the current node. The algorithm is run in simulation mode until the best child node is identified. The results are generally considered to be more accurate than for example when $h=0$ but the heuristic function is computationally more expensive. RTA* provides the next sequence of moves by allowing intelligent backtracking while preventing infinite loops [Korf 1988].

3.1.3 Reinforcement learning

Reinforcement learning (RL) involves learning the mapping of inputs to actions so as to maximise a numerical reward signal. In unsupervised learning, an agent learns from experience generally using trial and error search and delayed reward, whereas supervised learning is performed from examples provided by some knowledgeable external supervisor. Over a period of time, the RL approach progressively learns to favour the most suitable actions [Sutton and Barto 1998, Singh *et al.* 1997].

As in the real-time search methods, RL attempts to solve the lack of knowledge of the environment by providing feedback (reward) from the world state. This agent uses a modular architecture which distributes the learning and control task among a set of separate control modules or agents, one for each goal. Each module learns the optimum policy associated with its goal without regard to other current goals. The learning process is performed by producing an action to execute and communicate to the environment. The action is based either on the current state alone or on the current state combined with a set of possible actions and the environment responds to the action by providing a reward, which is passed back to the agent. The statistics governing the transition function and the reward are normally unknown and therefore the module cannot compute the optimal policy directly, but must explore its environment and learn an optimal policy by trial and error. Information from each module is merged to determine the policy for the combined task [Whitehead *et al.* 1993, Crites and Barto 1996].

A neural network is often used to store the reward value function. The action-value reward function network normally needs to be trained over an extended period of time before it gains the necessary experience and converges to a near optimal solution. Unlike other learning methods, for example GA, where unpromising solutions are pruned, RL explores such solutions and is so able to learn about suitable actions in some states which have high reward value [Singh *et al* 1997, Sutton and Barto 1998]

3.1.4 Neural networks (NN)

As the complexity of the problem increases, the lookup tables used to represent the state space become more inefficient with more memory needed to save both the heuristics and the reward values for the states that are likely to be visited during the control process. In addition, lookup tables provide no mechanism for generalising (or interpolating) neighbouring states, which can result in slow learning To achieve generalisation, neural networks are often used [Whitehead *et al* in Connell and Mahadevan 1993, Crites and Barto 1996]. A NN is a system loosely based on the interconnection of neurons in brains which is able to model and predict real-world situations by learning by example.

The development of NN systems is radically different from that of other techniques, as no programming is required. The philosophy is that the network is taught how to carry out its task by being presented with representative examples and their correct interpretation (termed the *training set*). The network can then be tested by applying further examples before being applied to live cases [Mulvaney and Sillitoe 1992].

The apparent simplicity of implementation hides a number of difficulties [Mulvaney and Sillitoe 1992].

- The size and quality of the training set are important in determining the performance of the network, but no fixed methods exist to help determine an appropriate set of examples

- There is a number of types of NN and a number of different learning methods. It is important that the appropriate structure is selected for a particular application.
- There are no strict guidelines which dictate the number of neurons required to implement a particular solution. If the network is too small then the system may fail to solve all possible examples, too large a network may cause stability problems

However, in many applications the difficulties are outweighed by the benefits and there are a number of applications in which *trained* NNs have been applied to real industrial problems, such as signal processing, complex pattern recognition and classification of tasks [Boullart *et al.* 1992] However, allowing the NN to continue training while it is connected to a real system is not commonly accepted practice due to the aforementioned stability problems [Mulvaney and Sillitoe 1992]

3.1.5 Fuzzy logic

The method adopted for the representation of the world state, whether general or domain specific, is extremely important in the search learning methods discussed above Fuzzy logic allows not only the generalisation of statistical data and the description of probability distributions, but also a means of representing expert system type rules and a method for combining probabilistic data. Fuzzy logic is an extension to normal logic which attempts to mirror the way people think about the problems they wish to solve. In the same way that people are able to respond to imprecise terminology such as "large", "not very large", or "little", fuzzy systems are designed to bring meaning to these terms, at least in the context of the application under development [Mulvaney and Sillitoe 1992]. However, the appropriate grade of membership of a particular variable is not always clear in practical systems.

3.2 Intelligent lift scheduling systems

This section gives a general introduction for the use of artificial intelligence techniques in lift scheduling systems. The section defines the main components of an intelligent real-time lift scheduling system, and gives examples of the work described in the literature which uses some of the AI methods described in the previous section.

It is the task of the lift scheduler to select which of the assignments is most suitable in order to achieve specific objectives. A lift only becomes irrevocably assigned to a call as it decelerates in order to stop at an assigned floor; at any other time the lift can be reallocated as the demands on the scheduler change.

In modern busy high-rise buildings a high performance lift scheduling system is essential to respond to the increasing demand for the rapid transportation of passengers within the building without excessive delays. Many lift companies already use AI in their lift control systems to minimise passenger waiting time and to improve lift assignment in different traffic modes (for example up peak traffic), to detect the beginning and the end of peak periods, and in lift car parking policies.

Artificial intelligence offers new possibilities for traffic forecasting and for optimising landing call allocation. It can be used to generate the most favourable group control parameters in response to time-varying traffic conditions. In order to determine the rules to be applied, a precise identification of the traffic is required, which unfortunately is difficult to achieve during transitions in the nature of traffic and in specific traffic conditions such as a sudden increase in demand at one or more intermediate floors [Chenais and Weinburger 1992]. As was found in the survey of conventional lift systems in the previous chapter, the conventional multi-algorithmic lift supervisory control system suffers from having little flexibility to adapt smoothly to sudden traffic intensity changes Modern lift systems need to be able to adapt to different goals, strategies and changes in building use. The use of AI methods in lift control can allow different goals to be dynamically weighted as the need changes. These goals may include:

- total travelling time,
- average waiting time,
- maximum waiting time;
- car loading ratio and total transportation capacity;
- energy consumption

In addition, it may be possible to reduce the complexity of determining the optimum route to the goal state by automatically identifying *sub-goals*, examples of which are landing priorities and parking policies

In order to satisfy a variety of goals and dynamically adapt to changes in traffic patterns, an intelligent monitoring system may be required to assist the real time scheduler. The monitoring system is responsible for analysing the domain knowledge available about the building and the traffic, and providing the scheduler with the following

- The strategy or the objectives that the scheduler must follow
- Long term prediction, including future traffic flow patterns (for example, during the Friday down peak period) which allow the monitoring system to adopt the most appropriate strategy. The operation of such a prediction module may result in strategy selection occurring shortly before it is normally required, thereby allowing lifts to adopt a suitable distribution in preparation for the change [Mulvaney and Sillitoe 1992].
- Short term prediction of the traffic type. For example, the most probable landing call or car call for the next state, the number of passengers waiting, the most popular floors, appropriate priority floors and the time taken to load and unload lifts [Mulvaney and Sillitoe 1992].
- An evaluation for the current state which helps the scheduler to correct its action, and suggest alternative solutions.
- Parking policy control to complement the real scheduler and maintain an even distribution of lifts among popular floors.
- Lift system status

Figure 3.1 shows an example of an intelligent lift scheduling system of structure typical of that found in the literature. The intelligent lift scheduling system is divided in to two main modules. The *on line* module which is responsible for providing an immediate response and the *off line* monitoring module in which the lift traffic is analysed and objectives are defined precisely in order to allow better rule selection. The separation of the definition of the objectives from the reminder of the system allows greater flexibility of the configuration. The monitoring system may also provide real time calculation (for example current passenger arrival rate) and lift system status definition (for example door status) before each scheduling cycle.

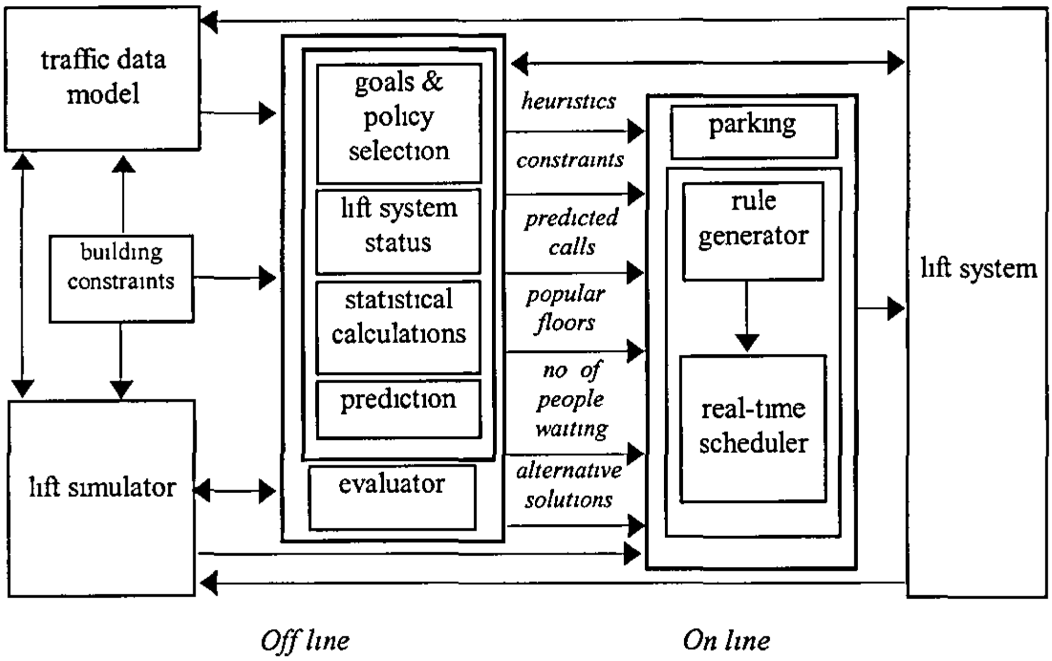


Figure 3.1 An example of an intelligent lift scheduling system.

3.2.1 Examples of lift monitoring systems

The monitoring system provides the scheduler with the information required to determine the strategy or the objectives that the scheduler must follow. This section describes some of the methods used in the implementation of the monitoring systems found in the surveyed literature.

To achieve an optimal call allocation strategy, it would be necessary to know completely the current and future passenger traffic flows. An intelligent lift scheduler uses the assistance of a monitoring system which analyses the daily traffic data of the lift system and compares it with previous analysed data in order to identify general trends in the traffic density and distribution. Such information is then used to help in making decisions based on future expectations. Most of this information can be communicated *off-line*, that is, it need not be synchronised with the immediate real-time scheduling solution. Some examples of monitoring systems are listed below.

- Neural networks have been used for traffic pattern recognition and data reduction, and also for learning suitable scheduling strategies [Imasaki *et al.* 1991, Chenais and Weinberger 1992, Kubo *et al* 1995, Crites and Barto 1996, Crites and Barto 1998]
- Statistical analysis and the calculation of principal parameters have been used to identify the nature of passenger traffic which tends to follow a certain pattern that characterises a certain building. An example of such a statistical technique is exponential smoothing to assess passenger arrival and exit rate [Chenais and Weinberger 1992, Thangavelu and Pullela 1992, Ujihara and Amano 1994, Siikonen 1997].
- Fuzzy logic has been used to represent combinations of statistical information and expert system knowledge in the form of rules [Imasaki *et al* 1991, Tobita 1991, Kubo *et al* 1995, Siikonen 1997].
- Genetic algorithms have been applied to parameter tuning in multi-objective systems The parameters are tuned according to the changing environment and strategy selection [Tobita 1996].

The monitoring system is normally implemented as a set of rules which is used by the scheduler in the calculation of the cost function required for the solution path.

In an off-line prediction system produced by Otis [Thangavelu and Pullela 1992, Thangavelu Aug. 31, 1993], the learning method had the ability to adapt to changes in the building's operational characteristics The authors used a prediction methodology to produce models capable of learning the best prediction factors for controlling the lift system. The prediction simulation is part of an automated learning system which also includes both an evaluation system and a learning system. The resulting system is an adaptive controller which is responsive to variations in traffic patterns in a building, and is able to select the models and parameters to be used in the traffic prediction system. The prediction results are then used to aid the following operations:

- to detect the start and the end of a peak traffic period,
- to intelligently group floors into sectors, ideally so that each sector has equal traffic volume;
- to determine the number of passengers waiting behind a landing call and assign a higher priority to those floors where larger numbers of arriving passengers are predicted;
- to vary door dwell time at each floor based on the predicted number of people boarding and de-boarding at that floor.

The methodology provides the capability to adapt to changes in building operation by estimating a number of characteristics. These include computing the best prediction model using a smoothing algorithm, finding the best factoring coefficients for combining real-time and historical predictors, determining the best data and prediction time interval lengths to be used and calculating the optimum number of lookahead intervals (for real-time predictions) or lookback intervals (for historic predictions) to the extent applicable to the prediction model. The authors describe an example of data which recorded landing calls at one minute intervals over a period of 20 days. Simulation experiments are conducted separately for boarding and de-boarding at each floor. Experiments were conducted separately for up peak, down peak, mid day and other periods, and from these, the results for several sets of experiments were derived, each set being applicable to one floor, one traffic pattern, one time period, different prediction models, data and prediction intervals, prediction coefficients, lookback days and lookahead intervals. The simulation results were analysed and suitable statistics obtained, for example the sum of the square of the prediction errors of each interval was computed for the prediction period for that day. Those results with the smallest associated error values were selected. The combination of prediction methodology and prediction parameters was applied over a number of subsequent days for each relevant period The experiment was run approximately once each week in order to determine and learn the latest best applicable models and parameters. This method was claimed to provide better prediction and the lift system using it showed an improved performance with respect to earlier systems, including a system that used a fixed number of floors for sectoring Although the patents described the prediction method used as an AI method, statistical forecasting methods were used, such as moving average and exponential smoothing.

Siikonen and Leppala [1991] describe a method used by Kone to predict traffic situations based on statistical forecasts. Exponential functions are used to represent passenger arrival and exit rates at each floor, which are stored in a database. The local and total volumes of the passenger traffic are based on continuously measured lift loads, photocell signals and both the call and lift status. The traffic patterns are recognised according to the traffic volume, intensity and its distribution in the building Values for incoming traffic from entrance floors, outgoing traffic to entrance floors and inter-floor traffic components are calculated from the traffic forecasts. Rule based programming with fuzzy logic is utilised to determine traffic types and, as a result, 25 different traffic types were defined, in a way that shows not only the main peak traffic periods (such as up peak) but traffic patterns such as heavy incoming and heavy outgoing demands were identified. This method is an example of using fuzzy logic to represent combinations of statistical information regarding traffic volume and intensity to define traffic transitions, but also to prevent abrupt changes in performance which may result from altering the scheduler's goals. The knowledge of traffic type can be used in conjunction with the real-time scheduler to trigger the appropriate strategy and to adapt to the change in traffic patterns, for example identifying priority floors and varying door dwell time according

Hitachi [Tobita *et al* 1991] has produced an off-line strategy selection module in which the client chooses the preferred strategy to follow that day or during a certain period. Lift assignments are based on the calculated cost found by evaluation function for each lift car. The cost is tuned using strategy selection parameters to suit the client requirement. The strategy model is implemented using fuzzy logic, an expert system and a lift simulator. Fuzzy logic is used to provide priority rates, transform coefficients based on the user's specified objectives and both building and lift specific information. An example of a fuzzy rule is, IF *traffic is medium* THEN *a transform coefficient of the number of* *passengers is small* These data are supplied to a control method decider (strategy selector). The control method decider consists of an expert system and a simulator. The simulator is used to simulate lift movements and return the results of each control method to the multi-target decision-making unit that evaluates the results and produces the chosen policy. The simulation must be carried out for a long sequence to be able to follow the traffic flow changes for 1 or 2 day periods. The chosen policy is displayed along with the simulation results and the elevator operator (a person) decides whether a satisfactory result has been produced. If the result is satisfactory, the scheduling method is sent to the lift supervisory controller, otherwise the procedure is carried out again to determine an alternative strategy. Tobita *et al* [1996] briefly discussed the use of hill climbing method with the simulator as part of the tuning and evaluation function The authors explained that for a multi-objective lift system the search takes longer and heuristics are needed to limit the search Rich and Knight [1991] define the *hill climbing* search as a variant of generate-and-test in which feedback from the test procedure is used to help the generator decide which direction to move in the search space. Heuristics are used to evaluate each state and the best state is chosen. Hill climbing differs from A* search in that it rejects all other states which means they will never be considered. Another multi-objective lift control system developed by Hitachi is described by Tobita [1996] and uses genetic algorithms to change the control settings according to floor utilisation. The method is argued to be different from other parameter tuning methods such as fuzzy logic and NNs which focus on the entire building In the Hitachi system, the number of tuning parameters is around 50, while only 1 parameter was used with hill climbing and 4 parameters were used in the heuristic search. The tuning parameters are evaluated using a lift simulator. The authors claim that the tuning process is performed on-line but there is no indications given of how long it takes to perform the search The latter method has an advantage over the former method in that it is less dependent on human efforts. Both methods are examples of how the operations of strategy, goal selection and goal evaluation are often separated from the real-time scheduler.

Kim *et al.* [1995] describe a two stage fuzzy inference mechanism used to determine the tuning parameter in a multi-objective system where average waiting time, long wait probability and power consumption determine the cost evaluation function The first stage includes calculating the degree of importance of each of the optimisation objective factors according to the traffic pattern (up or down) and the second stage adjusts the weight according to the evaluation function The authors claim that the system performance showed improvements with respect to conventional methods due to the ability to dynamically change parameters rather than leaving these pre-set by the building operator and the results of simulated traffic conditions.

Another lift control system using fuzzy logic is described by Kim *et al* [1998]. Fuzzy logic is used for traffic classification, strategy selection and lift assignment. Eight different traffic types are defined for different times of the day. The performance measures (optimisation goals) are average waiting time, the percentage of landing call waiting times greater than 60s and the number of lifts moving in a unit time is used as an energy consumption measure. The lift system manager is required to set the importance levels of these optimisation goals according to the time of the day and traffic type. The definition of traffic type depends on the fuzzy inference (based on traffic features and the time of the day) and a predefined constant that protects against oscillation between traffic modes, however no information is provided on how this value is calculated The cost evaluation function used in lift assignment is determined using a fuzzy model based on the assigned importance level and features such as landing call waiting time and maximum waiting time, hence increasing the tuning capability to reflect changes in demand. The authors claim a performance improvement over the results produced by other lift systems (for example the lift system given in [Kim *et al* 1995]). A special emphasis is given to the manager's facility for changing the optimisation factors, however a method such as that used by [Siikonen 1997] is less dependent on the need for human intervention, where the scheduler mode of operation dynamically changes according to the traffic type.

Toshiba [Imasakii *et al* 1991, Kubo *et al* 1995, Nakai *et al.* 1995, Imasaki *et al* 1995] incorporate a fuzzy neural network in the implementation of a forecasting model and a performance tuning function that utilises the forecasting model in order to search for the optimal control parameters which give the best system performance in the present traffic condition. The fuzzy neural network is shown in Figure 3.2, where each if-part network stores membership functions for an input variable (control and traffic information) and each then-part network stores an equation regarding the output (predicted waiting time) of the fuzzy rules. The reasoning network stores data regarding logical relations between the condition part and the conclusion part of the fuzzy rules The predicted passenger waiting time is then used to choose the optimal control variable used by the lift scheduler A simulator was developed to verify the facility and support the design of the forecasting model The predicted and actual waiting time distributions were similar and the system showed a 10% improvement in long waiting rate when using the adaptive parameter tuning model. The authors state that further investigation and analysis of the field operation are planned to validate the system and to develop a system which has better adaptation to user needs. A development of this work could be to use NNs in the implementation of a completely autonomous lift controller by tuning control parameters according to traffic flow.

Mitsubishi [Amano and Yamazaki 1995] used NNs to identify the change in traffic flow in real-time, and its inputs are obtained from land sensors to estimate the number of passengers entering and leaving lifts. The gathered traffic data are used for training the NN at regular intervals in order to classify traffic in response to the number of passengers entering and departing the lifts in either direction. The scheduler is thus able to adapt its policy according to the identified traffic type. Simulation results showed a 10% reduction in passenger waiting time and a 20% reduction in long waiting rates in comparison with a conventional lift control system. The above paper is a good application example of the use of a NN in real-time traffic classification. However, it is not known whether this system has been used in practice.

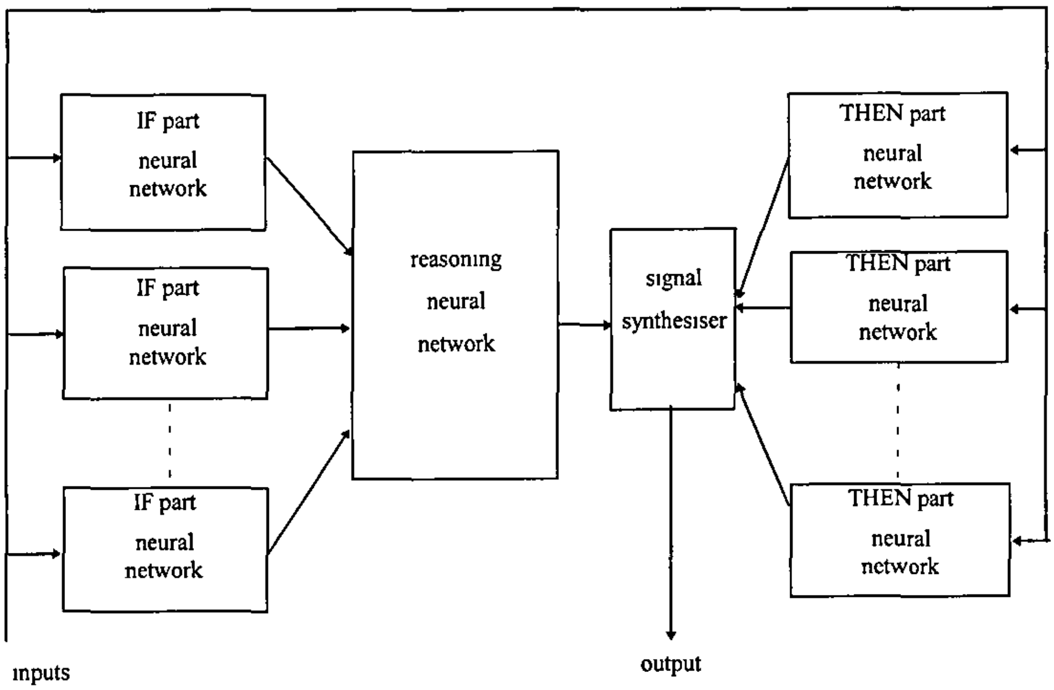


Figure 3.2 Fuzzy neural network configuration [Imasakii *et al.* 1991].

Mitsubishi [Amano and Yamazaki 1995] used NNs to identify the change in traffic flow in real-time, and its inputs are obtained from land sensors to estimate the number of passengers entering and leaving lifts. The gathered traffic data are used for training the NN at regular intervals in order to classify traffic in response to the number of passengers entering and departing the lifts in either direction The scheduler is thus able to adapt its policy according to the identified traffic type. Simulation results showed a 10% reduction in passenger waiting time and a 20% reduction in long waiting rates in comparison with a conventional lift control system. The above paper is a good application example of the use of a NN in real-time traffic classification. However, it is not known whether this system has been used in practice.

Several methods of estimating passenger entry and exit rates have been patented, including Sattar *et al.* [1993] and Sirag [1993] who used weight, car calls, landing calls and lift car stop signals as inputs to a fuzzy logic system. Siikonen [1997] used photocell activations and load sensor readings as inputs to a system designed to count passengers. Computer vision was also used by Mehta *et al.* [1995]. The above systems would require expensive modification to a system currently installed, and therefore a method which is able to derive the number of passengers from information normally available in the lift systems is preferred. For example, using the formula to calculate RTT (Section 2.2 3) and where the number of stops a lift makes during RTT is known could be used to calculate the number of passengers. However, in practice, the values of several parameters are not actually known, such as the relation between contract capacity and the number of passengers and the percentage of a floor currently populated [So *et al.* 1994].

Chenais and Weinburger [1992] describe a lift simulator developed for Schindler which incorporated a traffic module to aid in the simulation of car journeys and passenger movements and to provide an evaluation module with factors concerned with passenger

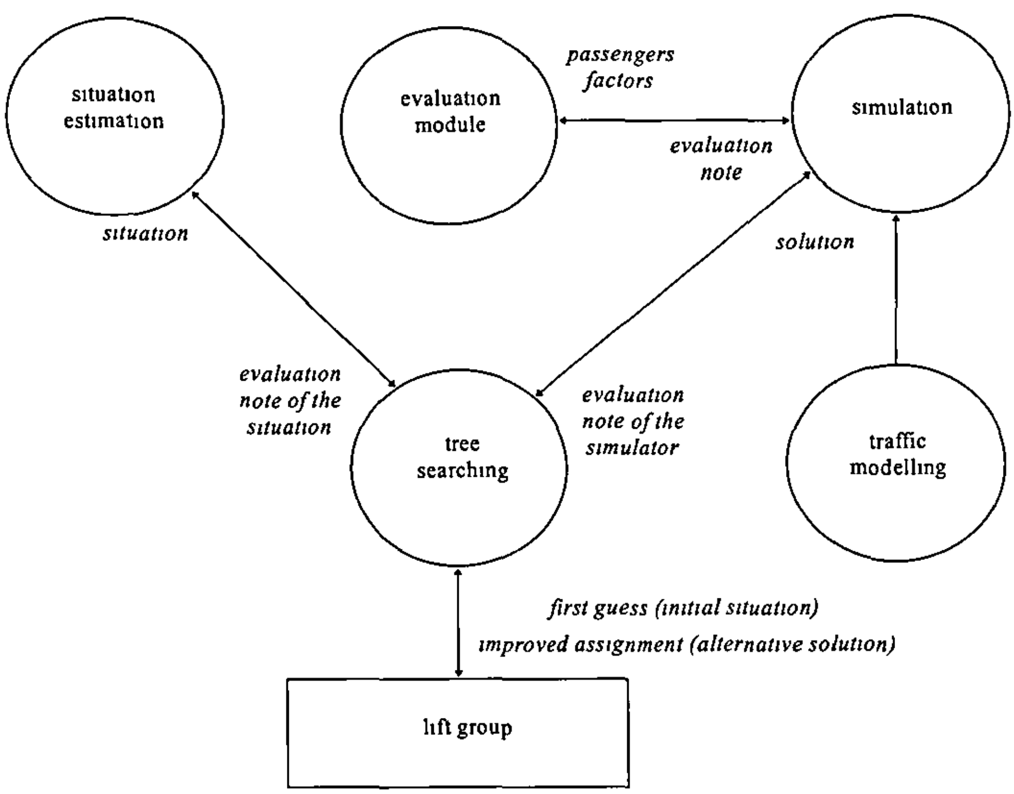


Figure 3.3 Lift system described by [Chenais and Weinburger 1992].

traffic (for example passenger average waiting time), Figure 3.3. The traffic module supplied the simulator with the likely number of passenger waiting at a floor and their possible destinations. The evaluation results, further assisted by the situation module, aid the real time scheduler in choosing an alternative solution (scheduling strategy). The real time scheduler used a search tree which will be further explained in the next section The work emphasised the need for a clear identification of objectives and the use of evaluation to obtain system optimisation rather than depending on traffic pattern identification. However, no comparison results were given.

In AI, prediction is normally considered as a part of the general learning problem. Many learning techniques are still in the research stage, but there is a number of techniques which are sufficiently mature to consider in the context of lift scheduling [Mulvaney and Sillitoe 1992].

A general concern when implementing any prediction system is to determine which events are actually worthwhile predicting. For example, the prediction of landing calls and the distribution of car calls may be far more accurate during the *up-peak* period than during periods of low traffic volume During the latter periods, it is even possible that scheduling performance may be impaired by using the additional data, and hence the intelligent monitoring system will need to be sensitive to which strategy is currently being operated by the scheduler so that prediction can be used appropriately [Mulvaney and Sillitoe 1992].

The above research examples define the main components of the intelligent lift scheduling system shown in Figure 3.1. Parts of the lift monitoring system may need to operate in real time, such as real-time prediction and evaluation. A lift simulator may also be used as part of the evaluation module. The real-time scheduler may be complemented by other policies such as parking and up peak policies which are triggered according to traffic flow. The next section gives examples of the real-time scheduler for this system.

3.2.2 Examples of Real-time lift schedulers

Dynamic scheduling schemes continuously observe the traffic change and produce a new schedule either on a timely basis or it may be triggered by an event, for example a new landing call or car call. In general, the lift scheduler will need to follow common rules, such as the lift must not change direction until all passengers currently in the lift are transported to their destinations. One of the most critical measures of a scheduler's performance is its action during up peak traffic. The scheduler performance can be assessed both in terms of the *quality of service* measured by the frequency of lift car arrivals at the terminal floor and indicated by the average value of either the passenger waiting time or the passenger journey time and the *quantity of service* measured by lift capacity consumed to serve a specific set of calls and indicated by the total journey times of all the cars. The scheduler must balance this effect against the corresponding reduction in the number of passengers per car as the trip time is reduced, and its effect on the overall waiting time [Barney and dos Santos 1985].

There are different methods by which the prediction data can be used to simplify and speed the task of finding the best assignment. Some schedulers consider only new landing calls at each schedule [Tobita *et al.* 1991, Siikonen 1997]; the effect of the new assignment on the old assignment is calculated but it is difficult to forecast the arrival time and trip time. The ACA (Adaptive Call Allocation) [Barney and dos Santos 1985] introduced floor destination buttons to overcome this problem while Siikonen [1997] used simulation of future events according to measured traffic flow. In some real-time schedulers, the first priority is to produce a schedule promptly in response to an event, and in such cases the scheduler is partly an event driven scheduler and partly a time base scheduler which gives the overall global evaluation of the current state and produces a schedule or a correction action [Hadavi 1994].

If only the passenger waiting time is considered as a performance objective, then the evaluation function is usually measured by calculating how long a landing call will be waiting if answered by a lift, that is the time interval between pressing the landing call button and the time when the assigned lift opens its door [Siikonen 1997]. Kone [Siikonen 1997] modified the usual collective control scheduling principle by introducing dynamic optimisation. The resulting system was an event driven scheduler using an optimal routing algorithm to reduce the number of route combinations in the search for a schedule in steady state For *N* calls, the algorithm starts its decision tree* with landing call *N*, followed by *N-1* calls. Landing call *N* is the landing call furthest in the lift travel direction and landing calls are allocated clockwise from floor *N*. The allocation starts from the lowest landing call and continues to the highest landing call, and the lifts travel around the building answering up landing calls and then down landing calls. The assignment of call *N* remains unchanged while assigning *N-1*, but the effect of assigning landing call *N-1* on landing call *N* is considered when the cost function is calculated. Without forecasting future landing calls, the algorithm sometimes leads to local optimum, because landing call times at the end of the route may change as new landing calls arrive. The author introduced a method of future event prediction using the simulation of future states in calculating the cost function to evaluate the consequences of an assignment. In the beginning, the intermediate stops of the lift are not known, but after the scheduling cycle they become fixed. The allocation then starts with the new intermediate stop information, and the cycle is iterated until the allocations to the cars remain unchanged The average waiting time was used as an optimisation goal and the author claims that simulation results have led to an improvement with respect to the ACA scheduler However, the author indicates that the algorithm still suffers from the fact that it does not always find the global optimum for route combinations. The method appears to have been tested only in simulation as the author states that the simulation of future events requires too much computation time for a real-time system. Considering new landing calls only is one way of reducing the search space However, it is difficult to simulate all future consequences Simulation may assist in the evaluation of the current state as by simulating future state as shown in the previous section

The Mitsubishi AI-200 [Ujihara and Amano 1994] also uses the principle of giving priority to the assignment of new landing calls. It incorporates an expert system using fuzzy logic rules for lift assignment with multiple strategy selection. The scheduler applies a set of rules which take into account estimated times of arrival of lifts and the floors where landing calls are expected. An example of a fuzzy rule found in AI-200 is:

* Representation for concepts used for data classifications [Rich and Knight 1991]

IF *there is a landing call registered on an upper floor* AND *there are a large number of cars ascending towards the upper floors* THEN *assign one of the ascending cars on the basis of estimated time of arrival*. This approach prevents all lifts from being assigned to upper floors. The terminal floor is supplied with destination buttons and each lift has a destination screen showing the areas each lift service. This method is used instead of zoning as the authors explain that zoning may have unsatisfactory results when one of the zones is much busier than others. No information was given on how the future state was predicted nor how the multiple strategy selection was achieved.

A version of A* search was used by Mehta *et al* [1994] and Mehta *et al.* [1995] in a lift scheduler which had the advantage of estimating the number of passengers waiting behind a landing call by using a computer vision system for counting passengers. The cost evaluation function was chosen to reflect the speed with which the passengers were transported and the search was directed towards the highest calculated cost. The search tree branched according to three lift moving states: up, down and stop. The cost calculations were dependent on the passenger's direction of travel and on whether the passengers were waiting at a landing or in a lift The best move is decided using a look-ahead search. The depth of the look-ahead search depended on the number of lifts and floors in the building. Limited look-ahead was preferred to exhaustive search as future assignments may need to be altered in response to changing traffic conditions. The results obtained using single step look-ahead search was compared to a conventional (collective control) lift control system and a performance improvement was found, especially during periods of random traffic. The algorithm showed a performance similar to that of a conventional lift system during up peak, as in such traffic conditions the number of passengers waiting at the terminal floor has little impact on lift movements, and the behaviour is likely to be similar to that of the collective method In order to extend the look-ahead search without incurring a significant computational overhead, the authors suggested running the system to learn patterns using pattern classification and clustering techniques, so that future decisions can be made more quickly. The pattern classifier achieved an accuracy of decision of around 87%. This work shows the advantage of a vision system for passenger identification to improve data quality. However, it is not described in detail how the A* search was implemented and, in particular, whether limiting the algorithm to a single step look-ahead search affects its optimality.

Schindler [Chenais and Weinberger 1992] used a tree search in which each branch of the tree corresponded to the assignment of a lift to serve a landing call. The search provides an initial solution following the action of a collection of rules drawn up from previous experience. The search for alternative solutions is carried out using *alpha pruning*, Section 3.1.2, and is continued for a period of time determined by the configuration of car and landing calls. The solution is moderated by an analysis of the configuration resulting from the simulation, for example the solution would be rejected if the lifts were tightly grouped. This approach is one example of the use of simulating future events as part of the evaluation process, but no information is presented regarding how the search tree was constructed.

Crites and Barto [Crites and Barto 1996, Singh *et al* 1997, Crites and Barto 1998, Sutton and Barto 1998] used a team of RL agents, each of which is responsible for controlling one lift car. The advantages of the technique are the ability to perform both with and without models of the system and that the team of RL agens can be used on-line as well as off-line, focusing computation on areas of the state space that are likely be visited during actual control. Omniscient reinforcement is used during the design stage to make use of the knowledge available with the aid of a lift simulator, such as how long each passenger has been waiting. Once the controller is installed in a real system, any fine tuning must be done without the benefit of this extra knowledge. A NN is used to save the reward values. The authors claim that the results surpass the best of the heuristic lift control algorithms, for example a sectoring based algorithm The following points arise from this work

- It is not clear which constraints are required for generating the action-value reward function The system was tested for down peak and some of the rules used are only suitable for the down peak control policy, such as an empty lift can not park at an empty floor and a lift should prefer travelling in the up direction to avoid bunching at the bottom floors. The constraints are used to limit the number of actions an agent is able to perform.

- In a decentralised RL system, each agent faces added stochasticity and nonstationarity because its environment contains other learning agents. Cooperation has to be learned indirectly using the global reinforcement reward signal. Although the work discusses the use of a parallel implementation of an RL system in which the agents use a central set of shared networks allowing them to learn from each other's experiences, this normally results in the agents learning an identical policy In fully decentralised implementations, the agents have their individual networks, allowing them to specialise their control policy.
- In order to produce the desired performance, the NNs must be trained with a suitable large data set. Training was required for four days of computer time, corresponding to about 60,000 hours of simulated time.
- The way the state was encoded into input units of the NN was found to be critical to the effectiveness of the learning In a decentralised RL system performance degrades as agents receive less state information
- The weights of the NNs can become unstable, their magnitudes increasing without bound This is likely to happen when the learning algorithms make updates that are too large. This can happen when the learning rate is too high, the network inputs are too large (for example in very heavy traffic situations), or both.
- Assumptions were made for lift system dynamics such as travel and load time rather than using actual values.

To maintain a global view, the lift scheduler needs to keep close contact with changes in the environment. One way in which this can be achieved is by implementing an off-line scheduler which aids the real-time scheduler by providing complete schedules as alternative solutions. A scheduler can also learn from experiences of earlier decisions by providing feedback of updated information from the environment in response to an action. It is apparent that both the quality and quantity of data provided are important in achieving good scheduling performance. Such data are required in order to generate an accurate representation of the current state and to provide suitable information to be used in the production of heuristics for the prediction of future events.

3.3 Conclusions

In this chapter a survey of modern lift systems has been given to help identify the main issues relating to lift scheduler performance. Although a suitable real-time scheduling algorithm needs to be chosen so that lifts can respond to passenger requests in such a way as to transport them quickly and efficiently to their destinations, its performance is diminished by the unpredictability of traffic patterns. This missing knowledge makes the task of providing optimum real-time decisions by the scheduler impossible in practice. However, the better the quality of the data provided to the real-time scheduler, the better will be its performance. This is recognised in the literature, in which the majority of lift schedulers are implemented as combinations of a real-time scheduler and a lift monitoring system. The real-time scheduler needs to be able to respond within one calculation cycle to landing calls in order to achieve minimum possible waiting time Perhaps due to commercial constraints, little detail regarding the internal working of real-time schedulers is available in the literature [Crites and Barto 1998]. However, one general method that appears to be becoming more common in modern lift schedulers is the incorporation of sophisticated assignment algorithms rather than the conventional approach of assigning lifts to zones, in which each lift travels within its zone answering the nearest landing call in its travelling direction. Examples of new algorithms were found for heuristic search and RL. In the case of RL, recent results appear promising and the incorporation of predicted knowledge into the decision-making operations of individual RL agents might enhance real time performance. However, the reinforcement learning system would need to be trained in advance on known simulated data with full knowledge of each passenger waiting time. Most of the literature concentrates on the development of monitoring systems to supply relevant data to the real-time scheduler, such as traffic identification and classification, counting the numbers of passengers and parameter tuning to achieve multiple objectives. Improving the quality of data reduces the assumptions that need to be taken and allows the scheduler to produce better informed and more appropriate decisions.

It is apparent that to achieve an optimum call allocation strategy, it would be necessary to know not only the current passenger traffic flows but also those that will occur in the near future. Clearly, it is only possible in practice to predict future traffic flows and the role of a monitoring system is to support the real time scheduler by supplying timely and relevant information which is derived from the available traffic data.

Based on the findings of the literature search, it is apparent that a combination of a suitable real-time scheduling technique in conjunction with an appropriate monitoring system is likely to prove the most appropriate direction for this research.

Chapter 4

Introduction to the intelligent real-time lift scheduling system

This chapter introduces the characteristics of the proposed intelligent lift control system This includes the need for a lift simulator as an essential part of its development and the need for a realistic passenger traffic model as part of a lift monitoring system which provides relevant information to aid the real-time scheduling system. Figure 4.1 shows the proposed intelligent lift control system.

Traffic analysis is the first step towards building the intelligent lift system, to provide the arrival profile for a building and the statistics of call distribution Section 4.1, Discusses traffic models that are used to drive the lift simulator as well as using them as historical examples for the monitoring system. Section 4.2 discusses the need for the construction of a lift simulation, such that when random calls are issued the model will simulate the lift operation, considering all the possible operation details and restrictions involved.

AI methods offer new possibilities for traffic forecasting and optimal or near-optimal landing call allocation They are used to generate the most favourable control parameters in response to time-varying traffic conditions. Based on the analysis of the lift systems (chapter 2) and modern intelligent approaches used for their control (chapter 3), the conclusion is that there is a need for a lift control algorithm that is adaptive to changes both in the design requirements and in traffic patterns, which provides a real-time response, and is both reliable and stable When choosing a suitable method for the real-time scheduler, the speed, optimality and reliability of the method are important. For example, stability and reliability may become a problem when using neural networks. Section 4 3 discusses why intelligent search was chosen for the real-time scheduler

In scheduling problems, the selected solution will depend on the quality and availability of data provided to the scheduler In particular, inaccurate data or poor estimates of values can cause the scheduler to choose the wrong solution. A monitoring system is required to provide the lift scheduler with information regarding lift status, passenger arrival rate and the detection of peak traffic periods. Section 4.4 introduces the lift monitoring system used in this project.

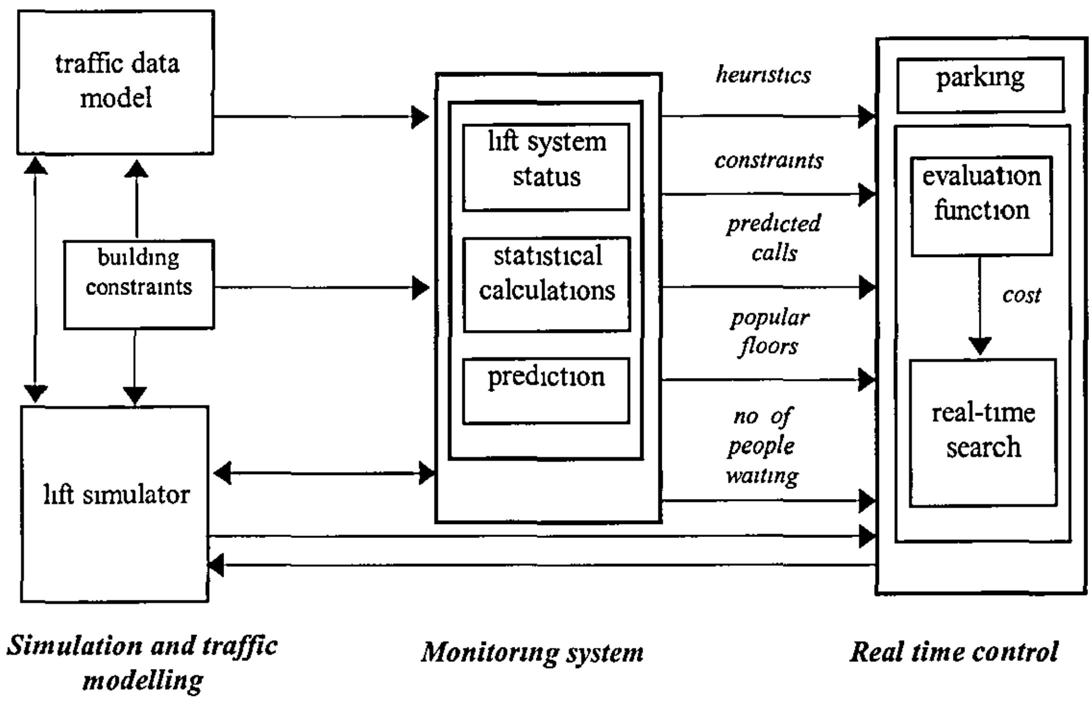


Figure 4.1 Proposed intelligent real-time lift system.

To assess the performance under realistic circumstances, the intelligent lift control system implemented by the author is compared to that of the ETQ scheduler (Section 2.5) which is currently used in the Kodak building [Appendix A], with the comparative average passenger waiting times being taken as the principal performance measure. A software implementation of the ETQ scheduler is also used to verify that the characteristics of the lift simulator conform to those of the actual system. The intelligent lift scheduling system simulation results were performed on a PC with 486/5x86 processor operating at 133MHz. The software is implemented in this work using MATLAB and C language.

4.1 Modelling of passenger movement

The analysis of the distribution of passengers arriving at a given floor requiring to use a lift falls in the domain of queuing systems [Gross and Harris 1974]. The literature indicates that by generating a random number from a suitable Poisson distribution the time of arrival of the next passenger arrival can be estimated [Peters and Mehta 1996]. This implies that the probability distribution of the time intervals between passenger arrivals will be exponential. In the current work, it has been possible to assess such theoretical assumptions as access has been obtained to real lift installations and relevant data have been gathered directly. However, passenger information is not available directly at a lift installation, as only data relating to passenger activities monitored by the lift system can be recorded. Hence, it was necessary to carry out further processing work on these data in order to produce an appropriate model to estimate the distribution of passenger arrivals simulator [Hamdi and Mulvaney 1998]. This model provides the lift simulator with the required daily traffic flow. Car calls are modelled using actual car calls arising from the actual landing calls. In addition, other models of traffic are derived to be used as a historical database for the lift monitoring system.

4.2 Lift Simulator

A lift simulator needs to be a discrete event, fixed-time increment, dynamic, stochastic simulation of a group of lifts [Galpin and Rock 1995]. Lift simulators are developed for two principal purposes, namely to aid the design of new lift installations and to assess the relative performances of alternative scheduling strategies. The current work falls into the second category [Hamdi *et al* 1995].

The first lift simulators were built in hardware; modern software systems bring flexibility to lift simulation. As well as aiding the understanding of the effects on the lift system of different traffic patterns, a lift simulator allows potential customers to view the performance of proposed lift installations. At the design stage, a simulator may aid in the selection of the number of lifts which will provide sufficient handling capacity under the control of a suitable scheduling system, while maintaining efficient building space utilisation. In buildings whose function may alter during their lifetime, a lift simulator enables the development of an adaptable control algorithm to provide flexible efficient transportation under a variety of operating conditions.

A simulation of a lift system was produced by Hummet *et al.* [1978], a software implementation was developed by Lustig [1986] and a discussion of their relevance and use in the development of lift systems can be found in Barney [1988]. Simulation has been used in the planning of lift groups by Siikonen and Leppala in Siikonen [1991] and it was proposed by Siikonen [1997] that the performance of a lift scheduling algorithm can be investigated using a simulation of lift dynamics. Galpin and Rock [1995] produced a prototype simulation tool to assess the need for such a tool by designers of lift systems.

The conventional way of calculating the performance of lift systems is based on probability theory in conjunction with simplifying assumptions. For example, during the up-peak period, most calculations assume evenly populated floors, transportation of the same average load in each lift car from the ground floor and equal interfloor heights However, as explained in Chapter 2, the calculated INT (the average time between successive lift arrivals at the main floor with lifts loaded to any level [Barney and dos Santos 1985]) and handling capacity do not give adequate information for all traffic patterns and a lift simulator is required to produce a more accurate description of system performance under different traffic patterns, building specifications and using different scheduling algorithms. Using a lift simulator, different patterns of traffic can be tested and goals such as minimising waiting or journey time can be monitored to optimise scheduler performance and to produce an efficient building design in terms of serving shops, restaurants, or entrances. For example, simulation results have indicated that there is no direct connection between INT and waiting time [Siikonen 1997]. INT depends on the number of lifts and lift capabilities, while in addition to lift performance, waiting time depends on passenger arrival patterns and, particularly at peak times, the performance of the scheduler.

In Chapter 3, examples were given when rapid data processing by a simulator allowed its use as a part of a traffic monitoring system which observes traffic fluctuations and feeds back an evaluation of the current state to the scheduler or helps in selecting the most suitable scheduling strategy.

4.3 Intelligent real-time lift scheduler

The proposed real-time scheduler will need to be able to respond within one calculation cycle to landing calls in order to achieve minimum possible waiting time [Hamdi and Mulvaney 1997].

Heuristic search techniques are the most suitable to achieve the task of finding suitable paths of sequences that map the initial state to the goal state The efficiency of search algorithms can be measured using the cost of the solution, in this case waiting time [Hamdi and Mulvaney 1997]. One popular algorithm, explained in Section 3.1.1, which can incorporate simple heuristics is the A* search. The disadvantage of A* is that the search grows exponentially with the number of landing calls and so the calculation of a solution may take longer than the cycle time to find the optimum schedule. However, since A* is the fastest algorithm for finding optimal solutions [Korf 1988], this problem may be lessened by providing a good heuristic function and high data quality incorporating load information, door status and landing call waiting time.

Real-time search techniques appear to be suitable for more critical cases, as they use a limited search horizon. That is, only a certain number of alternative solutions are considered in order to ensure the search stops within a predetermined time limit. Since decisions are based on limited information, an initially promising direction may appear less favourable after gathering additional information in the process of exploring it, thus motivating a return to the previous choice point [Hamdi and Mulvaney 1997]. Both RL and RTA* methods (Sections 3.1.2 and 3.1.3) need to maintain close contact with the state environment which reflects on their judgements and actions. However, in the case of RL, the agents build up experience with time which eventually allows them to explore all possible states In order to gain the required experience, the RL system would need to be trained in advance, and for a sufficient simulation time, on known simulated data with full knowledge of each passenger waiting time plus several other stability and reliability issues discussed in Section 3.2.

RTA* is a modified version of A* for real time systems, designed for use when information about the current and future states is incomplete and a decision has to be made in critical time under those circumstances. Rich and Knight [1991] give an example of the task of navigating between rooms in an unfamiliar building. The search horizon is limited by how far the search can progress at any given time. RTA* can make progress towards the goal state without having to plan a complete sequence of solution steps in advance. The idea is to take steps in the physical world in order to see beyond the horizon, despite the fact that the steps may be non-optimal. The key difference between RTA* and A* is that in RTA* the merit of every node is measured relative to the current scheduler state. Hence, the initial state is irrelevant and the g value presents the cost from the current state to the next state rather than from the initial state [Korf 1990]. This algorithm is useful when a single goal is defined, such as a robot navigation, and where small steps can be more useful than continuing to a large depth, thereby following a policy of least commitment.

However, lift systems have well defined search horizons in terms of actual and anticipated landing and car calls. The missing knowledge can be predicted to help in the process of searching for the optimum schedule, and the search result would be optimal in terms of the waiting time for the current state. However, as new calls arrive, the scheduler has to conduct a new complete assignment by search. In lift systems, only the first assignment of each lift is required at the end of each search and the goal is to achieve the minimum cost in terms of waiting time to answer all landing calls. Although the future state of the lift system is not fully defined, the current state is well defined especially when accurate data sources are used. The assignment of any landing call with respect to the assignments of all landing calls must be achieved with respect to the initial state, and the deeper the search the better the resulting assignment.

Another version of A* search is the iterative deepening A* (IDA*) [Korf 1988, Korf 1990, Russell and Norvig 1995] which is similar to depth-first iterative deepening illustrated in Figure 4 2, with the difference being the cut-off criterion is changed from being pre-set depth to a path cut-off when the cost exceeds a predefined value of f. IDA* starts with an initial threshold equal to the heuristic estimate of the distance from the initial state to the goal. Each iteration of the algorithm is a pure depth-first search, cutting off branches when the f value exceeds the threshold. The search continues until all nodes inside the contour for the current f cost are opened, and terminates if the goal state is found. Otherwise, the threshold is increased to the minimum f value that exceeded the previous threshold, and another complete depth search is started from scratch.

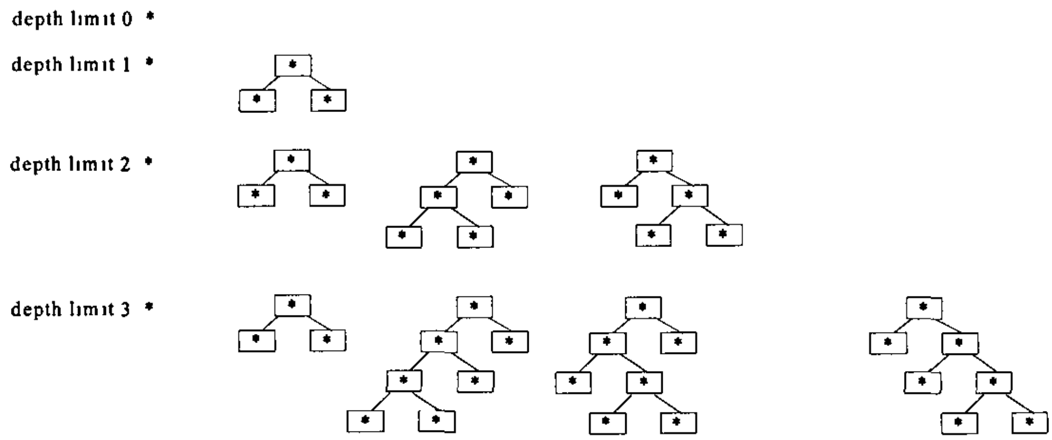


Figure 4.2 Four iterations of Iterative Deepening search on a binary tree [Russell and Norvig 1995].

The main points that characterise IDA* are described below

- Like A*, IDA* is a complete and optimal search. The memory space required is proportional to the longest path explored and with a good heuristic function, f, will only increase two or three times along any solution path.
- Its efficiency is similar to A*. in fact the final iteration of IDA* expands roughly the same number of nodes as in A*.
- Only a comparison operation is needed in order to determine the next state, and as such, IDA* has a smaller calculation overhead compared with A* which needs to perform sorting, insertion and deletion operations.
- According to Korf [1988], empirical results show that even though IDA* generates more nodes than A*, it actually completes a search faster in practice than A*. The reason is that, as explained above, IDA* has less overhead per node
- IDA* generally uses less memory than A*.

- The IDA* algorithm is easy to implement.

IDA* has two disadvantages:

- It can take a long time to compute if the f value changes many times.
- It does not remember repeated nodes.

Provided that the search forms a tree and not a graph, in other words no nodes with the same assignment are generated, and the f value does not change more than two or three times It should be possible to use the IDA* algorithm to solve the lift scheduling problem should the A* search prove to be unsuitable due to the speed of the search or insufficient memory.

In order to produce a suitable schedule, the following research approach was taken.

- Ensure a good heuristic function is used, which implies spending more time generating nodes but results in the need to expand fewer nodes. This has the advantage that if the search needs to be interrupted at the end of the allowed scheduling period, the scheduler then is more likely to have either the optimal assignment or one close to the optimal.
- Should the above solution be found unsuccessful, perhaps due to speed or memory problems, it may be suitable to consider IDA*.

4.4 Lift monitoring system

It is apparent from the literature and from Section 4.3 that to achieve an optimum call allocation strategy, it is essential to know the current and the future passenger traffic patterns. Historical traffic models could be obtained for different days through the month or the year. Popular floors and future passenger arrival rates can be predicted using a historical traffic model with an arrival pattern similar to that of the current traffic pattern This information can be used to generate false landing calls (a false landing call is issued by the lift control system) or to detect up peak traffic modes. Popular floors can dynamically change with the flow of traffic through the day and hence this information can be used to distribute free lifts evenly around the building A car call model can be used by the monitoring system to predicted car calls which will result from landing calls.

The monitoring system needs to·

- incorporate a dynamic model of the lift movements, so that it can predict times between lift assignments accurately.
- be responsible for providing the lift scheduler with information such as the times taken to open and close doors and the first floor at which a currently moving lift can stop following deceleration.
- use constraints such as 'lifts with maximum load should not be further assigned' to determine accurately the number of lifts available for the next assignment cycle.
- make sure that rules such as 'a lift must answer all car calls in its direction before reversing direction' are forced, by hiding car calls in the opposite direction to the current direction of a lift from the scheduler until all car calls in the lift direction are answered.

4.5 Conclusion

In this chapter the characteristics of the proposed intelligent real-time lift scheduling system have been discussed with Figure 4 1 showing the general structure of the proposed intelligent lift system The monitoring system will use traffic models based on actual examples extracted from the Kodak building to enable the daily traffic patterns to be recognised. The traffic model together with the simulator will serve as an evaluation tool for evaluating alternative scheduling strategies [Hamdi and Mulvaney 1995]

Heuristic search techniques appear to provide the most promising approach for solving the real-time scheduling problem. In the approach, the basic scheduler is kept simple, with the advantages that it is easy to configure and adaptable without the need for additional parameter tuning. The scheduler goal is that used most commonly to compare scheduler performance, namely to minimise average waiting time to achieve fairness among all waiting passengers

Chapter 5 discusses in detail how data extracted from an actual lift installation have been used to develop a model of passenger movements while Chapter 6 describes the implementation of the lift simulator as a part of an intelligent real-time scheduling system. The implementation of the A* scheduler can be found in Chapter 7 and Chapter 8 shows the test results using the lift simulator and the effect of using prediction and high data quality on the performance of the lift system.

Chapter 5

Modelling of passenger movements

In this chapter, data extracted from lift installations have been used to develop a model of passenger movements. Such a traffic model is needed for generating traffic in the lift simulator and to predict passenger arrivals in the lift monitoring system. This chapter explains the methods of analysis applied to the lift system data and detailed descriptions of the development of the passenger arrival model and the passenger destination model are given. The data were obtained from the Kodak lift system (represented in Figure 5.1) which serves a building used for business purposes with a population of approximately 900

Traffic information consisting of data sampled each second and containing landing call locations, car call destinations, photocell activations, lift positions, door status, directions of movement of the lifts, whether each lift is stationary or moving and the load status of each lift. It is important to note that the raw data available do not contain information regarding the movements of individual passengers in the lift system, yet this is precisely the information required for the lift simulator and scheduler. An important aim of the work is to extract a suitable estimate of the passenger arrival rates and their movements through the building If these can be estimated adequately from the data then this would avoid the need to resort to counting passengers manually. Another advantage of being able to gather the traffic information from a real lift installation is that it can easily be acquired over a longer period of time and from a greater number of landings and lifts simultaneously than is normally possible using manual counting. This has allowed a detailed assessment of assumptions often made regarding the nature of the probability distribution of the passenger arrival rates, namely that it follows a Poisson distribution and hence the time interval between passenger arrivals follows an exponential distribution [Hamdi and Mulvaney 1998]

5.1 Display module

As a first step toward understanding a lift system and to be able to test its performance, a lift traffic display screen was implemented as a separate module that shows the movement of lifts and passenger calls, Figure 5.1. The display module uses data that are read and updated each second from serial files recorded from the Kodak building and other Express Evans Ltd. lift systems.

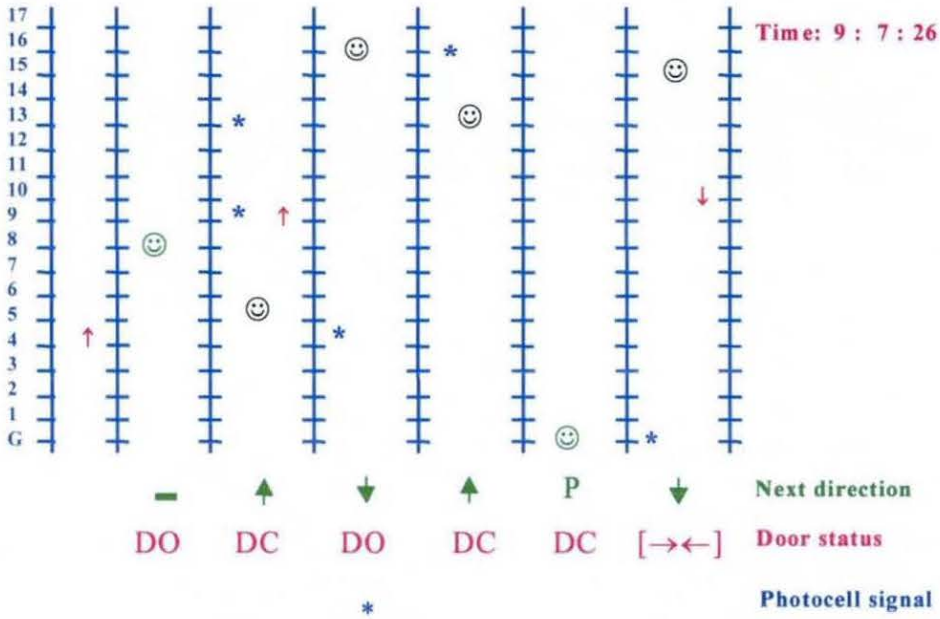


Figure 5.1 Lift system display viewing the Kodak building.

The display module divides the screen into columns and rows representing floors and lifts in a building depending on the building information supplied to it. The first column represents the floor numbers. The second column represents landing calls down (↓) and landing calls up (↑), the following columns represent the lift paths with indicators for car calls (*), allocated landing calls (↑ or ↓), and the lift position ☺. Time is shown at the top right of the screen. The lift cars are represented by symbols, ☺, that flash together with the direction status signal symbol indicating lift movement. The direction and other signals are displayed below each lift column. The lift load is shown by changing the colour of the lift signal (normally white, green for 60% load, cyan for 90%, and magenta that in Figure 5.1 the unloaded lifts are green and the background colour has been inverted to white from black for the clarity of illustration.

5.2 Analysis of Kodak data files

The display module allows the author to monitor the behaviour of the lift system and the following subsections highlight examples which were obtained from the serial files for 4/7/1994 at the Kodak building:

Example 1

Figure 5.2 shows the position of a lift travelling without stopping from floor 14 to the 2nd floor between 8.00 00 am and 8:00·25 am. While the lift is moving down between the 7th and the 8th second, its position changes from floor 11 to floor 9 and remains at floor 9 for the following second The data then show the lift has moved to floor 7 with no data sample at floor 8 and remains at floor 7 for the following second

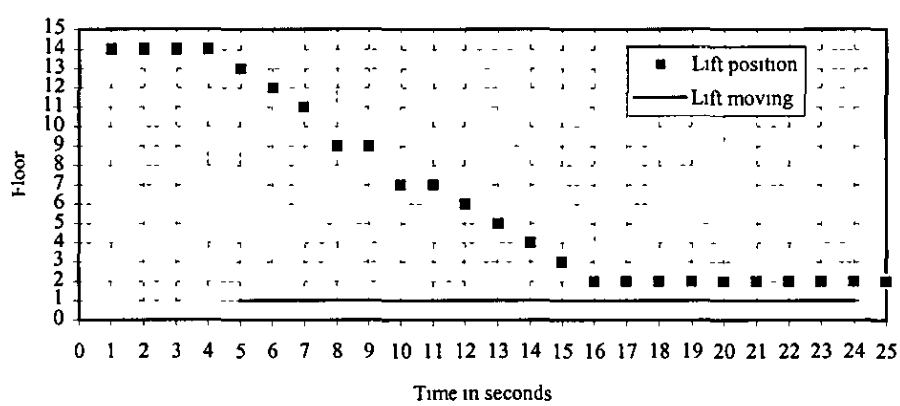


Figure 5.2 Example illustrating the effect of the sample rate on the data obtained from the Kodak building.

The data indicate that the lift begins moving at the 5th second and stops at the 24th second, whereas the lift position is the destination floor (2nd floor) at the 16th second. The lift next direction is down for the entire period.

The example reflects the accuracy the serial files are able to provide at the sample rate (1 second) employed The lift system gives an advanced lift position to the scheduler once a lift starts moving. For example, if a lift were travelling from the 3rd to the 2nd floor, then the 2rd floor would be given as the lift position by the lift control system, explaining why the lift position remains unchanged while the lift is still moving In this example the lift takes 8s when decelerating, but the typical deceleration time is 4s in the Kodak building, Appendix B.

Example 2

Figure 5.3 shows a lift movement between 8:01:10 am and 8:01:45 am, where the lift has left floor 11 and moved down to the ground floor at the 81st. Although the lift does not stop moving until the 88th second, the next direction changes to indicate UP as soon as the destination floor is reached The lift remains at the ground floor until the 105th second when it starts moving in the up direction

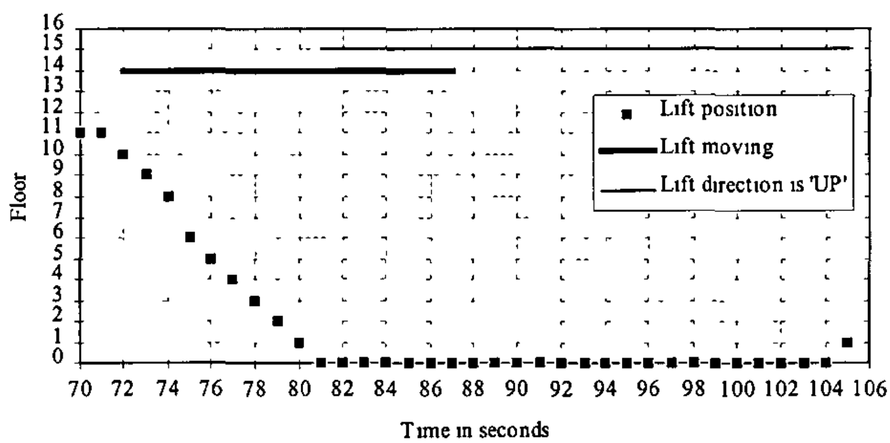


Figure 5.3 Example to illustrate the next direction indicator.

Example 3

Figure 5.4 shows the movement of a lift between 8 00:32 am and 8:00:53 am. When the lift leaves floor 11, it is assigned to answer a car call at floor 6 and to answer a landing call up at the terminal floor. At the 40th second, the assignment changes and the lift's next assignment is to answer a landing call up at floor 6 and its next direction changes to UP. The car call indicator is cancelled while the lift is still moving at the 41st second and the lift stops moving at the 42nd second. At floor 6, the doors open, the passengers leaving and entering the lift activate the photocell indicators at the 44th second for a two second interval, and then again at 47th second for a further second. At the 53rd second the lift starts moving up.

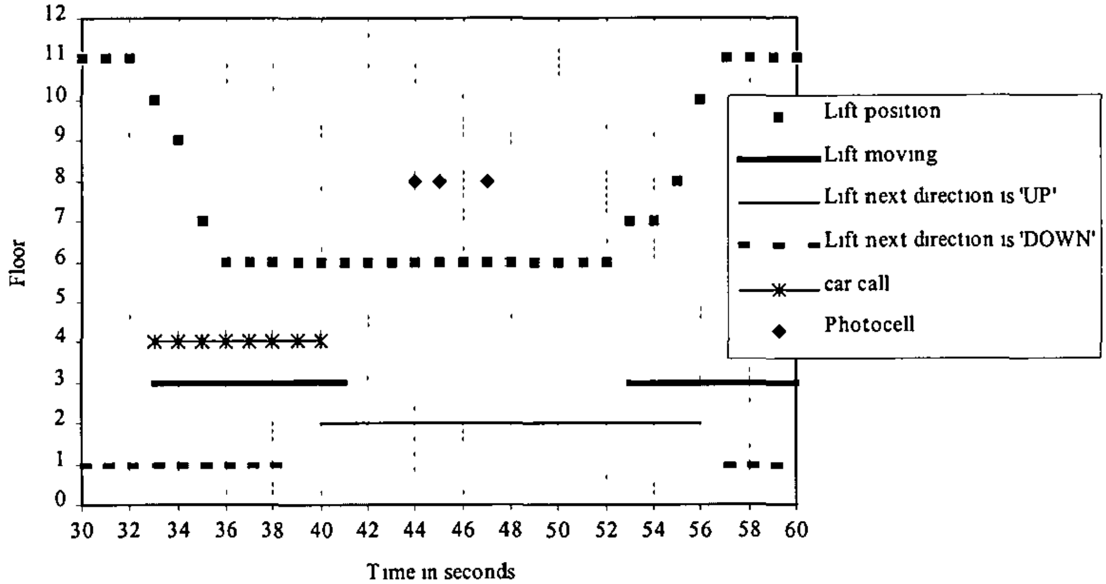


Figure 5.4 Example to illustrate assignments changes during lift movement.

This example shows that the lift next direction indicator changes before reaching the destination floor to allow the lift to indicate to waiting passengers its next assignment direction The lift assignment may change while the lift is still moving and not decelerating, and once the lift is decelerating, the assigned landing call at the destination floor is removed and no longer considered by the scheduler.

From the analysis of the data, the following points are relevant to the work carried out in the thesis·

- As the data were originally sampled every second, it is not clear when the lift position is duplicated in successive sample whether the lift is slowing down or whether the effect is created as a result of the sampling rate. Consequently, the time when a lift starts decelerating is not directly available.
- Lifts arrive at floors either in response to a landing call, a car call, or following a specific action of the scheduler, such as parking at a popular floor.

- The next direction signal of a lift may be changed while the lift is still moving.
- When the lift moving signal changes to ON, the lift position signal indicates the floor in advance of the actual position.
- Landing calls and car calls are cancelled when the lift begins decelerating to stop at their floor. The landing call indicator is usually switched off before the assigned landing call indicator.
- Photocell activations and the number of car calls registered can help to indicate the number of passengers entering or leaving the lift.

The display module is the first step towards understanding the lift system The available data are useful for obtaining the passenger movement model which will be explained in the following sections The data accuracy affects the comparison of final simulation results to the actual system as will be explained in Chapter 6 and Chapter 7.

5.3 Lift system traffic patterns

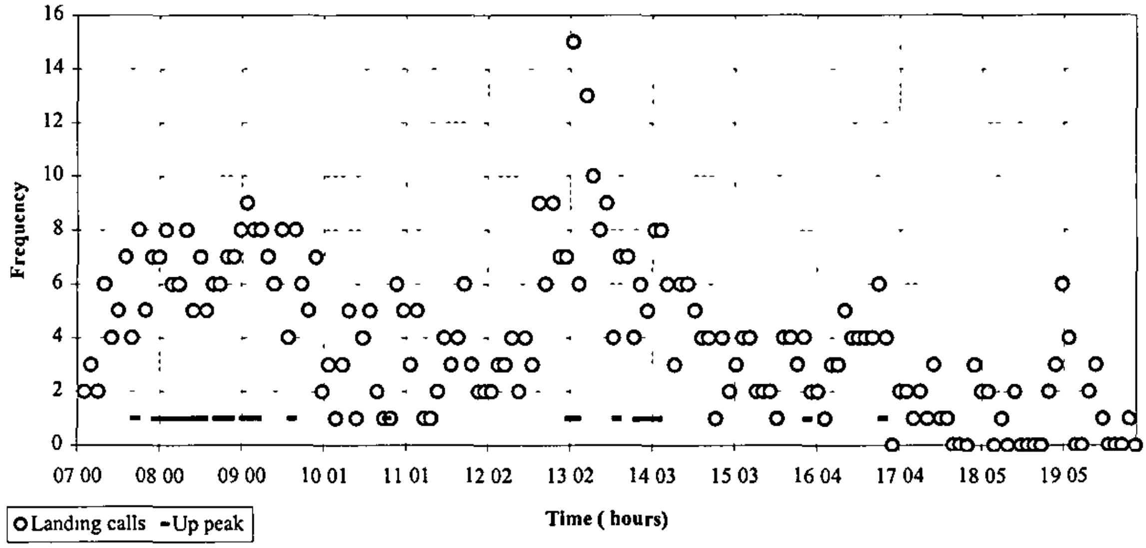
There are two alternative approaches that can be used to obtain data suitable for the use in the model of the lift system. The first is to model landing and car calls and permit the scheduler to respond to them and the second is to model the passengers themselves In the first approach, the model could be produced directly from the data obtained from the Kodak building, but it would be difficult to provide car calls, which were consistent with the landing calls In the second approach, the extraction of the model is more complex, while implementation would be more straightforward. The author chose to attempt to extract a passenger model as this would appear to give the more elegant solution in the main areas of the current work The patterns in the Kodak data depend on the number of people in the building, the number of passengers using the lifts, popular floors (such as those having a restaurant or smoking room), and the time of the day (such as up peak and lunch time). Traffic behaviour can be monitored to predict information useful to the lift scheduler, such as passenger arrival time and car call A monitoring system has the responsibility of monitoring these patterns and identifying popular floors and expected peak times. For example, at the terminal floor, the morning arrival in a building can start at 7am, its peak state is reached at around 8am but then begins to lessen at about 10am The traffic peaks again at lunch time when traffic is more evenly distributed between floors, including the restaurant and terminal floors The same traffic pattern can repeat the next day, but may be shifted forward or back in time, or another pattern may emerge with different passenger arrival rates It often appears that, for the same number of passengers entering the building, the intensity of traffic at any given time during the up-peak period depends on when the main arrival begins and with what initial intensity [Hamdi and Mulvaney 1998]

To study the traffic behaviour in the Kodak building, landing calls were counted in five minute intervals and the number of car calls were recorded for each hour for all floors for several days Two different traffic patterns are given as examples, the first in Figures 5 5, 5 6, 5 7 and 5 8 for 3/10/94 and 4/10/94 and the second in Figures 5.9, 5 10, 5.11 and 5 12 for 24/10/94 and 25/10/94 Data such as those shown in these figures were investigated to determine if there is similarity between pairs of days in the week or between the same days of consecutive weeks For example, the traffic between floors for 24/10/94 and 25/10/94 is less than that on the same two days of the week, beginning 3/10/94, table 5 1 The arrival rate on the 24th and 25th, between 7am and 8am for the ground floor is relatively low, while at 9am there is marked increase in traffic especially in the number of car calls, Figure 5 13

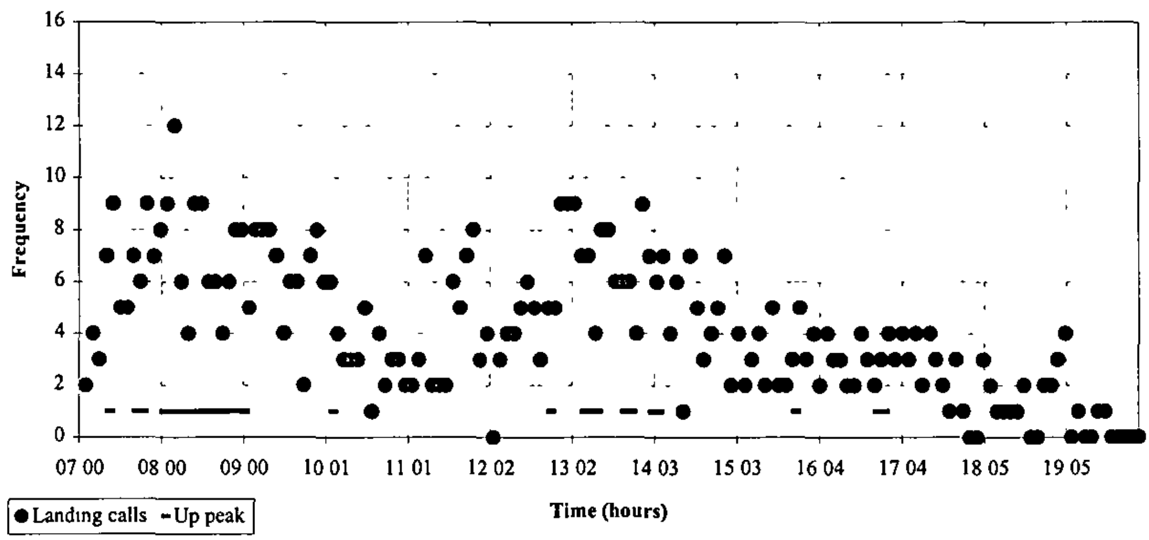
	03/10/94	04/10/94	24/10/94	25/10/94
Total landing calls	620	646	554	568
Total car calls	590	578	463	478

Table 5.1 Total landing calls and car calls at the terminal floor between 7am and 8pm for the 3rd, 4th, 24th and 25th of October 1994 in the Kodak building.

Figure 5 6 shows that the traffic for the second floor in (the restaurant floor) peaks around 1pm on the 3/10/94 and 4/10/94, while in Figure 5.10, the restaurant floor traffic peaks at around 3pm on 24/10/94 and 25/10/94 Similarly for the terminal floor, Figure 5 14 shows that more passengers arrive in the building after 2pm on 24/10/94 and 25/10/94 in comparison with the numbers arriving on 3/10/94 and 4/10/94

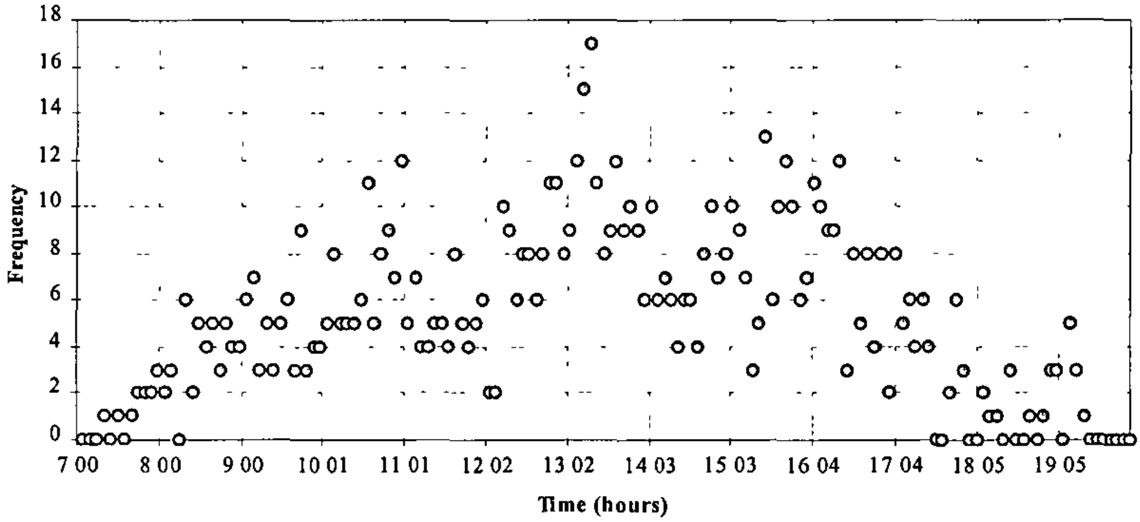


(a) Landing calls on 3/10/94

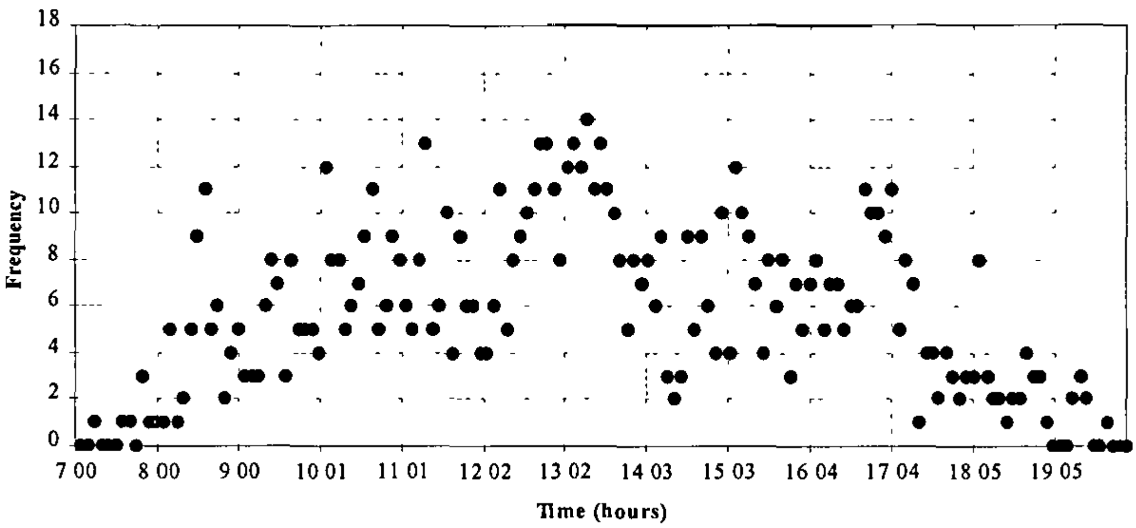


(b) Landing calls on 4/10/94

Figure 5.5 Landing calls calculated every 5 minutes at the terminal floor between 7am and 8pm in the Kodak building.



(a) Landing calls on 3/10/94



(b) Landing calls on 4/10/94

Figure 5.6 Landing calls calculated every 5 minutes at the second floor between 7am and 8pm in the Kodak building.

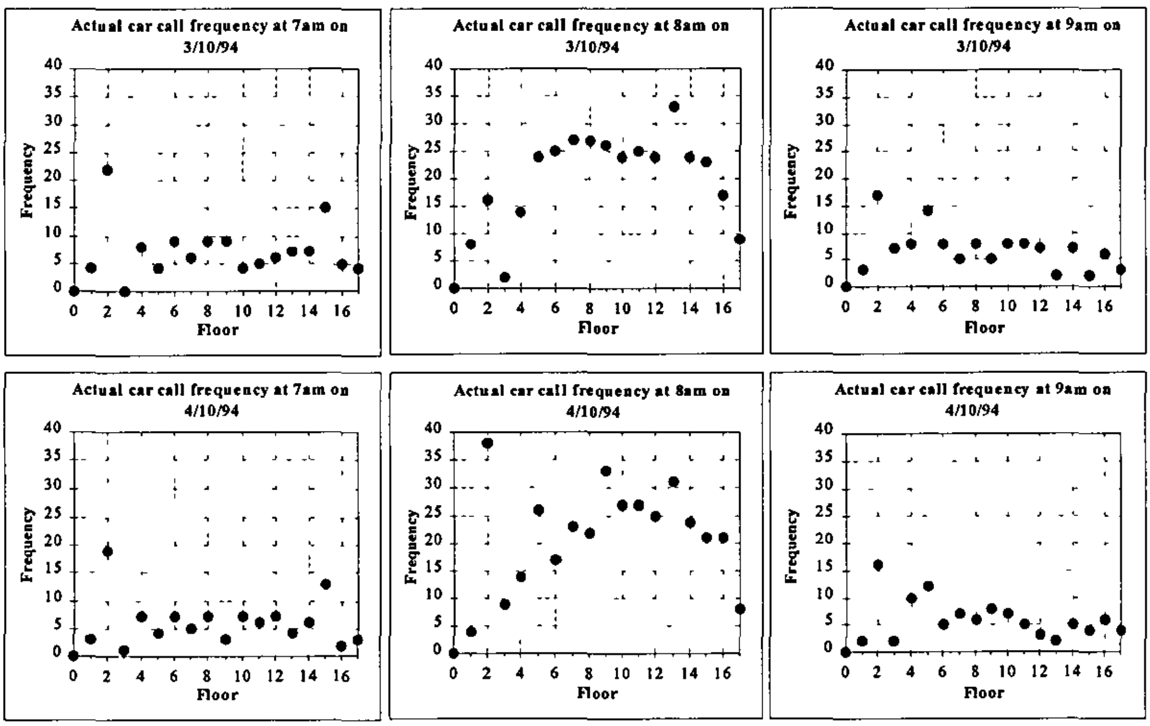


Figure 5.7 Actual car call frequency distributions from the terminal floor to all floors between 7am and 10am in the Kodak building.

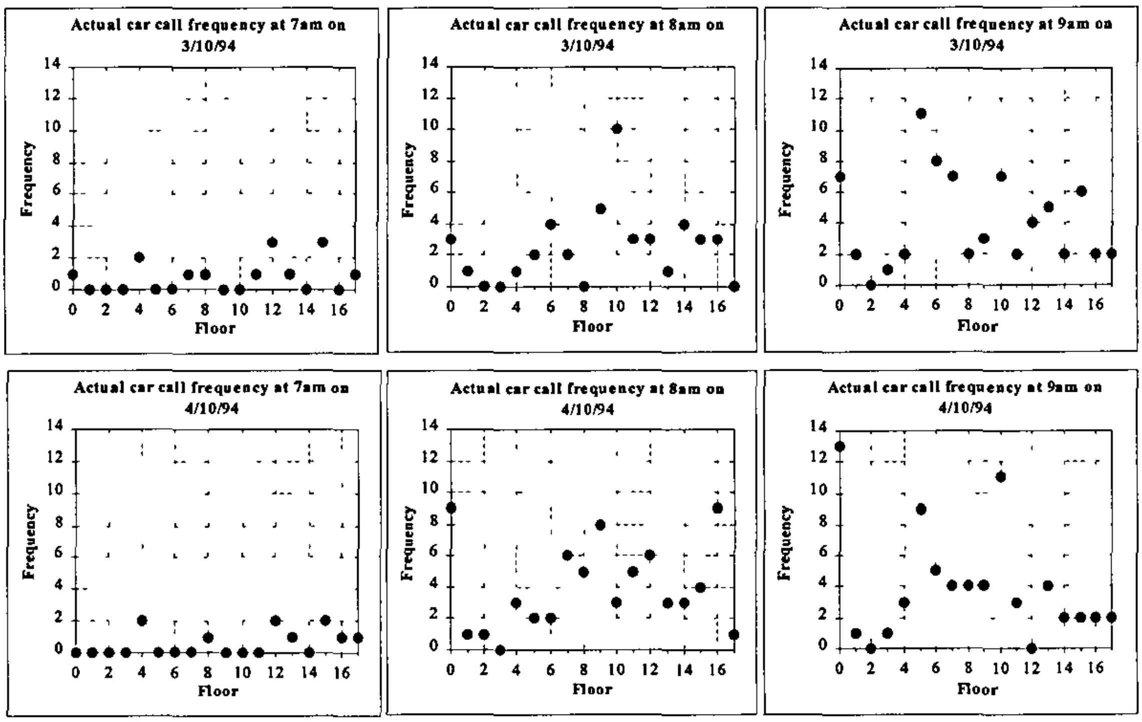
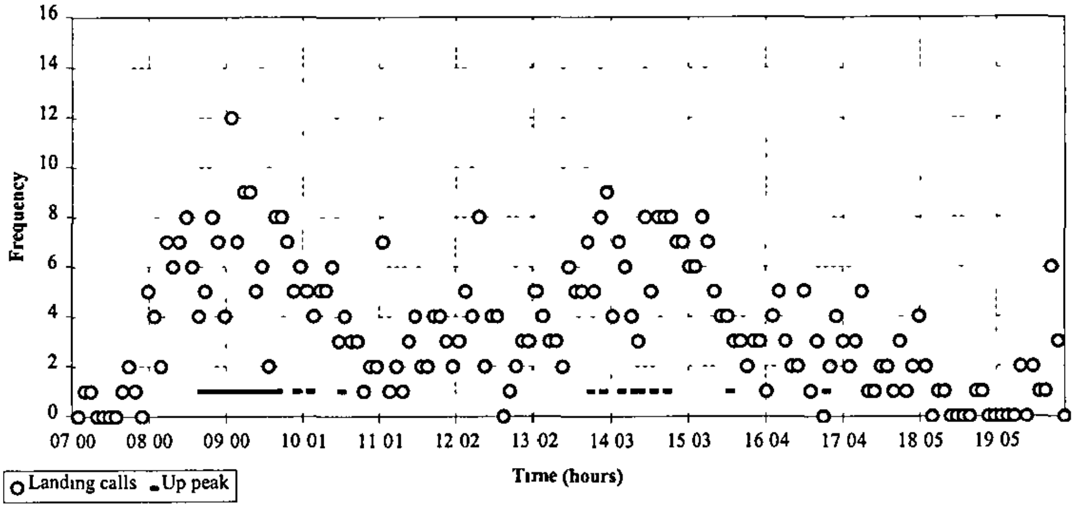
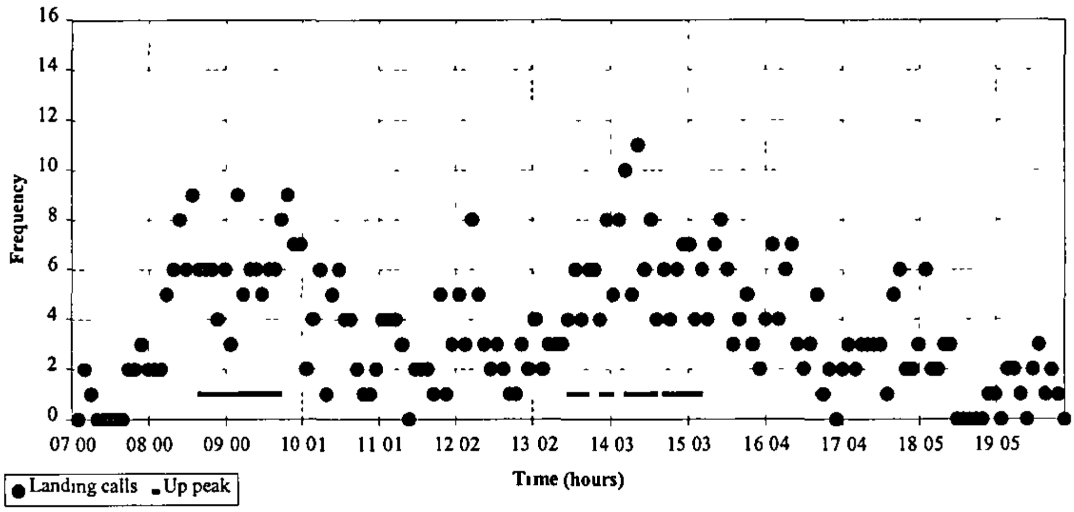


Figure 5.8 Actual car call frequency distributions from the second floor to all floors between 7am and 10am in the Kodak building.

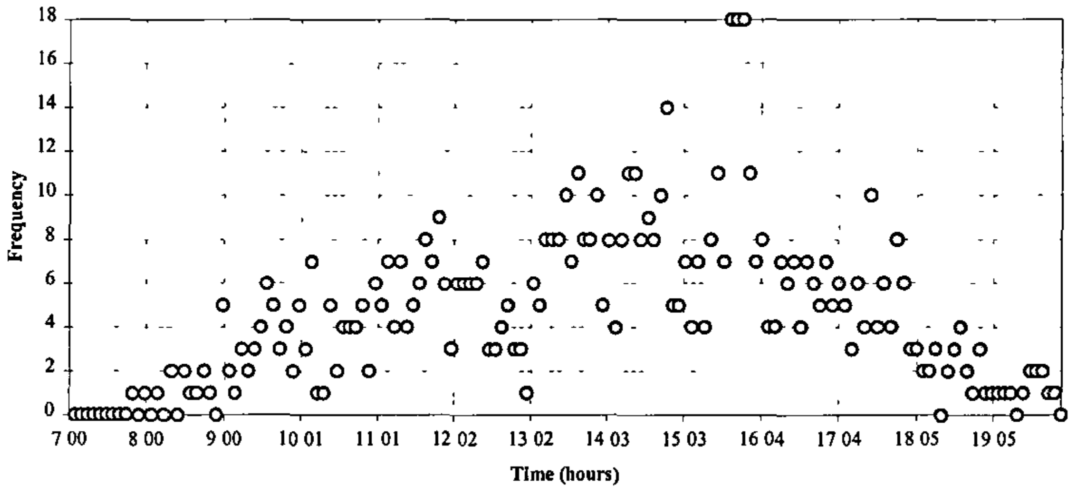


(a) Landing calls on 24/10/94

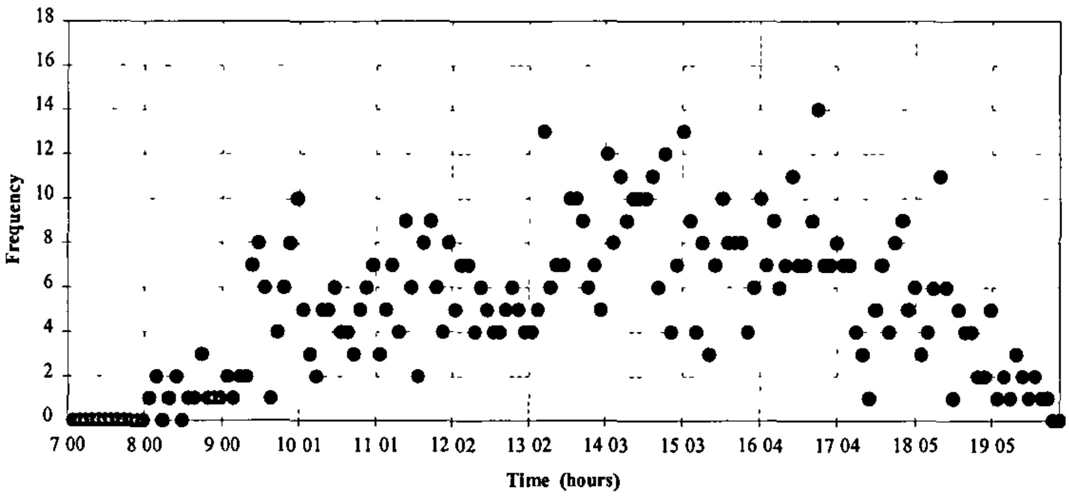


(b) Landing calls on 25/10/94

Figure 5.9 Landing calls calculated every 5 minutes at the terminal floor between 7am and 8pm in the Kodak building.



(a) Landing calls on 24/10/94



(b) Landing calls on 25/10/94

Figure 5.10 Landing calls calculated every 5 minutes at the second floor between 7am and 8pm in the Kodak building.

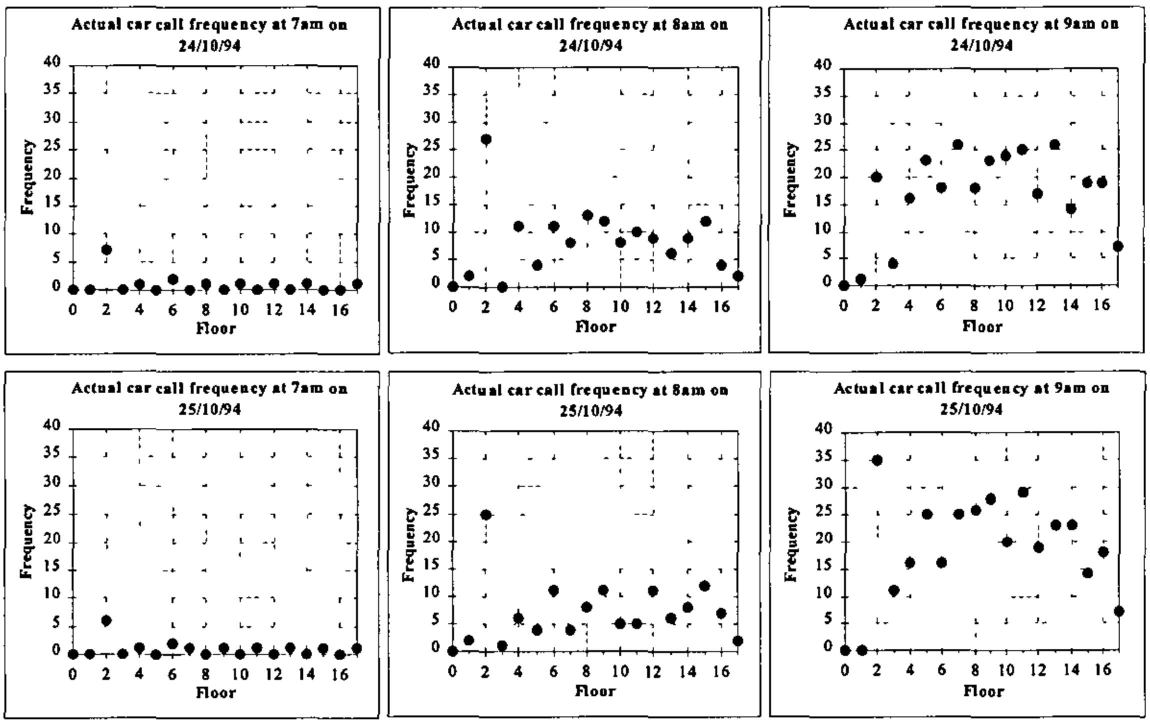


Figure 5.11 Actual car call frequency distributions from the terminal floor to all floors between 7am and 10am in the Kodak building.

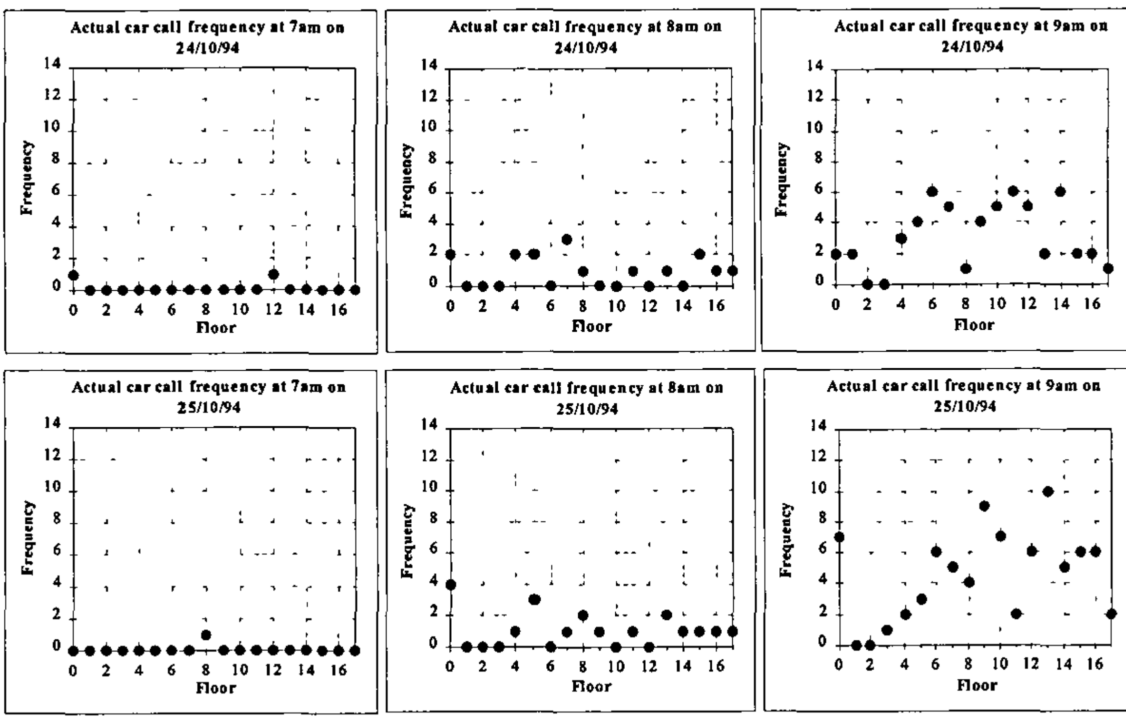


Figure 5.12 Actual car call frequency distributions from the second floor to all floors between 7am and 10am in the Kodak building.

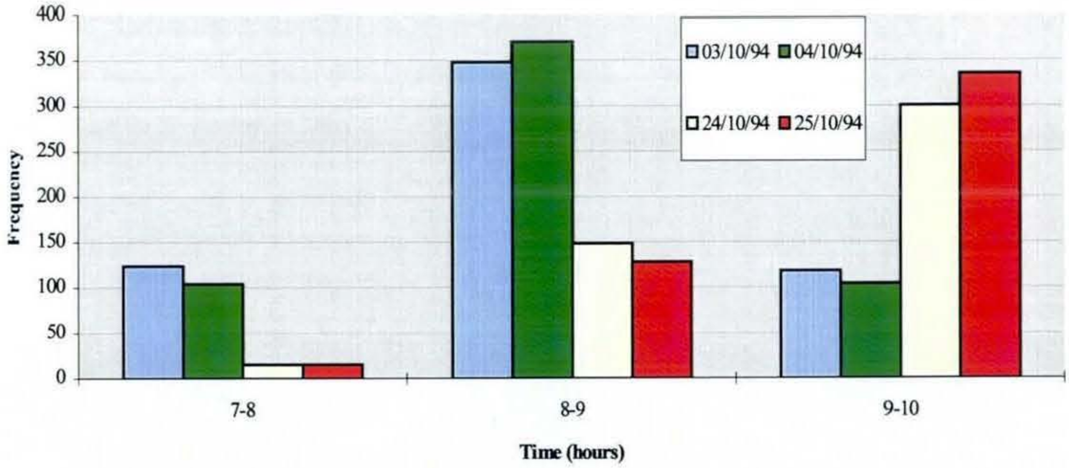


Figure 5.13 Car call frequency calculated every hour between 7am and 10am in the Kodak buildings for the 3rd, 4th, 24th and 25th of October 1994.

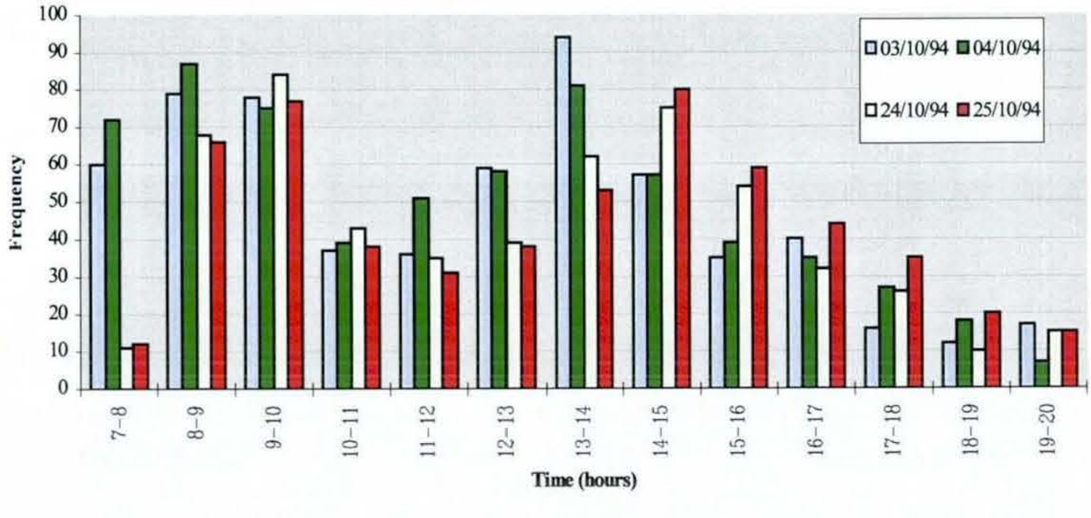


Figure 5.14 Landing call frequency calculated every hour between 7am and 8pm in the Kodak buildings for the 3rd, 4th, 24th and 25th of October 1994.

The Kodak building does not appear to have a single uniform traffic pattern. Consequently, a lift monitoring system would have to calculate the current arrival rates and compare these with historical samples of daily traffic in order to select a suitable traffic sample for prediction purposes.

5.4 Derivation of a passenger arrival and car call model

The passenger and car call models are extracted from a real lift system, overcoming the need to identify theoretically the traffic patterns Although theoretical traffic calculations would still be needed in the planning stage, following installation the initial traffic assumptions would be gradually replaced by real traffic examples when the intelligent monitoring system starts working. Average arrival rates and passenger destination frequencies from each floor can be calculated from information such as car calls, landing calls, photocell activations, door status and lift moving status. Examples of full working days of traffic patterns have been prepared to test the performance of the scheduler.

The investigation of the data acquired from the lift installations has been extensive, and the following figures are able to provide only a brief but representative insight into the results which have been produced. In order to make clear the meanings of some of the terms used in the figures which follow, Figure 5.15 shows the typical sequence of events which occurs when a lift is sent in response to a landing call.

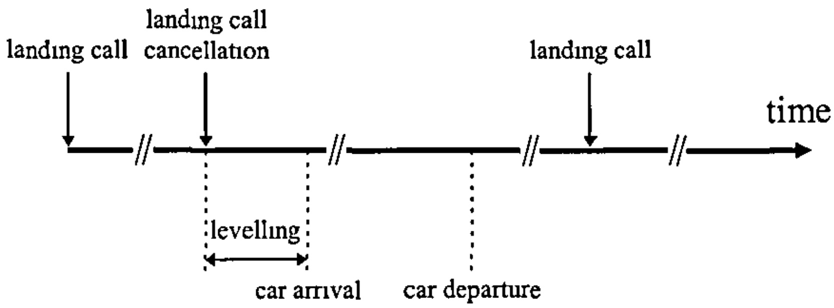


Figure 5.15 The sequence of events following the issue of a landing call.

In order to develop the passenger model, the validity of the assumption found in the literature that the time interval between passenger arrivals follows an exponential distribution was assessed. Figure 5.16 shows the distribution of intervals between consecutive landing calls. It can readily be observed that this distribution is not exponential in nature and this can easily be explained by considering the sequence of events following the issue of the landing call. Most importantly, if a lift is not already waiting at a landing, it will take time for a lift to arrive in response to the call and for the doors to open. Consequently, as can be seen from Figure 5.16, the time period between the issues of consecutive landing calls will take at least 12 seconds. Furthermore, during this period the landing call remains in force and, hence, additional passengers may arrive who will not be directly detected by the lift system.

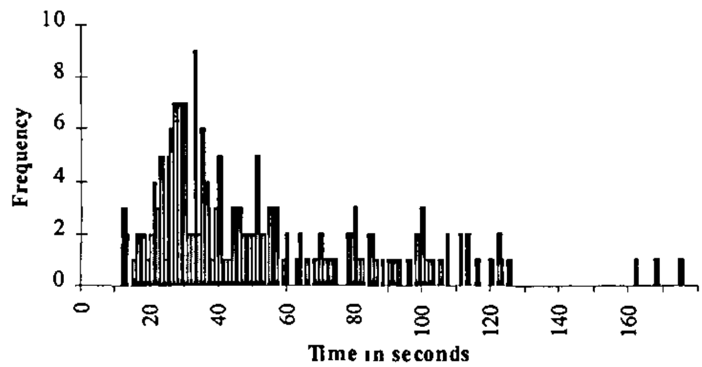


Figure 5.16 Distribution of the intervals between landing calls on the terminal floor between 7am and 10am on 3/10/94.

The significant difference that can occur in the distribution of intervals between passenger arrivals and that of intervals between landing calls can be illustrated by considering two types of passenger traffic In the case where there is a relatively large number of passengers arriving over a period of time at a single floor, the performance of the scheduler may worsen and it will take longer for landing calls to be answered and hence for the next landing call to be issued. In the case where there are few passengers entering the system, there will be correspondingly long time intervals between landing call buttons being pressed. Comparing the two cases, although the time intervals between passenger arrival is significantly different, the time intervals between landing calls may be similar.

In order to better understand the nature of the traffic, a variety of intervals between the events highlighted in Figure 5.15 was considered By plotting for a given landing the time interval between the issue of landing call and its cancellation, the performance of the scheduler can be studied, for example in Figure 5.17. An example of the distribution of the time intervals between landing calls and lift arrivals is drawn in Figure 5.18 In comparison with Figure 5.17, the main part of the curve is shifted to the right reflecting the lift levelling time of around four seconds, but the frequency values at one and two seconds largely remain, as these occur when a lift is already at the landing and no levelling period is required. Including the loading time and both the door opening and closing times provides the distribution of time intervals between landing calls and lift departures, Figure 5 19. During this period, any further landing calls cannot be observed.

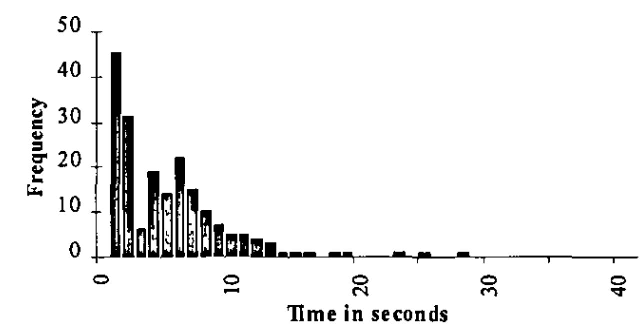


Figure 5.17 Distribution of the intervals between landing calls and their cancellations on the terminal floor between 7am and 10am on 3/10/94.

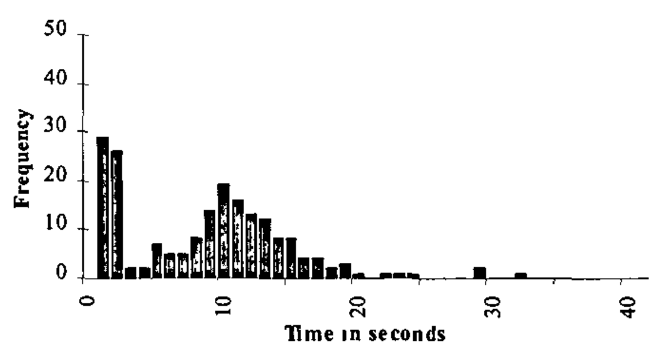


Figure 5.18 Distribution of the intervals between landing calls and lift arrivals on the terminal floor between 7am and 10am on 3/10/94.

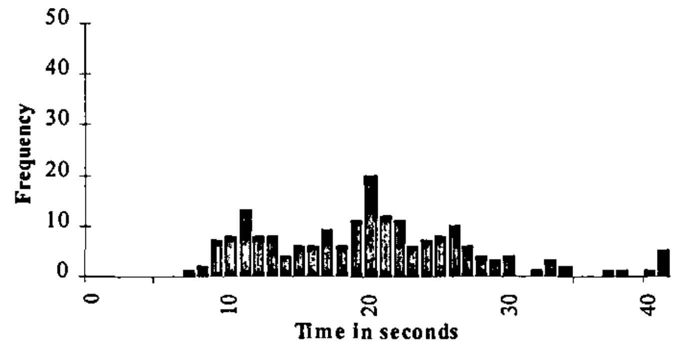


Figure 5.19 Distribution of the intervals between landing calls and lift departures on the terminal floor between7am and 10am on 2/10/94.

In order to estimate the passenger arrival rate at a particular landing, the current work investigated employing the time interval between lift departure and the issue of the next landing call This is effectively the time interval remaining once the system delays illustrated by Figures 5.17, 5.18 and 5.19 have been removed from the time intervals between consecutive landing calls. Only during this interval is the landing call button available for passengers to press, and hence it is the only time during which the arrival of a passenger can be observed (and their arrival which is indicated by their pressing of the landing call button also ends each interval). Clearly, this can only be an *estimate* of the passenger arrival rate, and it is recognised that errors will occur as the arrival of further passengers cannot be observed until the lift departs as more than one passenger may arrive at a time. An example of the results obtained from this approach are shown in Figure 5 20. Figure 5.21 shows the results of fitting to the raw data an exponential curve *f(t)* of the form shown below.

$$f(t) = A\lambda e^{-\lambda t}$$

where λ is the average time interval between the lift departure and the next landing call being issued, $A = \lambda T$ and T is the time during which observations are made.

In general, the hypothesis of an exponential form for the distribution of the intervals between lift departure and the next landing call agreed with the data However, on investigation of the area under the curve in Figure 5.21 (which should be equal to the number of passengers arriving during the observation time), it was found that 206 passengers arrived, similar to the number of landing calls issued (217). Measuring the time interval between the lift departure and the next landing call did not provide sufficiently good estimates of the number of passengers. This is probably due to the fact that the data are non-stationary, that is, there is significant change in the arrival rate over the length of the observation period and hence the shape of the underlying exponential function is continually changing. Rather than requiring the fit of single exponential curve as shown in Figure 5.21, the real data comprises, over time, a family of exponential curves. By adopting sufficiently short time windows, the variation in the arrival rate could be identified, but as it would contain fewer examples of arrivals, the variance of the data would be significantly increased resulting in estimates of reduced quality.

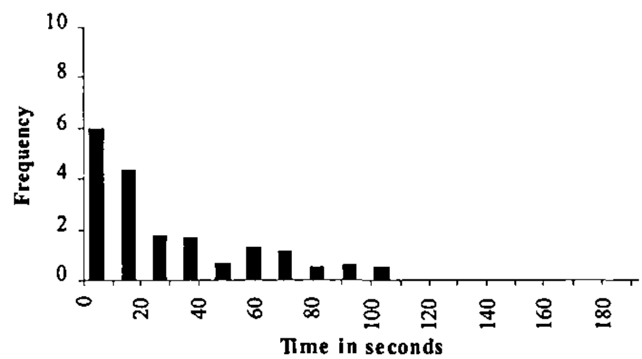


Figure 5.20 Distribution of the intervals between lift departures and next landing calls on the terminal floor between 7am and 10am on 3/10/94.

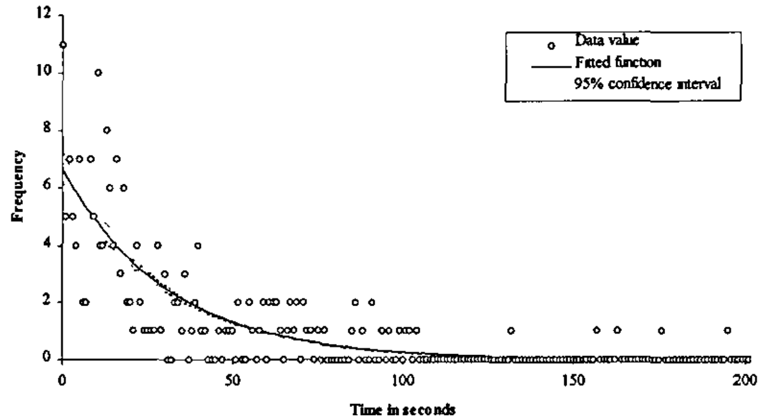


Figure 5.21 Test of the exponential nature of the distribution of the intervals between lift departures and next landing calls on the terminal floor between 7am and 10am on 3/10/94.

An alternative method was now required to model the passenger arrival. In its development, the following points were considered [Hamdi and Mulvaney 1998].

- To observe changes in arrival rates, the mean arrival rates during consecutive five minute windows were used The window length was so chosen as to be consistent with that used in the calculation of lift handling capacity during the up-peak period [Barney, 1988]
- Lifts arrive at a floor in response to a landing call at that floor, a car call to that floor being issued by passengers already in the lift, or following a specific action of the scheduler.
- The number of car calls issued and the number of photocell activations both provide an indication of the number of passengers entering the lift.

A set of rules was developed to estimate the number of passengers waiting at landings to enter the lift, based on the number of car calls issued and the number of photocell activations. The rules are shown in Figure 5.22 In general, when passenger departures are detected, or when an inconsistently large number of photocell activations is detected, the method bases its estimations on the number of car calls issued following boarding. When the photocell count is valid and is greater than the number of new car calls, it is used directly to represent the number of passengers.

As a result of the investigation of car calls and photocell activations, a model of passenger arrival could be generated for use in the simulator. Figure 5.23 shows the passenger arrival rate extracted from a real lift installation using the rules of Figure 5 22 and that obtained from the model by adopting the same arrival rate for a Poisson distribution in the lift simulator. To assess the validity of the rules, Figure 5.24 shows a comparison between the landing call rates for both the real lift installation and for those obtained in the lift simulator using the model. It can be seen that there is close correspondence between the two curves, except during the morning up-peak period (between about 8am and 10am). This is probably due to a known difference in the performance between the real and the simulated scheduling systems during the up-peak period and will be discussed in detail in Chapter 6. Moreover, it was also observed that different simulation runs do not necessarily produce exact similarity for the same number of passenger as their arrival time and their destination would have an effect on the results

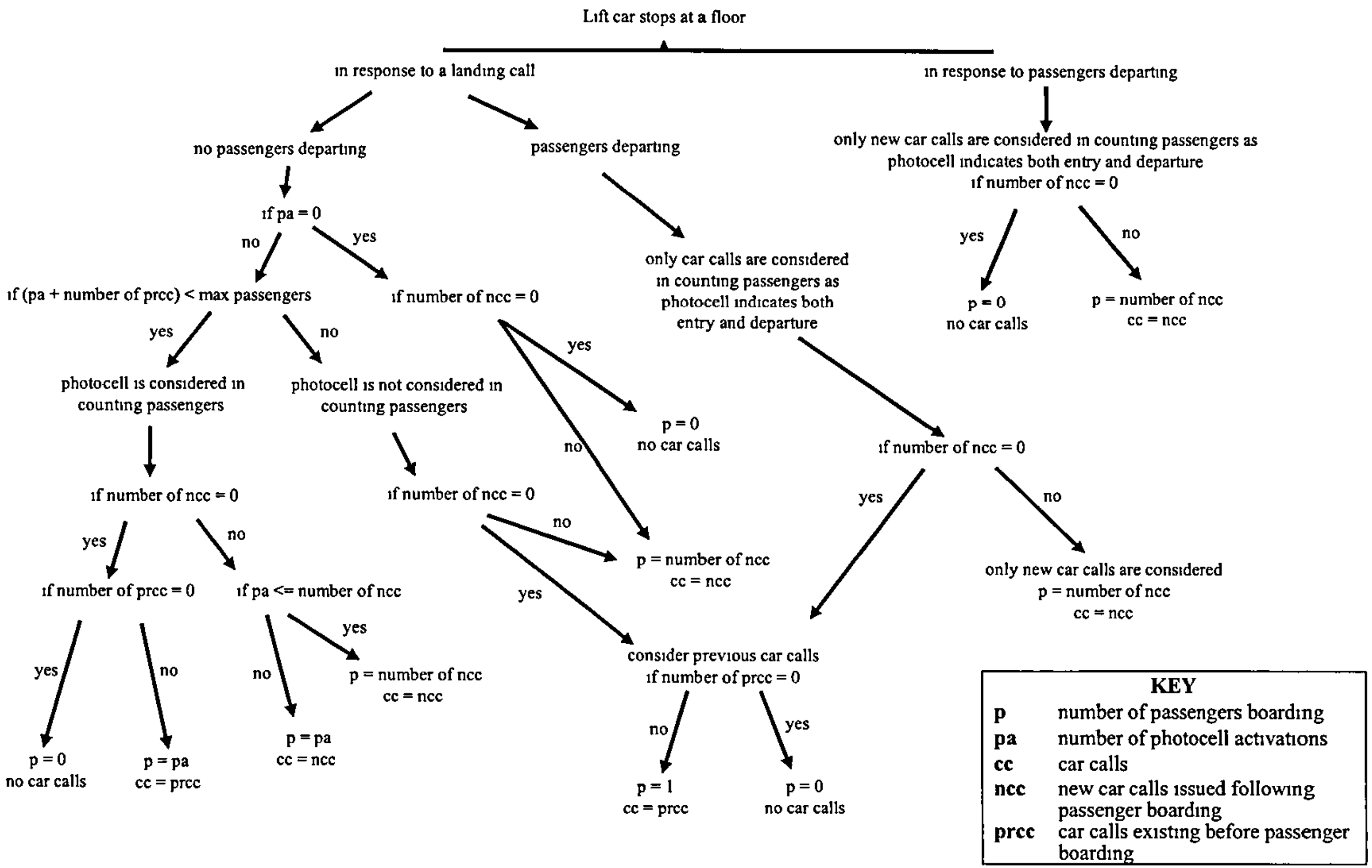
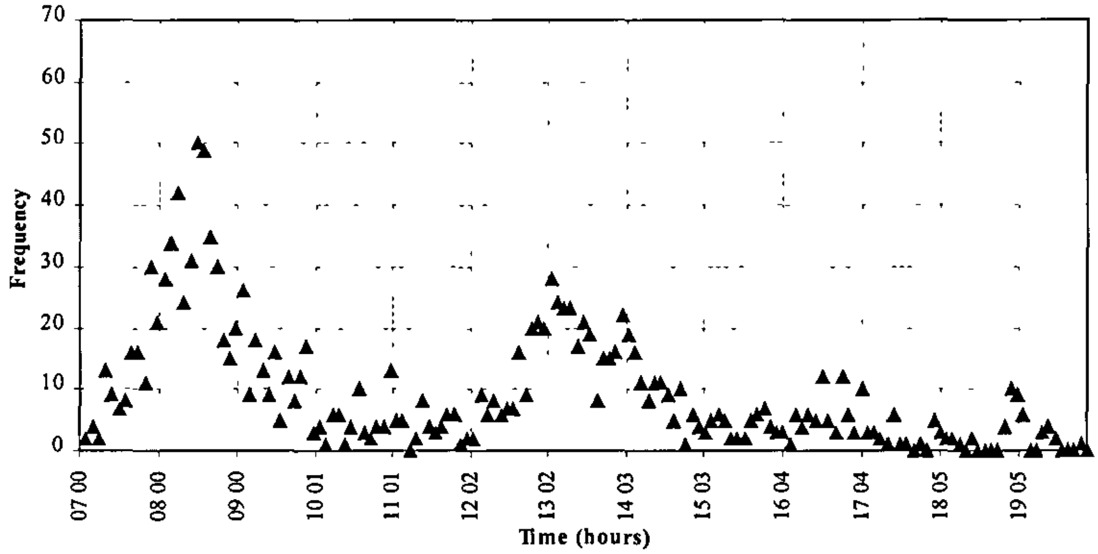
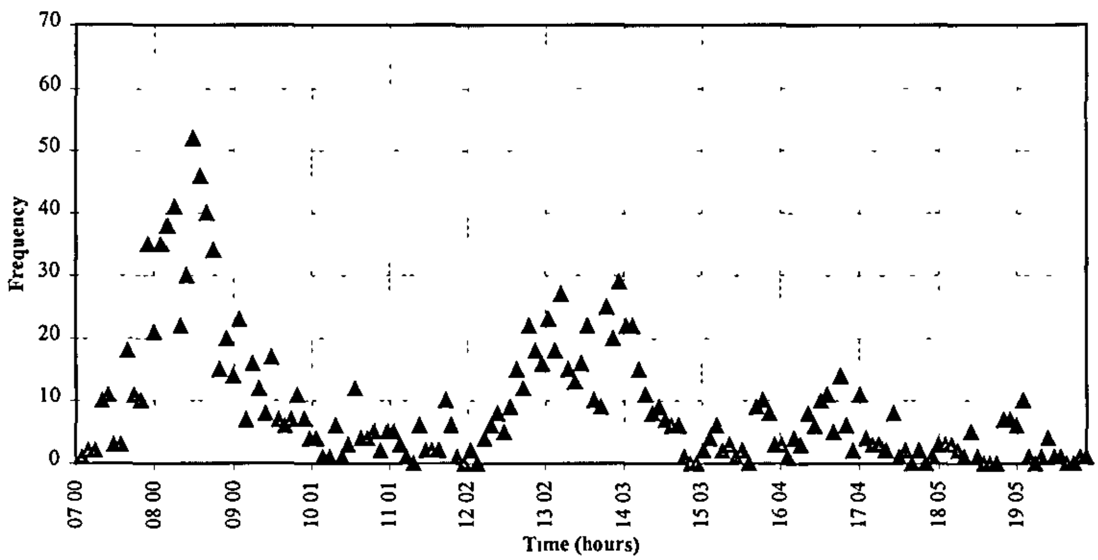


Figure 5.22 The rules used to extract an estimate of passenger numbers from car calls and the number of photocell activations

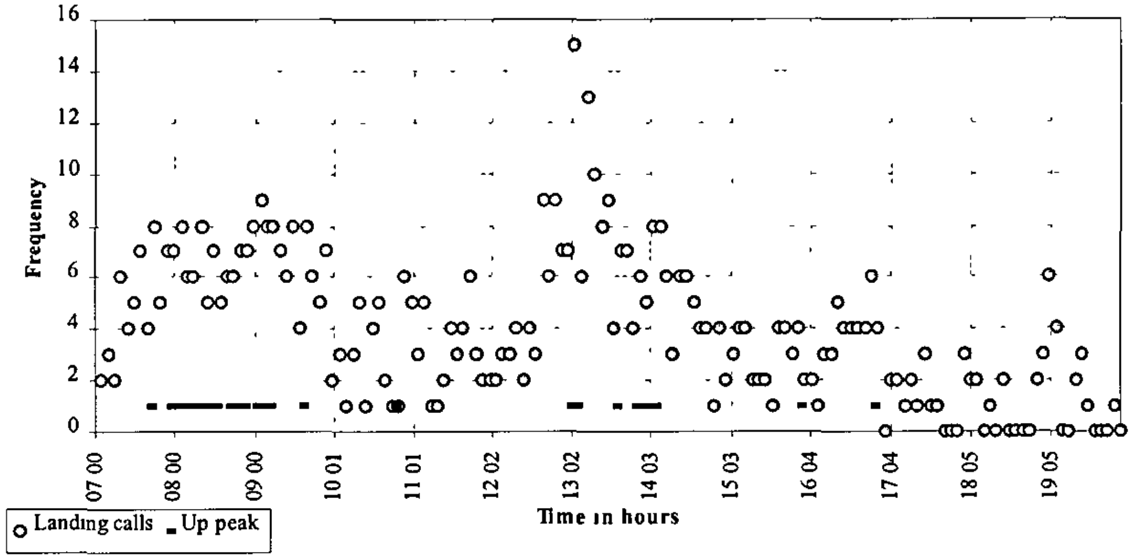


(a) Estimated Passenger arrival in a real lift installation extracted using the rules in Figure 5.22

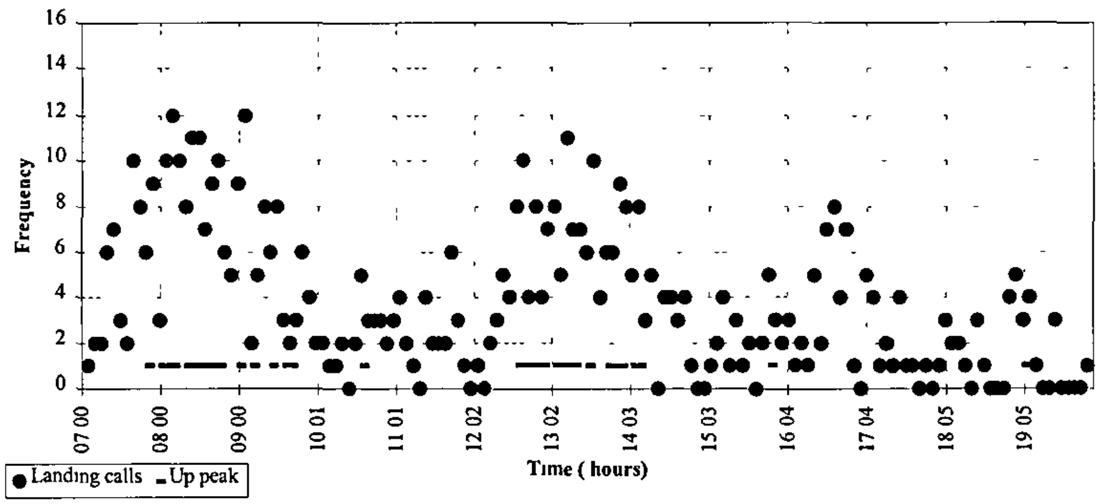


(b) Simulation of the passenger arrival in (a)

Figure 5.23 Comparison of extracted passenger arrival using the rules of Figure 5.22 with simulated passenger arrival at the terminal floor between 7am and 8pm on 3/10/94.



(a) Landing call variation in a real lift installation



(b) Simulation of the landing call variation in (a)

Figure 5.24 Actual and simulated landing calls at the terminal floor between 7am and 8pm.

The passenger model described above permits the simulation of passengers entering the lift system, but not their movements within it. An additional model is required in order to simulate the pressing of car call buttons by passengers to communicate their desired destination once they enter a lift. Figure 5.22 includes the rules applied to observations of real lift installations to obtain the car call distributions for use in the lift simulator. Applying the rules to each landing, the number of car calls to all other landings served is recorded during each five minute window. This car call frequency model can be used directly in the simulator, or alternatively a mathematically smoothed version may be used in which the car call distribution for a given interval is found from the mean of current and previous windows [Hamdi and Mulvaney 1998].

5.5 Conclusion

In this chapter, traffic data analysis was given, the hypothesis of an exponential passenger arrival was found to agree with the its results and a passenger and car call models were described. The availability of the traffic model can replace the need to identify separately specific traffic patterns such as up peak and down peak, with the monitoring system being given the responsibility of generating relationships between floors and their traffic density. The results shown for the arrival model apply to the terminal floor where the quantity of data available allows statistical assessment to be made Data are also available for other floors, but, due to their relatively sparse nature, they must be analysed typically over a period of weeks in order to obtain statistical significance. The up peak period of three hours between 7am and 10am was chosen in order to have adequate data to represent the passenger arrival pattern The period of observation can be shorter, but it would then need to be combined with similar observation periods of other days. Experiments have been conducted over a period of two weeks for two separate installations and these results all confirm that the arrival model is valid.

The knowledge gained from analysing the lift traffic pattern is necessary for developing a suitable lift simulator for the current work and for providing the data prediction required by the scheduler In the next chapter, the passenger and car call models developed in this chapter are used to simulate passenger traffic and the simulation results are compared with actual data obtained from the lift system.

Chapter 6

Lift simulator

This chapter describes the implementation of the lift simulator developed to test and assess the performance of the intelligent real-time lift scheduling system, Figure 6.1. In the following section a general introduction to the lift simulator main functions is given. The remaining sections discuss the lift simulator functions in details and how passenger and car call models described in Chapter 5 have been used to develop the traffic movements

- Section 6.2 describes in details the main functions of the simulator. This includes the calculation of lift trip time in Section 6.2.1 Updating landing call assignments is described in Section 6.2 2. The simulation of passenger arrival and landing call generation is explained in Section 6.2.3 Car call generation and the simulation of lift car, such as door opening and passenger entry is given in Section 6.2.4.
- Section 6.3 discusses the effect of the time step chosen on simulation performance.
- In order to test the simulator a simulated version of the ETQ was used to compare simulation results with the actual Kodak traffic information The simulated ETQ is described in Section 6.4, as well as the parking and up peak policies that is used in Kodak lift system, Sections 6 4 1 and 6.4.2.
- An analysis of the simulation results is given in Section 6 5.
- A comparison of the actual and the simulated ETQ is given in Section 6.6

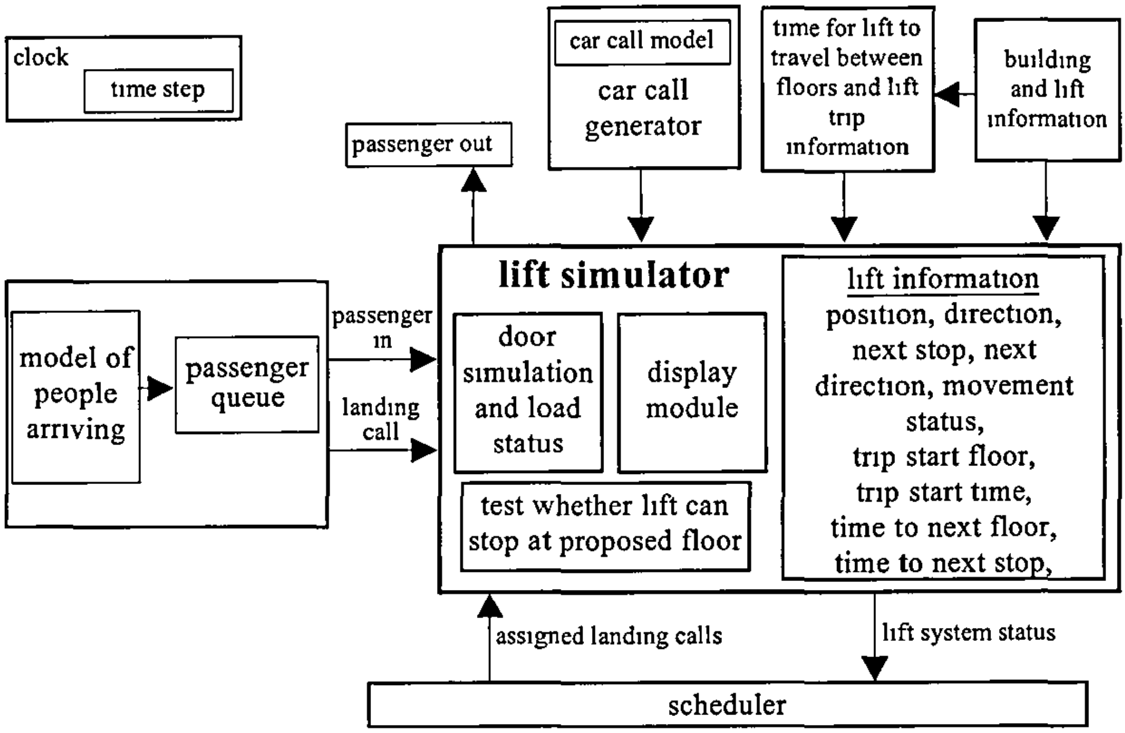


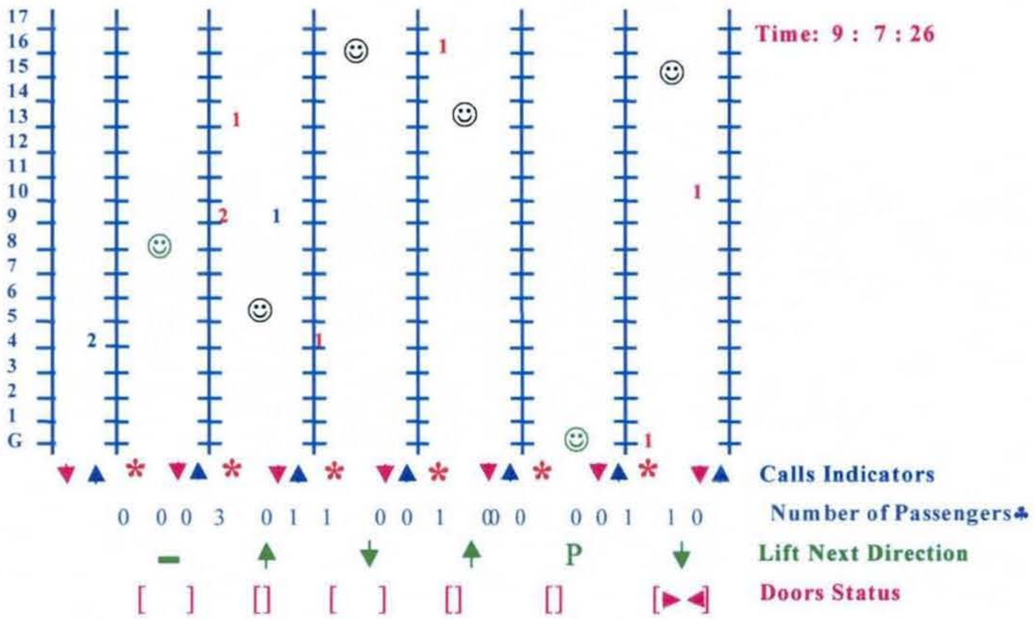
Figure 6.1 The main software modules of the lift simulator.

6.1 General Introduction to the lift simulator main functions.

The simulation should give as close a resemblance to actual world lift systems as possible Figure 6.1 shows the main software modules of the lift simulator. The simulator is configured during an initialisation stage in order to define information about the building, namely the number of floors, floor heights, lift speed, acceleration and door timings The simulator normally begins at the start of the day when there is little or no activity in the building and assigns the individual lifts to pre-defined parking floors ready to receive the early morning traffic. In the simulator, the following sequence of events occurs for each passenger. Following arrival, a landing call is generated, the scheduler assigns a lift to answer the call and the car doors open As passengers enter they press their destination buttons thereby generating car calls. After the last passenger has entered the lift its doors are closed and the lift moves to answer the car calls. At each time step, the lift system simulator state is updated and the scheduler receives information of landing calls, lift car calls, lift position, lift loading and door status The time step can be set to meet a range of accuracy requirements (Section 6.3). As the day goes by, the traffic intensity changes through up-peak period, normal activity, lunch peak period, normal activity again and finally down-peak period.

Figure 6.2 shows the visual part of the simulator displaying the lift group activity in an 18 floor Kodak building. The first column represent the floor numbers, and the second column is divided into two sub-columns, that show the number of passengers behind a landing call at each floor, one sub-column shows passengers travelling in the down direction and the other for up direction. The remaining columns represent the individual lift shafts and the rows represent the floors. Each lift shaft is divided into 4 sub-columns. The first row below the lift shafts shows the function of each sub-columns, * indicate car calls sub-column, ↓ assigned landing calls down and ↑ assigned up landing calls. The numbers shown in the shafts themselves represent the number of passengers behind an assigned landing call or a car call at each floor. The total number of passengers in a car and the total number of passengers assigned to a lift car in each direction are displayed in the same sequence, in the second row below the lift shafts. The remaining two rows show the lift's next direction and the door status. Lift movement between floors is indicated in the simulator by scrolling the lift symbol ☺ within the shaft, and the lift load status is indicated by the colour of the lift symbol A clock displaying the time of day is shown at the top right of the screen; the user can freeze the display to permit closer inspection of the scheduler's operations. The simulator display is capable of showing an installation having a maximum of 18 floors with 6 lift.

When the lift is assigned a landing call as its next destination it starts moving towards that floor Jerk, acceleration, maximum speed and floor heights are used to determine the time a lift takes to make interfloor journeys, and these are saved in a lookup table (Figures 6.1 and will be further explained in Section 6.2.1). While a lift is moving, the scheduler might assign other landing calls to the lift provided that the lift has sufficient time to decelerate and stop. As the lift stops at a floor, its doors start opening.



♣ For each lift shaft, the number of passengers in a lift is displayed under the * symbol followed by the total number of passengers assigned to travel in the lift, first in the down↓ direction and then in the up↑ direction.

Figure 6.2 Example of the lift simulator visual display.

Figure 6.3 shows an example of door states and timings. Each time a passenger gets into the lift, the counters on the display are updated and a car call is generated in the direction of the landing call. As passengers leave the lift, the doors remain open in dwell state until a passenger arrives and a car call is issued (or until the lift is reassigned by the scheduler). Otherwise, at the end of the dwell period, the lift doors begin closing While closing, if further passengers arrive, then it is assumed that they press the landing call button and the

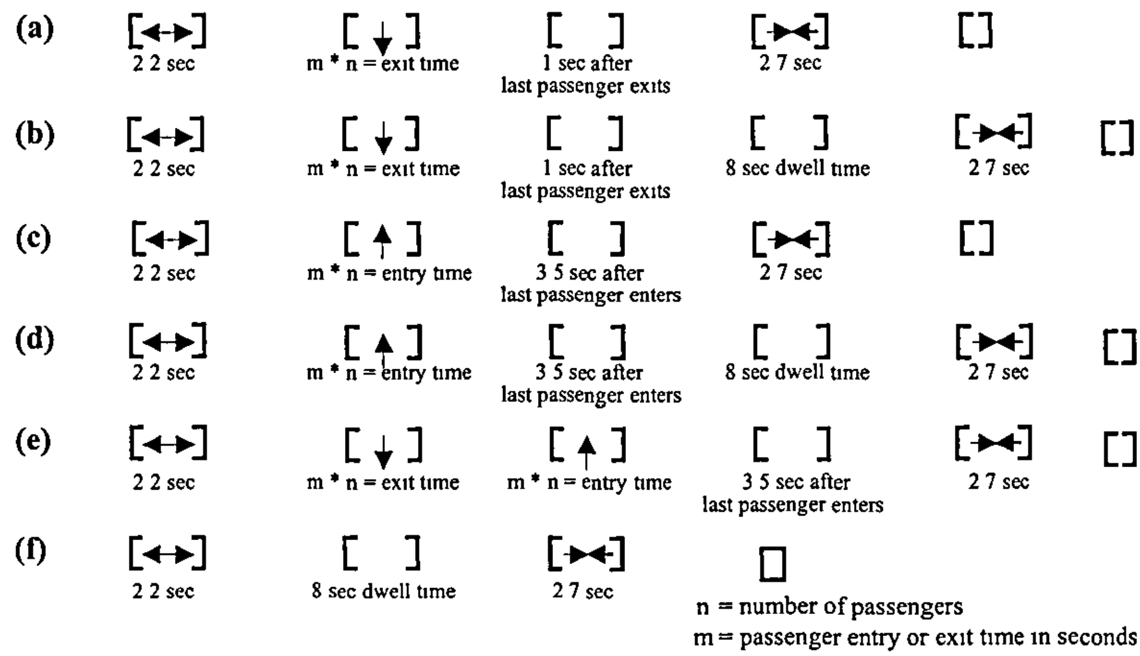


Figure 6.3 Door states and time intervals for (a) normal passenger exit, (b) passenger exit but the lift is not further assigned by scheduler, (c) normal passenger entry, (d) expected passenger entry but no car call is issued, (e) normal passenger exit and entry, and (f) free lift sent to parking landing.

doors reopen as long as the number of passengers in the lift car is less than the maximum number allowed. The dwell period and the passenger's entrance and exit times are usually set at the beginning of the simulation, but, if required, these can be varied by the scheduler to suit a particular traffic intensity requirement.

The following can all be monitored for performance assessment purposes· passenger arrival time, landing call time, landing call cancellation time, door opening time, loading time, passenger car call, door closing time, car departure time and transfer time. The validity of a traffic pattern under the influence of a given scheduler can be assessed by comparing the outputs of the simulator with those of the actual system.

6.2 Simulator functions

This section describes the main functions of the simulator shown in Figure 6 4. After initialisation, the lift system simulation time starts at 7am with low intensity traffic which builds up gradually as passengers begin to arrive. The simulator operates as a continuous loop, the time taken for each loop is a time step of 50ms. The choice of the value of the time step is explained in Section 6.3 Following initialisation, the simulator at each time step carries out the following:

1. Update landing call assignments: checks for new landing calls assigned by the scheduler;
2. Generate passengers and landing calls: detects passengers arriving and landing calls being issued,
3. Simulates passenger entry and departure, car call generation, lift movements, door operations, and updates the status of the lifts.

The following sections describe the simulator functions in details.

Figure 6.4 Lift simulator main function sequence and affected data arrays.

6.2.1 Trip time calculation and trip information arrays

The trip information is required by the simulator in order to determine the position of the lift car, if the lift car is about to stop and if the lift car can stop at a proposed floor. The trip information is calculated at the initialisation stage (see figure 6.4) using the building specification (Appendix A) and the lift speed profile, Figure 6.5 The calculations required at each stage (namely, jerk, acceleration, and full speed) are performed at intervals determined by the time step currently used by the lift system.

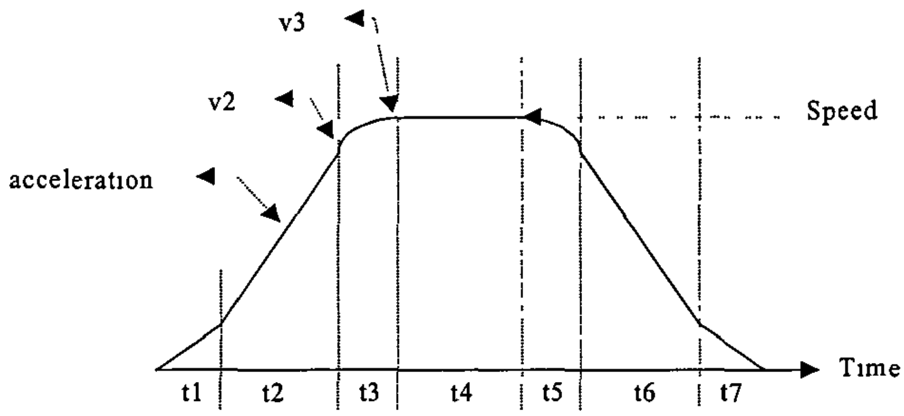


Figure 6.5 Lift trip profile giving (velocity/time) in (m/s).

The following is a sample of the calculation results of the trips between the terminal floor and the 2nd floor sampled at a time step of 0.05 seconds, where

t1	**Jerk interval (s)**	**d2**	**Distance travelled by lift during t2 (m)**
t2	**Acceleration interval (s)**	**d3**	**Distance travelled by lift during t3 (m)**
t3	**Jerk interval (transition to full speed) (s)**	**v2**	**Lift velocity during t2 (ms^{-1})**
t4	**Full speed interval (s)**	**v3**	**Lift velocity during t3 (ms^{-1})**

From floor	To floor	Trip distance (m)	Trip time (s)	t2 (s)	d2 (m)	v2 (ms^{-1})	d3 (m)	v3 (ms^{-1})
0	2	8 100	6 214	2 107	2 746	2 357	1 262	2 607

Full speed time = t4 = 0.000s (lift does not reach full speed for this trip distance)

Time until deceleration = (t1+t2+t3+t4) = 2.607s

Car position floor	Bypass floor time (s)
1	3 350
2	6 214

Further examples of trips can be found in Appendix B

In order to provide the scheduler with information such as lift position and deceleration time, trip information arrays is used. For each journey a lift may make between any pair of floors, a pass by floor time array store the time at which a lift will pass a given floor. A lift may pass a floor at any stage of its journey speed profile and whether it reaches full speed or not depends on the distance travelled Therefore at each time step, the distance travelled is calculated and compared to the next floor distance. In addition, for each trip between floors that a lift can make, information such as the time it takes a lift to accelerate t2, to travel at full speed, and the time interval since the start of the journey until deceleration are also saved in a lift trip information array. This information is used to test whether a lift can stop at a proposed floor, at which point the scheduler can decide if a lift is available for such a reassignment. If the lift is available to stop at the proposed floor then the trip information arrays pointers change according to the newly assigned destination floor

The time consumed by the scheduler to produce a schedule is considered so that a lift car's ability to stop is tested at the time a schedule is produced.

6.2.2 Updating landing call and passenger assignments

At each time step the simulator checks if there is a new assignment from the scheduler. A new assignment is supplied from the scheduler either on a regular time basis or following a significant change in the lift's state (such as a new landing call or car call is generated) that requires a new schedule Therefore *assignment time* is defined as the time interval between scheduler assignments, which depends on the schedule calculation time (which is the time the scheduler requires to perform calculations), and a time interval T, where

$$\textbf{assignment time} = \textbf{schedule calculation time} + \textbf{T} \qquad (6.1)$$

T is either fixed or varies depending on changes in schedule calculation time

As the simulator and scheduler are running in serial rather than in parallel, the simulator runs for a time equal to the current assignment time before the assignment is updated. Assignment time as defined in Equation 6.1 was used for the simulation results discussed in this chapter with T=150ms when the schedule calculation time is no more than 100ms and T=50ms when the schedule calculation time is more than 100ms, Figure 6.6. Appendix C details the scheduling time log for the simulation results, which varies between 0ms and 110ms, resulting in the time between scheduler calls being approximately 200ms.

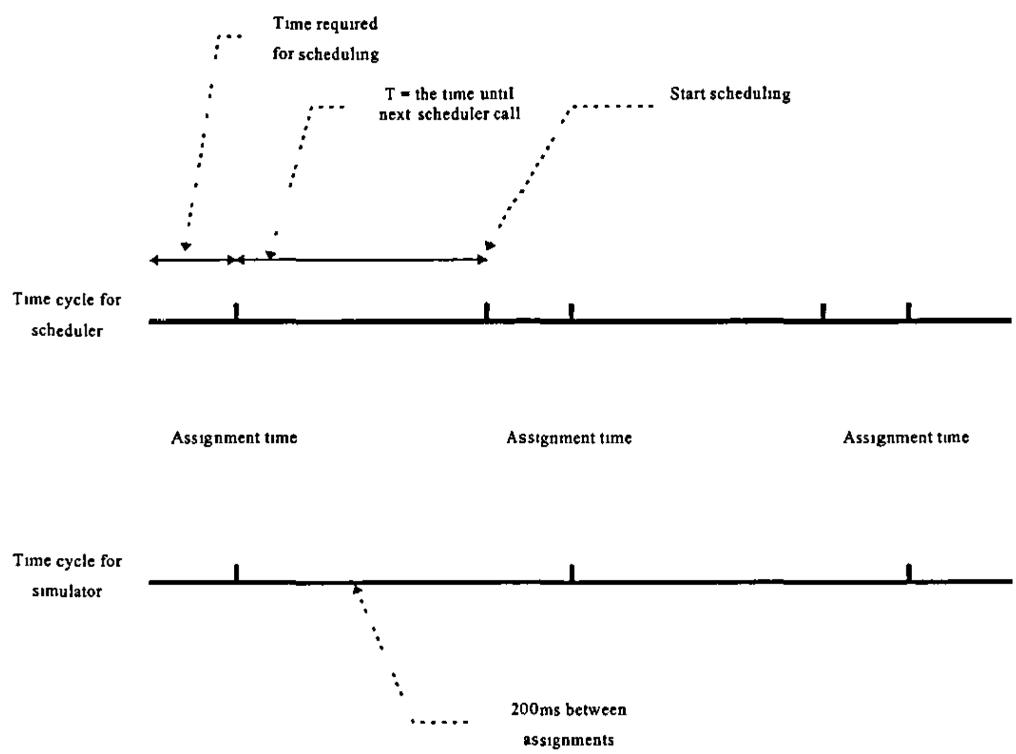


Figure 6.6 Time between scheduler calls used for the simulation results in this chapter.

For the simulation results in Chapter 5 and 8 the time between scheduler calls is fixed at 250ms, while the assignment time depends on the schedule calculation time, as illustrated in Figure 6.7. Consequently, the scheduler calculation time has to be less than 250ms

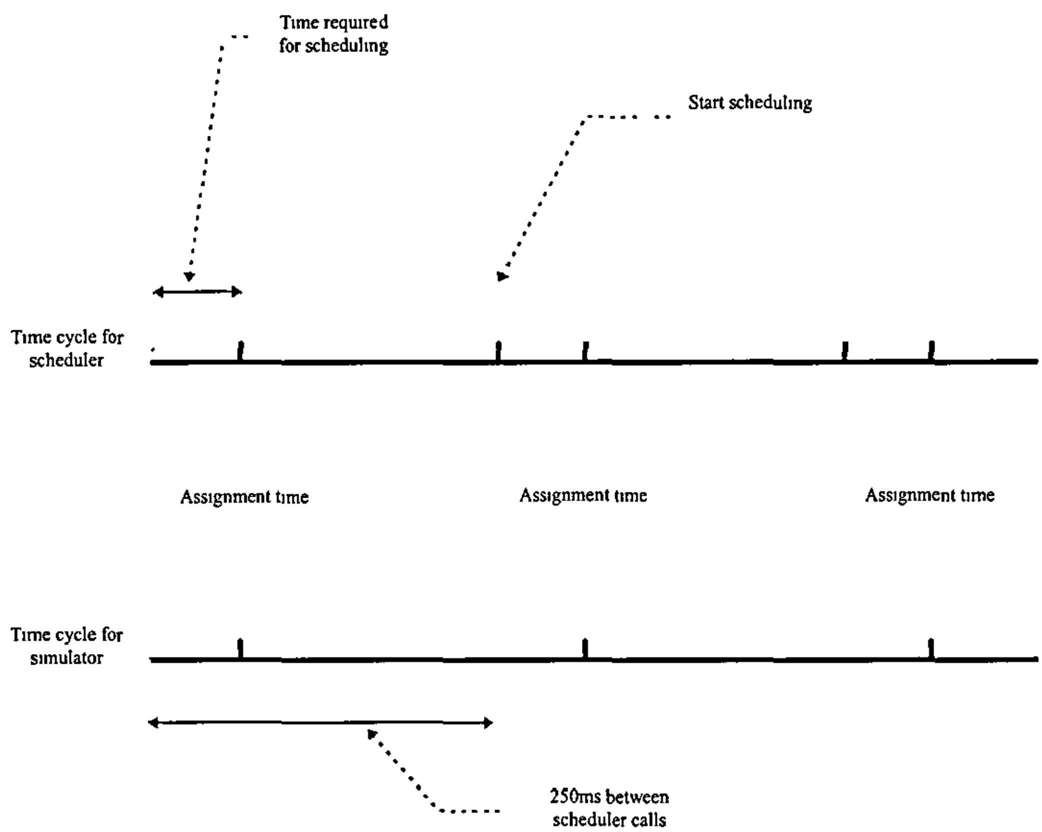


Figure 6.7 Time between scheduler calls used in simulation results in Chapters 5 and 8.

The simulator checks for the following, Figure 6 4.

- Reassignment of previously assigned landing calls
- Assignment of new landing calls.
- False landing calls generated by either the scheduler or the monitoring system.
- If the assigned lift is already at the same floor as the landing call assigned to it, then the assignment would only be instigated if, following consideration of the lift status, it is possible for the passengers to enter the lift.

As a result the following arrays may be updated.

- The landing call array indicates the number of passengers behind landing calls at each floor. Following each assignment, the number of passengers waiting behind a landing call is transferred to the assigned passenger array of the lift car assigned to respond to this landing call

- If a landing call is assigned to a lift parked at the same floor, then the landing call is cancelled to indicate to the scheduler that a lift car responded to the landing call, therefore further reassignment is not required. The landing call is only turned off if the lift door status is *CLOSED*. If it is *CLOSING* then it is only considered if the following conditions are true: the lift car status is *STOP* or it is travelling in the direction of the landing call, provided that the doors have not been reopened before and the load status is below the maximum. The car door opening time is calculated so as to start from the time the landing call is assigned. If the door status is *CLOSING* then the reopening time is assumed equal to the time taken by the lift car doors to reach their current position

- Updating lift destination and lift direction. The next direction (Figure 6.2) follows the direction light indicators that exist in an actual lift installation and which are usually positioned on the top of the lift doors at each landing. As a lift starts decelerating to stop at a landing, the next direction status gets updated to indicate the landing call direction, thus giving priority to those passengers behind that landing call The same occurs if a passenger enters a lift during the dwell state and issues a car call, namely that the next direction immediately reserves the lift to travel in the direction of the first car call registered.

6.2.3 Passenger arrival and landing call generation

After updating assignments, the simulator checks all floors to detect passenger arrivals, figure 6 4 At the start of each time window:

- Update Passenger and car call models: The passenger mean arrival rates and car calls distribution arrays are updated.

- The time intervals until the next arrival are generated for floors which have no previously set interval A floor has no set arrival time interval when no arrivals have been produced during that window of time by the passenger arrival model
- No interval is generated if the time window has no registered arrivals.

The interval until the next arrival is generated using a negative exponential random number generator using a specific mean arrival rate for that floor during that time window:

$$Interval = -\frac{1}{\lambda} ln(U) \qquad (6.2)$$

where U is the output of a uniform deviate random number generator and λ is the mean arrival rate. The time when a landing call is generated is given by

$$passenger\_arrival\_time = Interval + last\_passenger\_arrival\_time \qquad (6.3)$$

If there has been no previous arrival then:

$$passenger\_arrival\_time = Interval + lift\_time \qquad (6.4)$$

where *lift_time* is the simulation time at the end of the current time step. When a passenger arrives, the following is simulated:

- The landing call button is pressed, the landing call time is registered and passenger queues are updated according to a set of rules shown in Figure 6.8.
- The time interval until the next arrival is generated.

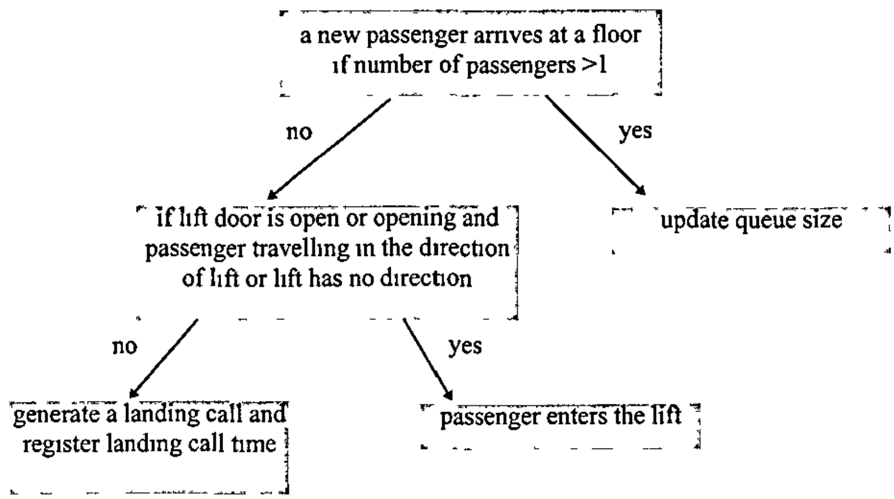


Figure 6.8 The rules used for generating landing calls.

6.2.4 Lift car simulation and car call generation

The final function of the lift simulator is the simulation of lift car movements to their destinations, followed by doors opening to allow passenger entry and exit, passengers issuing car calls, lift doors closing and reopening in response to passenger demand and lifts departing to their next destinations, Figure 6.4.

At each time step, if a lift car is moving then the following actions are performed.

- As a lift car starts decelerating the simulator indicates lift deceleration status. If the lift is decelerating to stop for an assigned landing call, then the landing call is turned off to indicate to the scheduler that the landing call has been answered, while at the same time the next direction changes to indicate the direction of this landing call.
- The lift car position is updated as the lift passes a floor and the indicator of the time when the lift car passes the next floor on its journey is modified.
- When a lift car reaches its destination, the lift status changes to *NOT MOVING* and the door status changes to *OPENING* if passengers are entering, departing, or if the lift has reached the terminal floor Door opening time is considered to start at the time the lift car arrives at the floor.

When the lift is not moving then it is either parked free waiting for an assignment or otherwise the lift car simulator would be performing one of the following actions (Figure 6.3):

- Opening a door in response to passenger arrival or departure. The load status will also need to be checked to prevent doors from opening when the maximum load is reached.
- When the doors are opened, passengers start departing. After all the passengers have exited, new passengers start entering the lift. Prior to each entry, the lift car load status is checked before allowing the next passenger to enter. The time interval assumed for passenger entry or departure is discussed in Section 6.5
- After each passenger entry, a car call is generated indicating the passenger's destination Passenger destination is generated using the car call distribution model discussed in Section 5.4 and a uniform random number generator, which acts as an index to search for the compatible floor frequency that falls in the range indicated by the random number generator output.
- The lift car enters a dwell period if it has not been assigned after passenger departure or no car calls have been generated after passenger entry or if it is parked with the doors open.
- After all passengers have exited or entered, the lift doors remain open for a predefined time interval, Figure 6.3. Both the dwell interval and the interval after passenger exit or entry may be interrupted by new passenger arrivals. The passenger entry time in this case may be less than the defined entry time, as passenger arrival is considered at the beginning of the current time step rather than the actual arrival, which can be any time during the time step. This assumption is made only when simulating passenger entry under these particular circumstances. When a small time step is used, its effect is negligible in practice and adds randomness to passenger entry time. After entering the lift, the passenger would then issue a car call which may not be in the currently assigned direction of the lift.

- During the dwell period or after the last passenger has exited when no landing call is registered, the assignment depends on the car call generated by the first passenger that enters the lift. The dwell interval can also be interrupted by the scheduler when a new assignment is issued, Figure 6 3.
- Only passengers intending to travel in the same direction as that of the lift are allowed to enter. When a lift is parked (with no direction) at a level with its doors open, then once one passenger starts entering, subsequent passengers who require to travel in the opposite direction are prevented from entering the lift. Since the simulator always checks landing calls in the up direction first, if two (or more) passengers arriving to travel in different directions arrive at the same time and the lift direction is STOP, for simplicity and provided that the simulation time step is small, only the passenger(s) aiming to travel in the up direction is allowed in For a simulation time step longer than 50ms, the exact arrival time may be used to distinguish between passenger entry times. The choice of time step is discussed in the following section.
- When all passengers have departed and entered the lift, or at the end of dwell period, the lift doors start closing. While closing, the doors can reopen again for a person travelling in the direction of the lift, provided that the lift doors have not already reopened with passengers in the lift car. Doors will only be reopened if the lift car load status is below the maximum number of passengers
- As the lift doors are closing, then, if there are passengers left when the lift is fully loaded, a new landing call is generated.

$$landing\_call\_time = lift\_time \qquad (6.5)$$

6.3 Effect of time step on simulation performance

A number of activities in the lift system have an associated time counter to indicate the time consumed. This includes passenger entry time, passenger exit time, door opening and closing time. The time counters are used to improve on the resolution normally available if only the time steps themselves were used. When a time counter of an activity is less than *lift_time*, a subsequent activity starting time is considered from the end of the previous activity time rather than from the current *lift_time*, thus preventing any time loss. One example of this, the passenger arrival time counter, is discussed in more detail in Section 6.2.3.

A limitation of the current implementation is that the time step should not exceed the passenger exit or entry time.

Since the simulator and its scheduler work in serial rather than in parallel, the simulator will run for a time interval which is normally chosen to be equal to the scheduling time, (Section 6 2 2). The time step needs to be chosen so that the assignment time is a multiple of a time step.

Clearly, the simulation is more accurate if the time step is small This is illustrated in the passenger entry examples shown in the previous section, and can also be seen when calculating the time when a lift passes by a floor, in which the distance travelled is calculated at the end of each time step and compared to the total distance the lift needs to travel. If the distance is equal or greater than the floor distance, then the lift is considered to be at that floor at the end of that time step. The smaller the time step the more accurate the calculated lift position.

A 50ms time step was generally used for the simulation run which was found also suitable for viewing the lift system activities on the display. The simulation is able to run considerably faster if the display is updated less frequently or switched off

6.4 The ETQ scheduler

A scheduler is required to supply the lift simulator with the lift assignments. The lift simulator is tested by using a simulated version of the ETQ scheduler developed by Mulvaney [1996]. In the Kodak building example, the ETQ scheduler is the lift dispatching system, as explained in Section 2 5, and is responsible for providing landing call assignments and the current destination and direction of each lift. Figure 6.9 shows the main inputs and outputs of the scheduler.

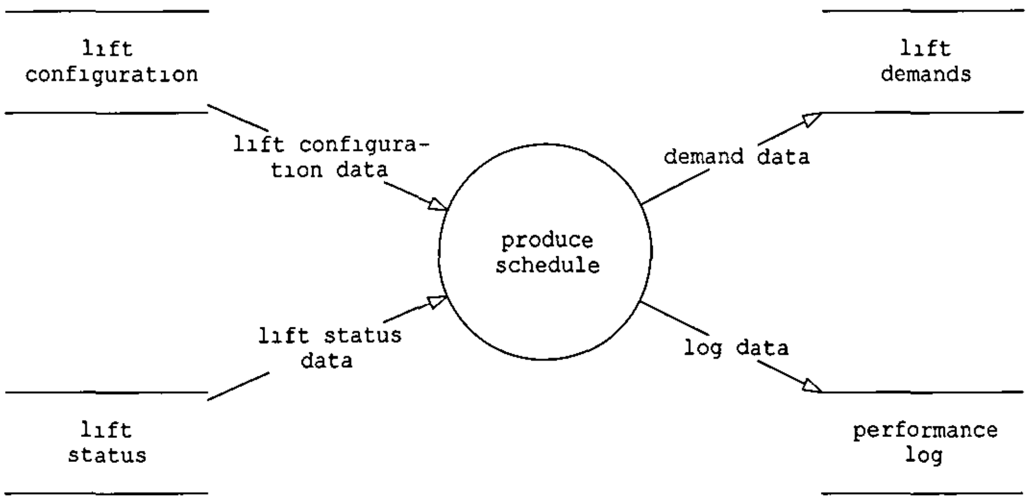


Figure 6.9 Data flow diagram for the ETQ scheduler [Mulvaney 1996].

Figure 6.10 shows the three main processes involved in the derivation of the schedule using ETQ

1. Calculate the paths for individual lifts.

2 Apply merits to determine the composite path.

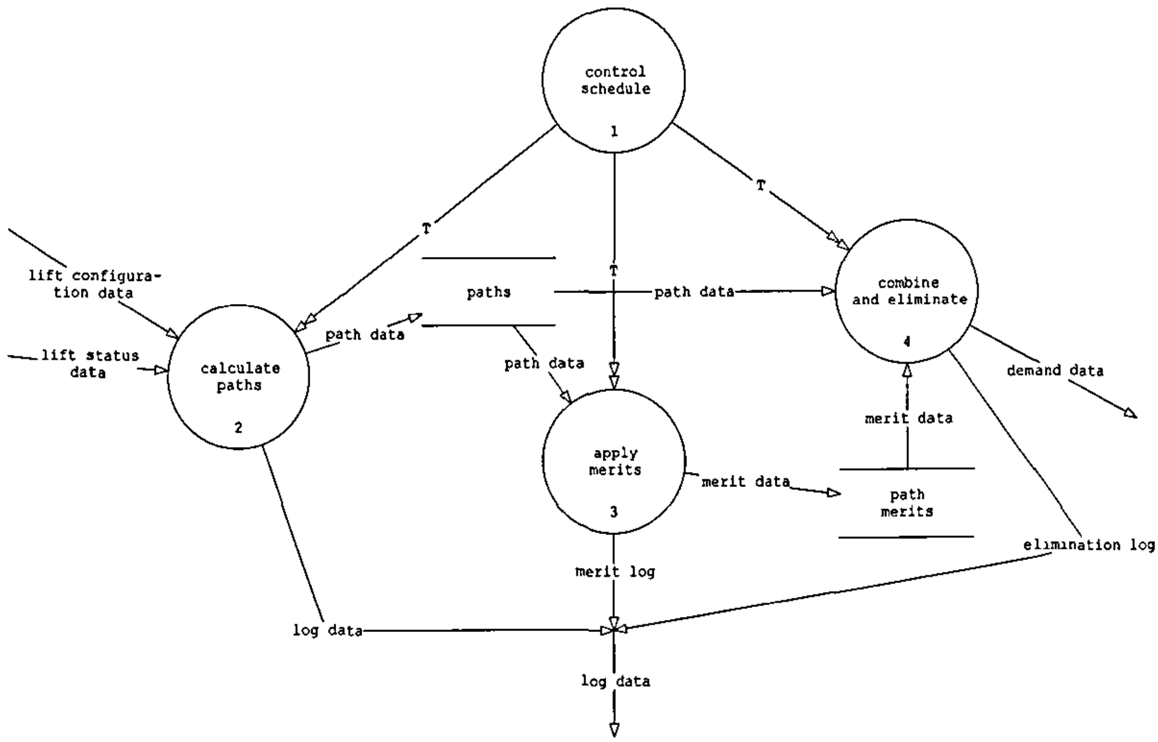
3. Perform elimination to reduce the number of alternative solutions.

The paths are stored in an array of estimated times of arrival (eta) for each car at each floor of the building. The following assumptions are used in the determination of the paths.

- A lift following a path to answer a car call on its way to answering a landing call is favoured.

- For each lift, the full trip path around the building (in both directions) is calculated. The lift answers landing calls in its way of travel and both anticipated landing call and car call delays are considered

- It is assumed that to travel between adjacent floors takes 2 seconds, to answer a landing call takes an additional 6 seconds and to answer a car call takes an additional 4 seconds.



Key: T= Trigger

Figure 6.10 Data flow diagram for the ETQ main functions [Mulvaney 1996].

Once the proposed paths for each lift have been found, a method of assessing the relative quality of the paths is needed The paths with the smallest values of eta are given the highest merits and an elimination process is carried out to produce the final selected paths. For each individual lift, only the five best paths (in terms of travel time) are allocated merits. The merit values are then used in the combination of the paths of the individual lifts to produce composite paths (solutions involving all lifts, which will answer all landing and car calls).

In addition to the ETQ scheduler, the lift supervisory system in the Kodak building employs both up peak and normal traffic parking policies, which are explained in the following sections.

6.4.1 Normal traffic parking policy

Normal traffic parking is the policy followed during non peak traffic periods Using the predicted popularity of floors, a policy using six zones has been implemented by Express Evans Ltd in the Kodak building, as shown in Table 6 1.

Zones	Floors
0	0, 1
1	2, 3
2	4, 5, 6
3	7, 8, 9
4	10, 11, 12, 13
5	14, 15 ,16, 17

Table 6.1 Zones during normal parking period in the Kodak building.

The choice of zones depends on the distribution of popular floors as shown in Table 6 2, where car calls, passengers boarding and photo cell activations collected at each floor from at 7am until 8pm show that the concentration of traffic is in the lower part of the building

Standard parking is activated when there have been no landing calls registered for 10 seconds The system then parks one lift in each of the zones, starting at the lowest zone, with the most popular floor in a zone being chosen as the parking destination. If a lift is already parked in the zone, but not at the most popular floor, then it will not be moved and no other lift will park in this zone

floor	car calls	passengers boarding	photo cell activations
0	1206	1438	1570
1	155	213	202
2	915	1242	1254
3	62	101	97
4	293	361	335
5	229	291	273
6	293	359	335
7	214	263	235
8	205	254	230
9	243	300	285
10	246	303	292
11	245	314	292
12	260	311	285
13	242	297	268
14	264	329	298
15	263	348	325
16	181	220	194
17	118	166	153

Table 6.2 Car calls, passengers boarding and photo cell activations in the Kodak building between 7am and 8pm on 3/10/94 calculated using the rules shown in Figure 5.22.

6.4.2 Up peak strategy

In the Kodak building, a special up peak detection strategy is implemented with up peak being triggered either when the car load exceeds 60% of its maximum capacity, or when four or more car calls are registered in a single lift. This condition is then maintained until the accumulated waiting times of lifts parked with their doors open at the terminal floor reaches 90s During up peak, lift cars may leave the terminal floor with a smaller load and fewer car calls than those given in the triggering conditions, with the aim that they will generally be quickly replaced by other lifts If a car leaves with either one of the triggering conditions intact, then the waiting time counter is reset The up peak strategy adopted in ETQ is that two lifts are sent to the terminal floor as soon as a car call is registered in the lift parked at the terminal floor During travel to the terminal floor, the first lift cannot be reassigned and the second lift is only allowed to answer down landing calls. The implementation of up peak parking by ETQ in the Kodak building is achieved by issuing false landing calls at the chosen parking floor and these stay in effect until the cars are assigned to new landing calls. In the simulator, up peak is implemented by adding weightings which will prevent re-assignments in all practical cases. In addition, if more than one car is free above the ninth floor then the highest in the building is parked at floors 2, 4 or 6 on a rotational basis If the second floor is free, however, a lift will always be parked there.

6.5 Simulation results

As a result of the data analysis described in Chapter 5, models of passenger arrivals and destinations are available for use in the simulation of the Kodak lift system. Figures 6.11 to 6.16 show at five minute intervals both the actual data and the simulation results. Examples from the most popular floors in the building are chosen For the ground floor, Figure 6.11 shows the passenger arrival rate extracted from the actual lift installation using the rules in Figure 5 22 and that obtained from the model by adopting the same arrival rate for a Poisson distribution in the lift simulator, Section 6.2 3.

Figures 6.12 and 6.13 show a comparison between the landing call rates for both the actual lift installation and for those obtained in the lift simulator using the model Figure 6.12 shows the landing call up count from the ground floor with the up peak periods as discussed in Section 6.4.2. Figure 6 13 shows landing call count for the second floor which is the restaurant floor in the building. It can be seen that there is close correspondence between the actual data and simulation results in both curves.

Two examples of the actual and simulated car call frequencies are shown in Figures 6.14 and 6 15. The car call results show close similarity to the original data The simulator uses the passenger destination model discussed in chapter 5, in which each model is the mean of two consecutive windows. If the interval between landing calls is longer than 10 minutes, no car calls can be generated in response using the current window model In such a case, the simulator uses the car call model of the previous window.

Another example of the similarity between the actual data and the simulation result is shown in Figure 6.16, which shows the correlation between the actual mean arrival time and the simulated mean arrival time for every 5 minute window on the terminal floor between 7am and 10am.

Figure 6.17 shows how the door opening interval varies with the number of passengers entering the lift The door opening interval is the interval between the door *OPEN* state and the door *CLOSED* state, which as shown in Figure 6 3, includes a 2.7s door opening time and dwell time of a 3 5s following passenger entry. Figure 6.17(a) shows the results for the actual lift installation. The door opening interval falls in a range between 5s and 20s, while the number of passengers entering ranges between 1 and to 8. A door time interval of more than 20s may result from a number of other factors, for example, a lift may have been parked with its door open long before a passenger arrival. Figure 6.17(b) shows the simulation results obtained using 0.5s as the passenger entry and exit times. The comparison implies that only when the number of passengers entering the lift exceeds 4, the door opening interval gets affected The actual data indicate that the entry time is increased by approximately two seconds for each passenger entering after the fourth. Figure 6.17(c) shows an improvement over the simulation results in Figure 6 17(b), as a result of using 0.75s as the entry and exit times for each passenger entering after the fourth A suitable approximation to the actual data was found by using a time interval of 0.75s for each passenger for the first three passengers entering or exiting, and an interval of one second per passenger when more than four passengers enter or exit. The results of this experiment are shown in Figure 6.17(d). It is important to stress that although only door intervals when passengers enter the lift have been monitored, passenger departure also affects door opening time. The number of passengers departing is not included because of the difficulty of detecting passenger departure from the actual data. Since the number of passengers entering was collected using photocell activations and car calls, this might be also considered as an underestimation. Moreover, the actual data are collected every second which introduces a longer time delay than that when using the simulator. According to the above and considering the difficulty in simulating passenger behaviour, the results in Figure 6.17 represent a suitable practical model.

Figure 6.18 shows the number of passengers boarding the lift and the car calls that are generated in response between 9am and 10am. These results follow from the application of the rules in Figure 5.22 to the actual lift system data, and a comparison is shown with the simulation results when applying the same rules. The simulation results are collected without considering the additional information that the lift simulator provides, for example a car call is only counted if there are no previously generated car calls to the same floor. The comparison indicates that in the simulation results, when a landing call is detected in Figure 6.18(b), more passengers enter the lifts from the second and terminal floor than in the actual data. Note that the largest value is for the terminal floor and the second largest is for the second floor. Again, there is close similarity between simulation results and actual data.

Figure 6 19 compares the actual and the simulated intervals between landing calls. Figure 6.19(a) gives 12s as minimum interval between landing calls. An interval between landing calls can be calculated as:

door opening time + passenger entry time + 3.5s after passenger entry + door closing time

Since the simulator observes the landing calls while the door is closing, the simulated results show time intervals less than 10s. The results demonstrate a close resemblance between the actual and the simulated data The reason that the actual data does not show intervals less than 12s is that there is approximately 2s of delay in data collected from the actual lift system.

The effect of the differences between the simulated and the actual scheduler response are apparent in Figure 6.20, showing the time interval between the issue of a landing call and its cancellation The differences in results are due to the following factors.

- The time interval between scheduler calls is discussed in Section 6.2.2 and in this case the scheduler calls depend on the time required by ETQ to produce a schedule, which is typically 50ms. Figure 6.20 shows the effect of using a range of values of T on the response of the ETQ scheduler, Section 6.2.2. T=150ms is chosen for the simulation results of this chapter.

- A stability strategy has not been simulated. In the actual system, whether a schedule is changed depends on how close is the match between the previous and current solution as explained in Section 2.5. This causes the simulated ETQ to produce new schedules more frequently than the actual ETQ. Figure 6 20 shows that the greater the value of T, the more similar the simulated response is to the actual response
- The actual data have been collected every second, and during this period it is possible that a landing call can be issued and cancelled without detection

Another way of comparing system performance is by calculating the average waiting time, which is defined as the mean interval between the issue of a landing call and car arrival. Figure 6.21 shows the waiting time for both actual and simulation results. The average waiting time between 8am and 9am from the terminal floor is 10s, while, for the same example, the simulation result gives around 8s. This may be due to two reasons. The first is that the simulated ETQ has a faster response time than the actual ETQ. The second reason is that the time delay in the actual data collection may make intervals appear longer. This is apparent in both Table 6.3 and Figure 6 22.

Table 6.2 compares the actual and simulated lift trip time, which exhibits difference of around 2s. Figure 6 22 compares the simulated and actual time interval between landing call cancellation and car arrival. The interval should be no more than the deceleration time of 4s and the actual intervals show a delay of approximately 2s.

Figures 6 23 and 6 24 illustrate a comparison between the actual and simulated intervals between car departure and next landing call. The simulation result shows a higher occurrence of time intervals of zero length, and these arise because the simulator is able to detect landing calls while the doors are closing. In such a case, as the car has not departed, the interval between landing call and car departure is considered as zero.

The above results show a close resemblance between actual and simulated lift systems.

Trip between floors		Trip time in seconds	
from	to	actual	simulated
0	12	18	15 971
12	9	8	6 860
12	14	8	5 701
0	15	21	18 843
9	10	5	4 195
0	2	8	6 214
4	5	7	4 195
11	9	8	5 701
3	0	8	7 337

Table 6.3 Comparison between the actual and simulated lift trip time.

6.6 Relative performance of the actual and the simulated ETQ scheduler

As no accurate source of information was available on exactly how ETQ reacts to different situations, the simulated ETQ is only able to follow the general policy of the actual ETQ. The following points gives examples of performance differences

- The ETQ simulator does not use lift status, such as door status and movement status If the scheduler does not sense lift movement then, as the lift starts moving, its recorded position remains unchanged until the next floor is reached As a result, the scheduler can re-assign the lift giving its own car position as a destination In the simulation, the lift has to return back to its original floor, and it needs to decide the next possible floor at which to stop in order to be able to reverse That is why in the actual lift system an advanced lift position is given to ETQ
- Since simulated ETQ does not sense lift next direction status, then it is possible for it to assign two landing calls in opposite directions to one lift As a lift decelerates to

answer the first landing call, the landing call is cancelled, thus removing the only indication of the lift next direction. When both landing calls are assigned to the same level, normally the simulator does not reopen lift doors for landing calls in the opposite direction.

- When two or more passengers enter a lift, passengers who wish to travel in the opposite direction to that registered by the first passenger can still board a lift whose door tatus is *OPEN* or *OPENING* when the next direction of the lift is *STOP*. The simulator only informs ETQ of subsequent car calls in the same direction as that of the first car call. When two passengers wishing to travel in opposite directions register landing calls simultaneously, the simulator has to decide which passenger's demand is registered. In such a case, the simulator checks for landing call up demands first and for simplicity the simulator prevents passengers who wish to travel in the direction opposite to that registered from entering the lift.
- The simulated ETQ assigns landing calls to fully loaded lifts. The lift simulator prevents doors from opening when the lift is fully loaded, but the lift will still stop for the assignment. In the actual lift system fully loaded lifts bypass assignments.

The above differences are listed for the information of the reader, but there was no observed impact on the performance of the lift simulator. The main difference in performance that does effect the results is due to the absence of the stability assessment policy from the simulated ETQ scheduler, which causes frequent fluctuations between assignments The stability assessment policy is required in order to limit the switching between different assignments and implemented in the actual ETQ system.

6.7 Conclusion

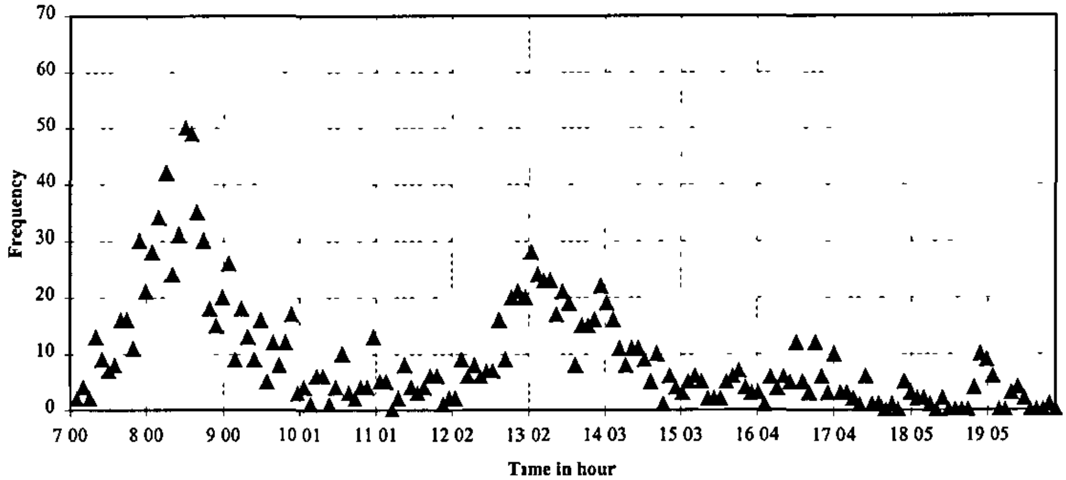
Lift simulation is an essential part of lift system design and testing. In the simulation, actual system data are used to maintain a close imitation of actual lift motions and time delays. The availability of a traffic model can be used as the basis of a technique to eliminate the need to identify separately specific traffic patterns such as up peak and down peak.

In addition to the fact that the simulated ETQ has no stability function implemented, the differences in the results were mainly due to the time delay in the gathering of the actual data. Despite this, the simulation results showed close agreement with those obtained from to the actual lift system.

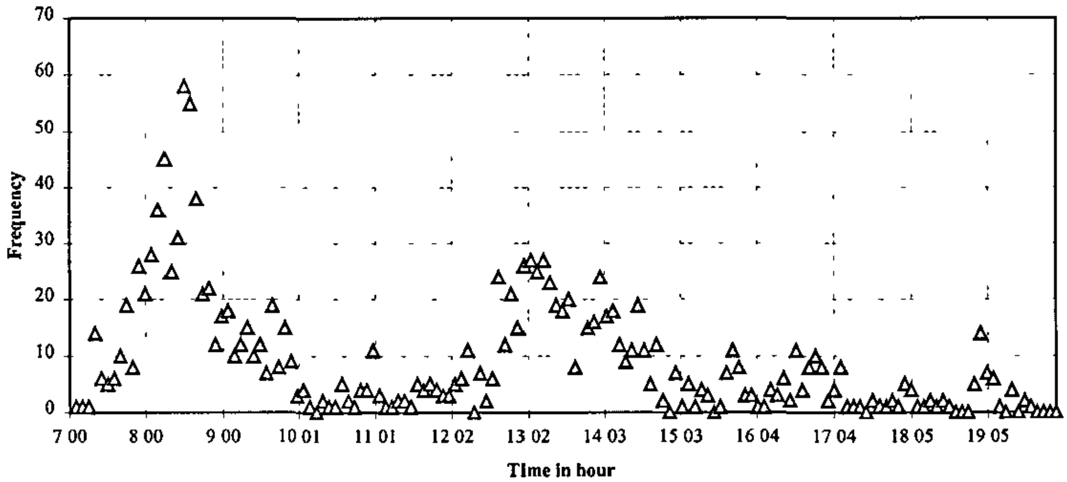
Lift simulation can also be used as a part of an actual-time system to assist the scheduler in searching for the best assignment solution, for example by supplying the expected next lift state configuration on which the scheduler can base its current decision. Also, the performance of the scheduler can be tested under unusual operating conditions, such as when one of the lift cars is out of order. Such conditions can either be induced directly from the keyboard at any time or by registering the time of occurrence before the start of the simulation run. The performance of any scheduler is greatly dependent on the quality of data it receives. The quality of data provided by the lift system can be improved by using simulation results. For example, during simulation, the lift simulator is able to supply the scheduler with better load status information than that usually available from the actual system. This can be used, for example, to assess the importance of attaching more accurate load sensors to an actual lift.

A number of extensions to the current lift simulator system could be made to improve the model of passenger behaviour, since only the important lift simulator functions for testing the scheduler were implemented. Under certain traffic conditions a passenger may choose to board a lift moving in the direction opposite to that of intended travel. Also under light traffic conditions, a passenger might press a button to close the lift car doors after getting into the lift.

In spite of the differences between the simulated and the actual ETQ, the simulation results showed that the performance of the simulator is suitable as a framework for conducting the research, that is studying and implementing the intelligent scheduler.

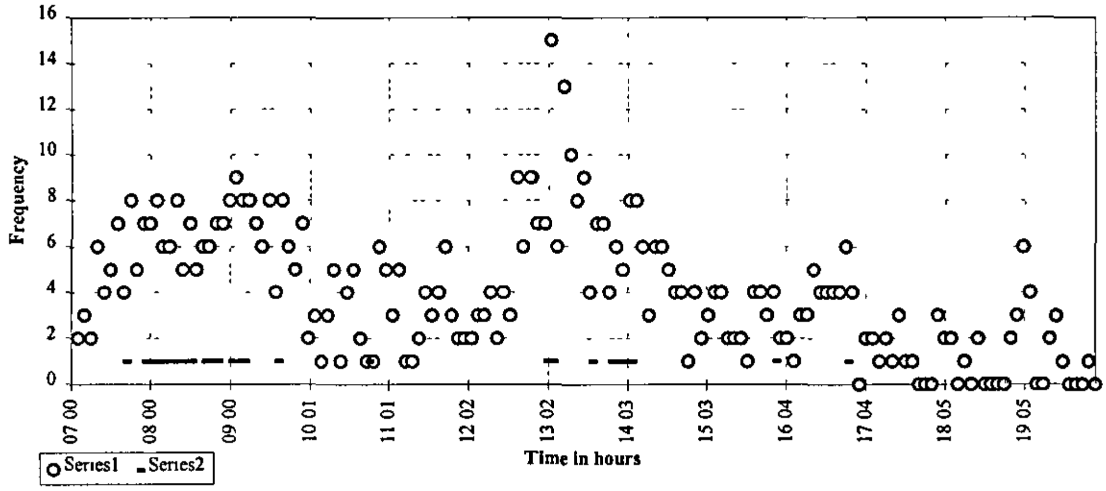


(a) Passenger arrival in a real lift installation extracted using the rules shown in Figure 5.22.

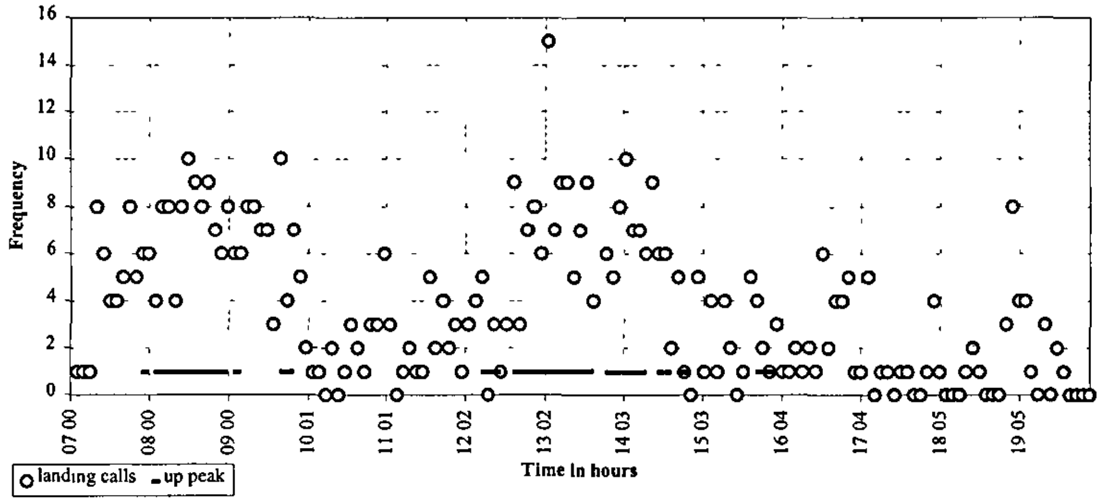


(b) Simulation of the passenger arrival

Figure 6.11 Comparison of extracted passenger arrival with simulated passenger arrival calculated every 5 minutes at the terminal floor between 7am and 8pm on 3/10/94.

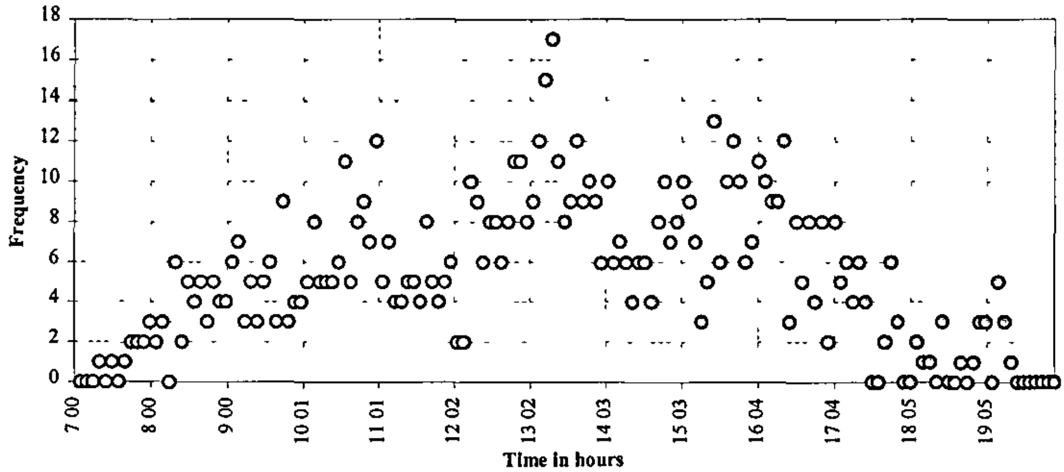


(a) Landing call variation in a real lift installation.

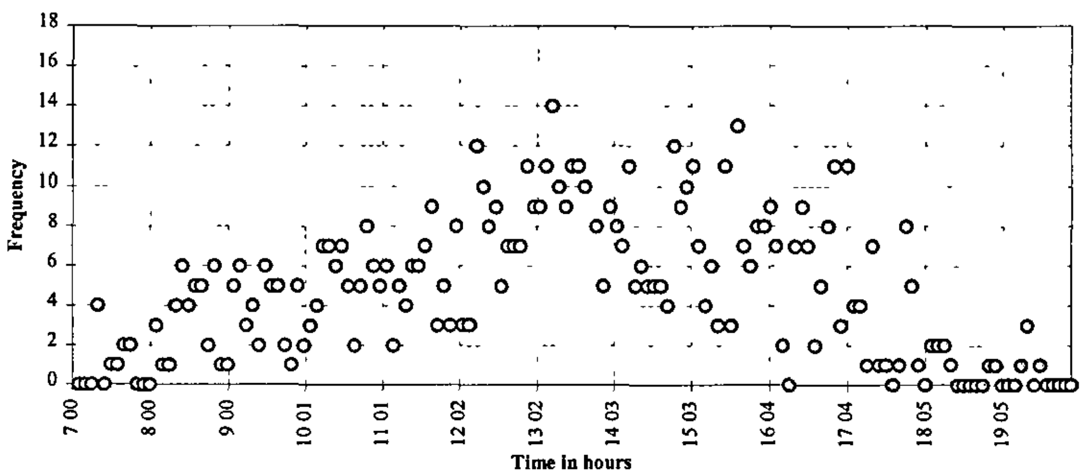


(b) Simulation of the landing call variation in (a)

Figure 6.12 Actual and simulated landing calls calculated every 5 minutes at the terminal floor between 7am and 8pm on 3/10/94.



(a) Landing call variation in a real lift installation.



(b) Simulation of the landing call variation in (a)

Figure 6.13 Actual and simulated landing calls calculated every 5 minutes at the second floor between 7am and 8pm on 3/10/94.

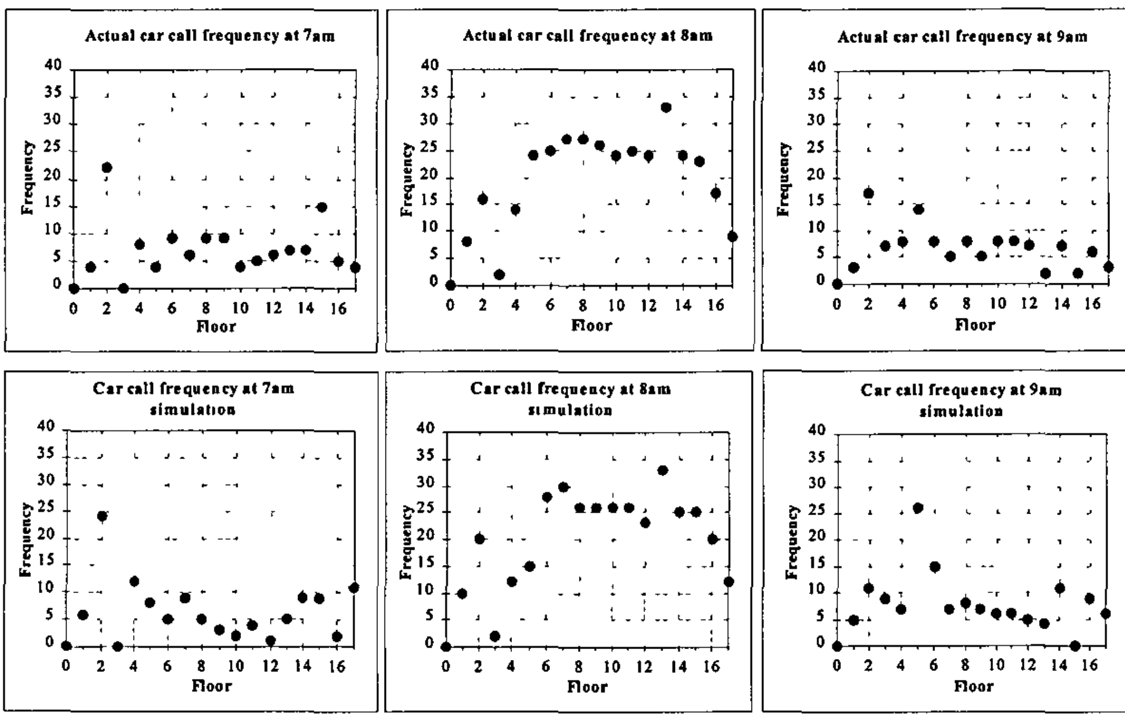


Figure 6.14 Actual and simulated car call frequency distributions from the terminal floor to all floors between 7am and 10am in the Kodak building on 3/10/94.

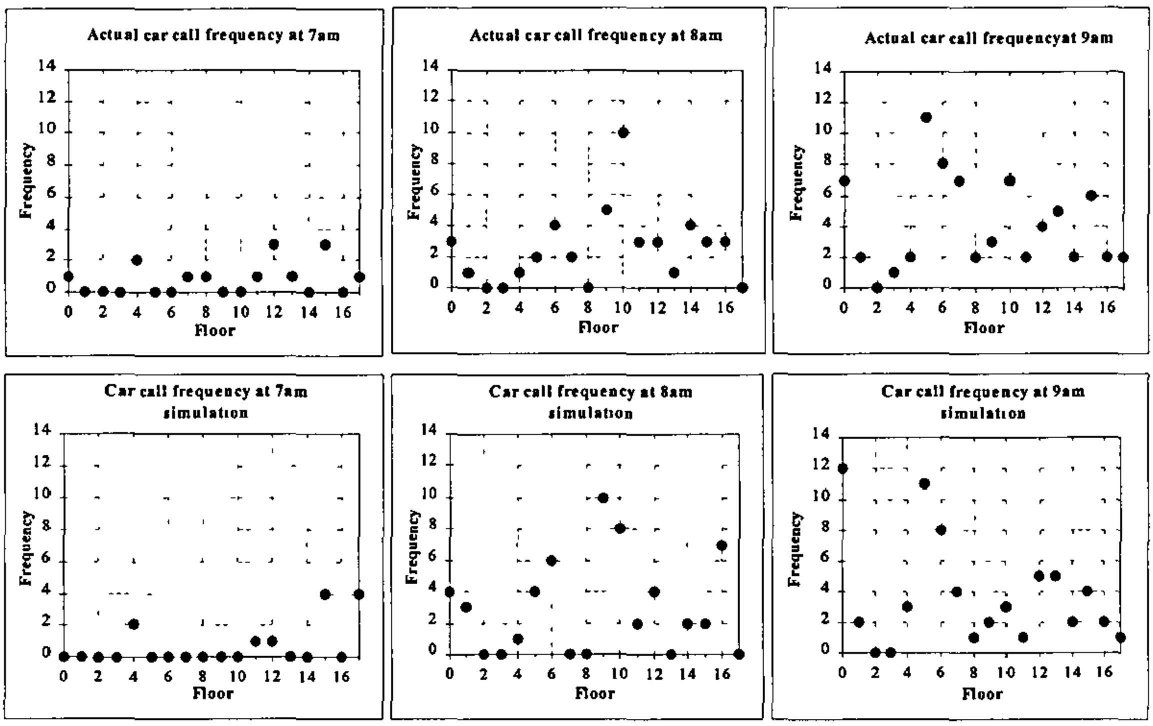


Figure 6.15 Actual and simulated car call frequency distributions from the second floor to all floors between 7am and 10am in the Kodak building on 3/10/94.

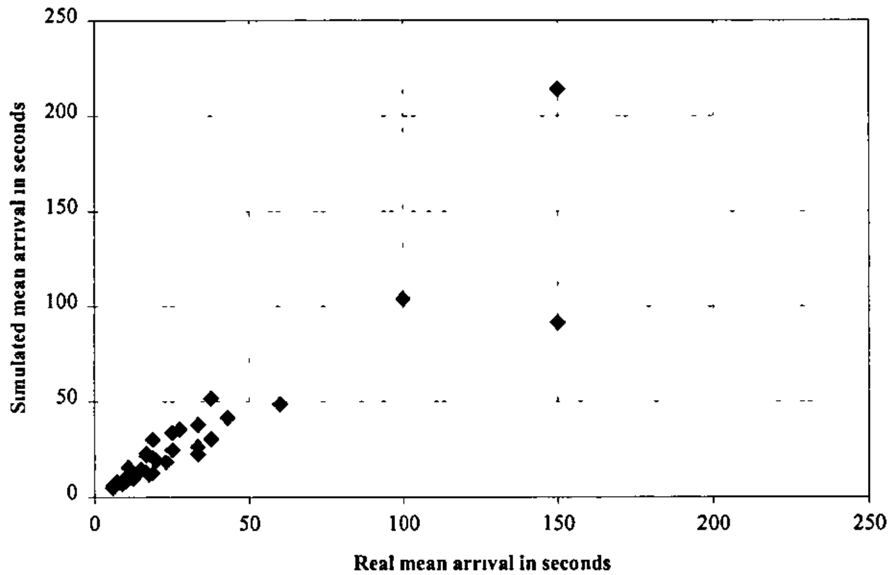
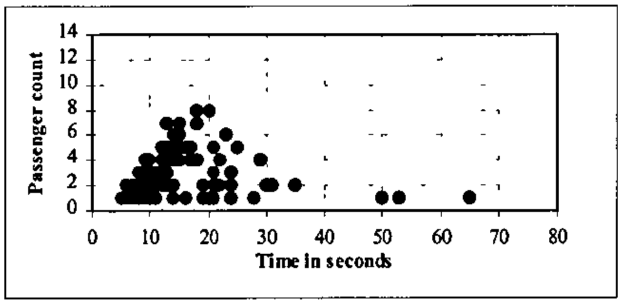
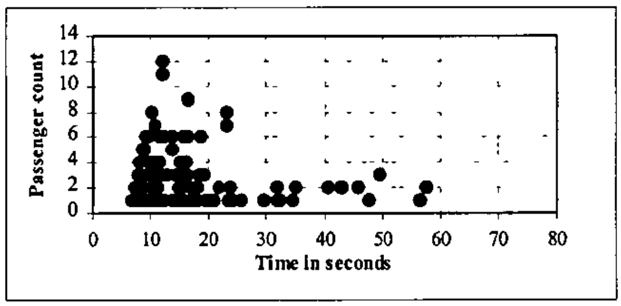


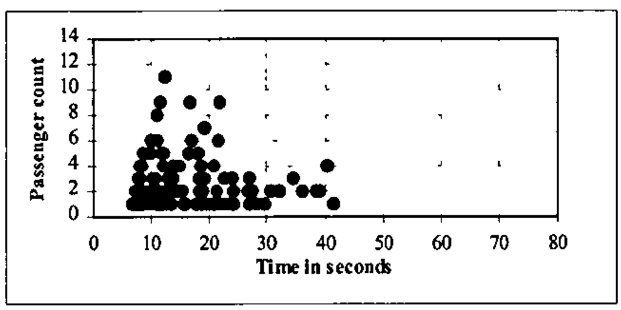
Figure 6.16 Simulated against actual mean arrival rates for each 5 minutes window between 7am and 10am at the terminal floor on 3/10/94.



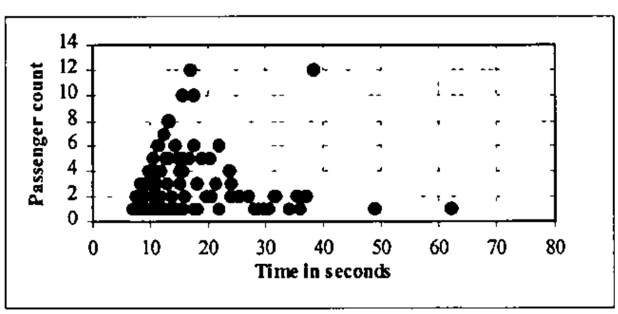
(a) Actual lift installation



(b) Simulation results when passenger entry and exit time are 0 5 second.

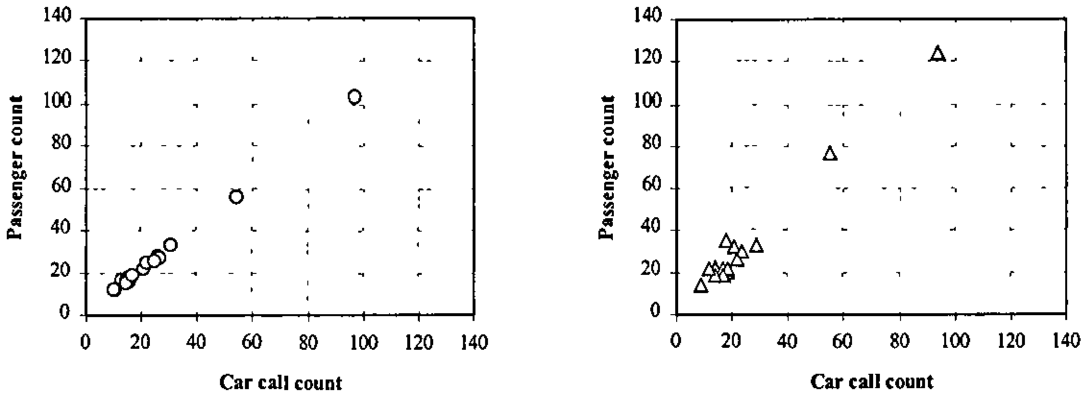


(c) Simulation when passenger entry and exit time are 0.5s for the first 4 passengers in the queue, increased to 0.75s when there are 4 or more passengers in the queue.

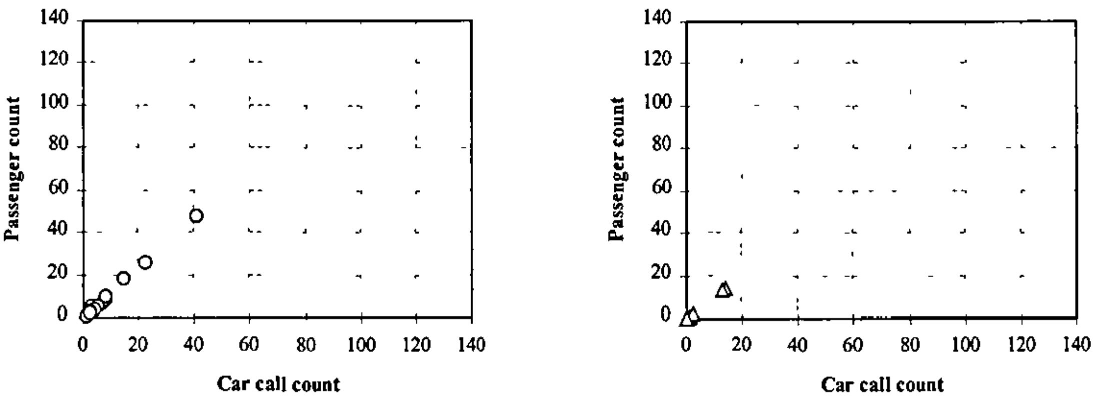


(d) Simulation when passenger entry and exit time are 0.75s for the first 4 passengers in the queue, increased to 1s when there are 4 or more passengers in the queue

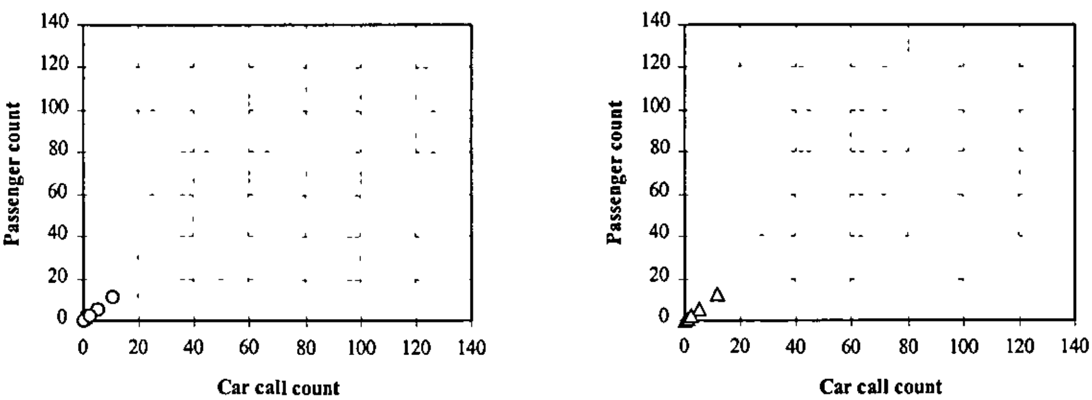
Figure 6.17 Door opening time (measured from door OPEN status until door CLOSED) as a function of passenger boarding numbers at the terminal floor between 8am and 9am on 3/10/94.



(a) In response to landing call when no passengers departing lift.

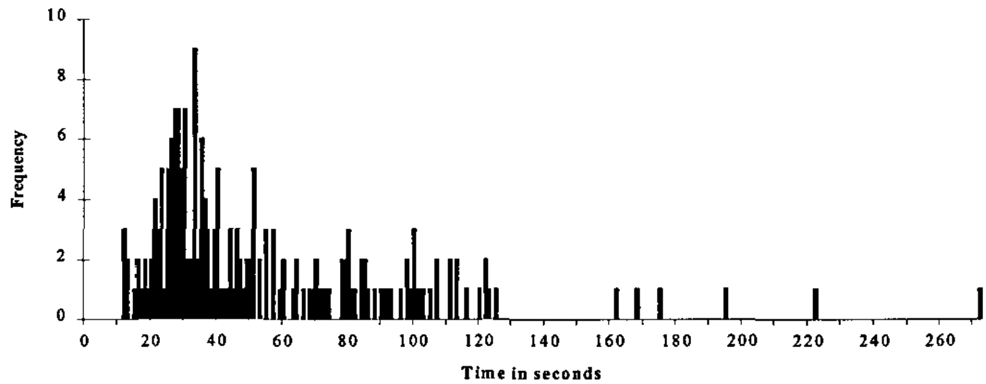


(b) In response to landing calls when passengers departing lift.

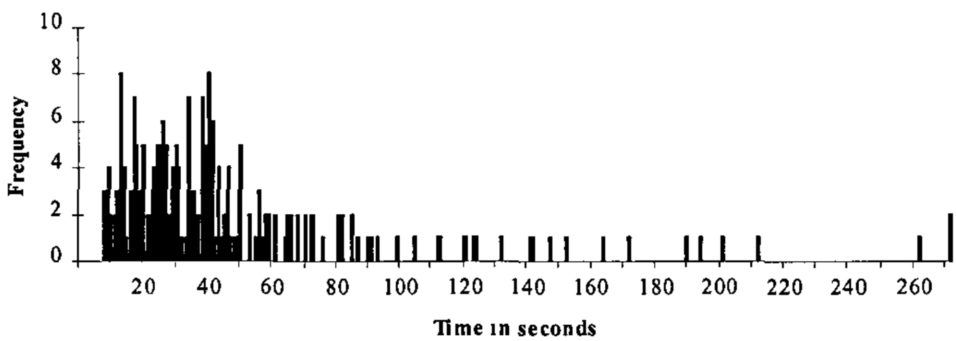


(c) No landing calls issued, when passengers departing lift.

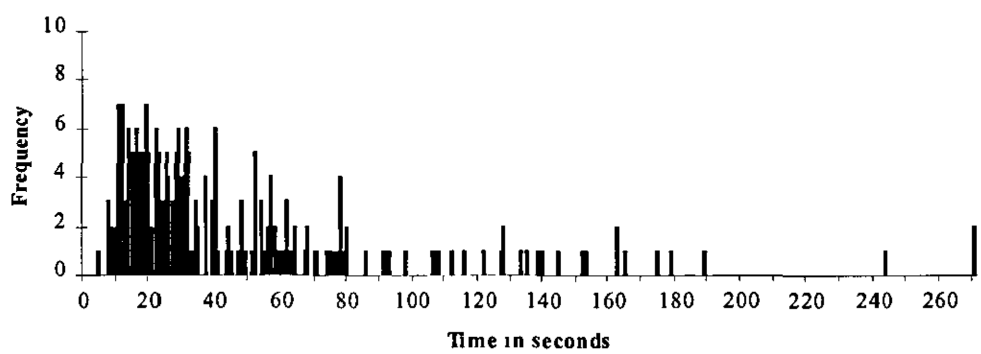
Figure 6.18 Passenger boarding count against car calls, for all floors between 9am and 10am. The right column shows actual data and the left column shows the simulated data.



(a) Intervals between landing calls in actual lift installation.



(b) Intervals between landing calls from lift simulation.



(c) Intervals between landing calls from another simulation run

Figure 6.19 Actual and simulated intervals between landing calls at the terminal floor between 7am and 10am, T=150ms, 0.75s passenger entry and exit time for the first 4 passengers in the queue, and 1s entry and exit time for each passenger entering after the fourth.

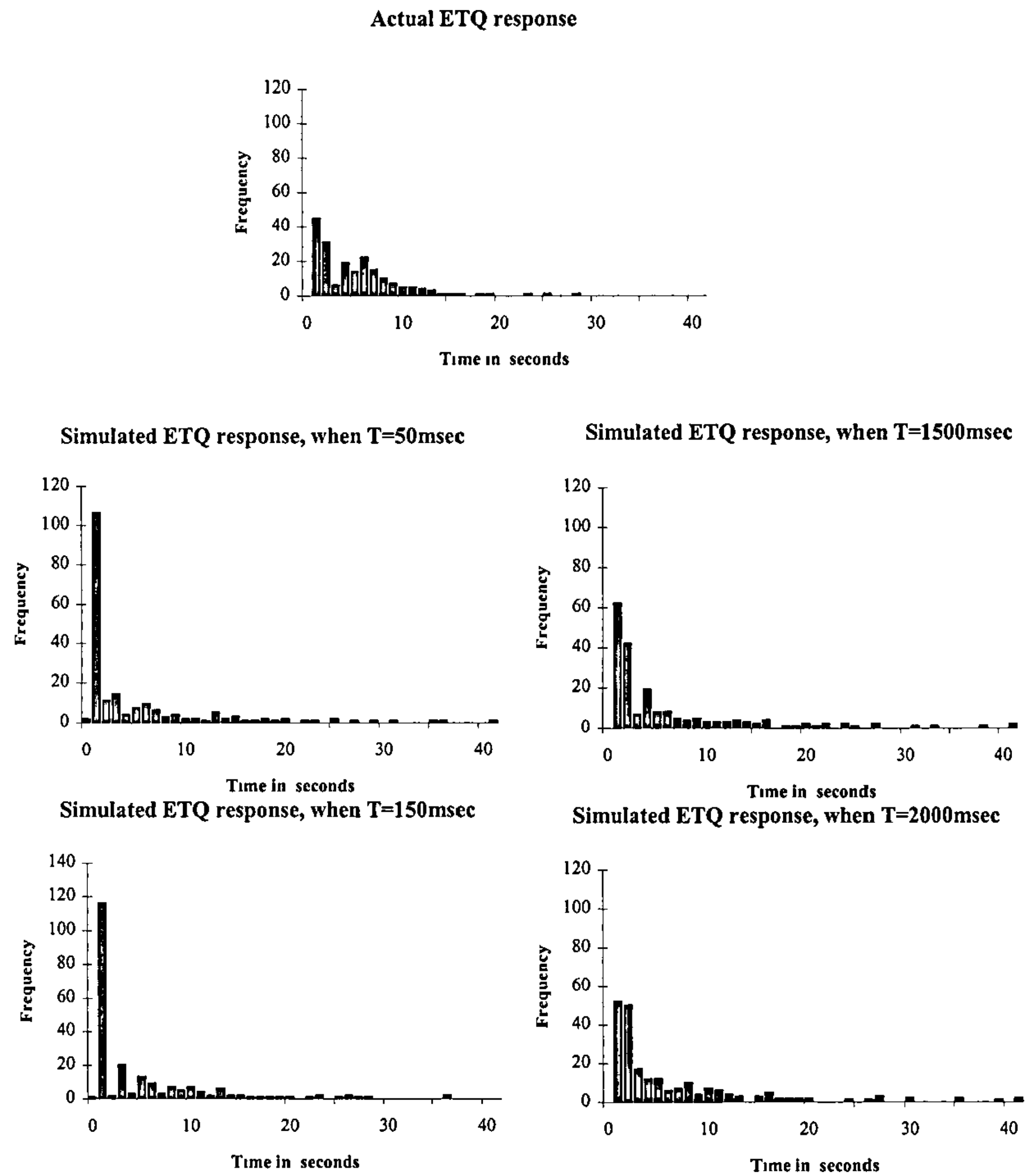
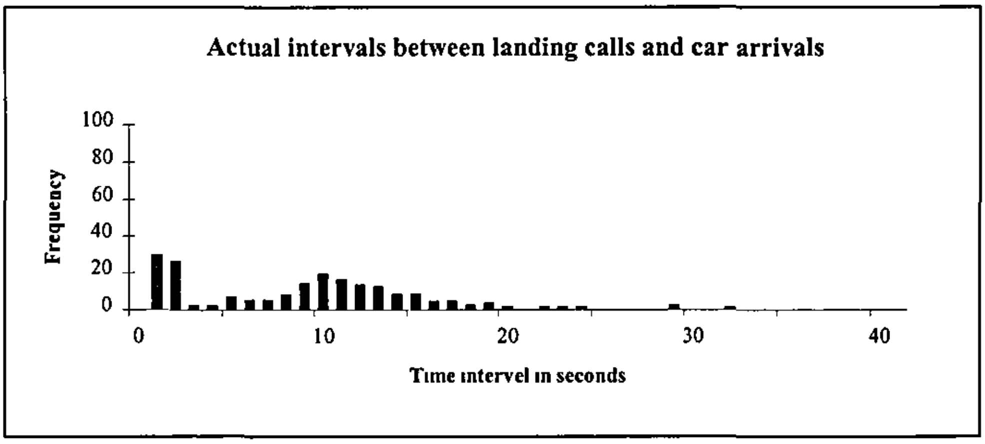
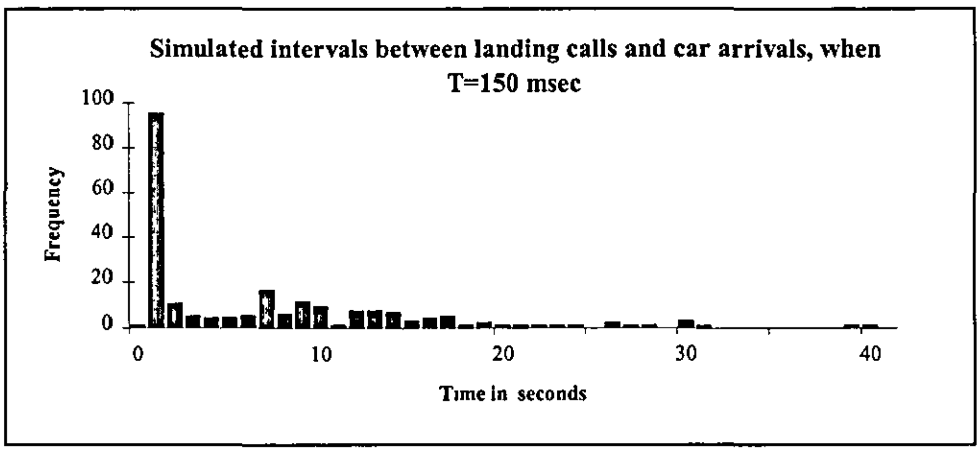


Figure 6.20 Actual and simulated distribution of the intervals between landing calls and their cancellations at the terminal floor between 7am and 10am on 3/10/94.





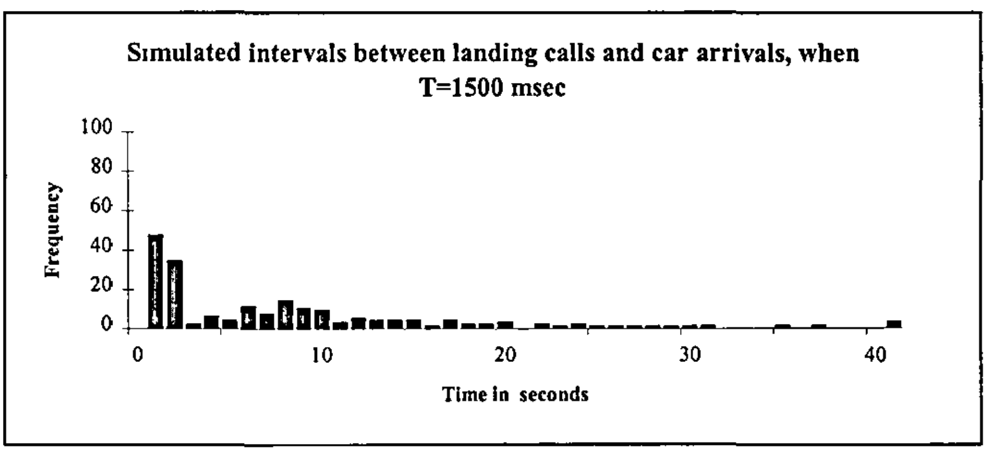
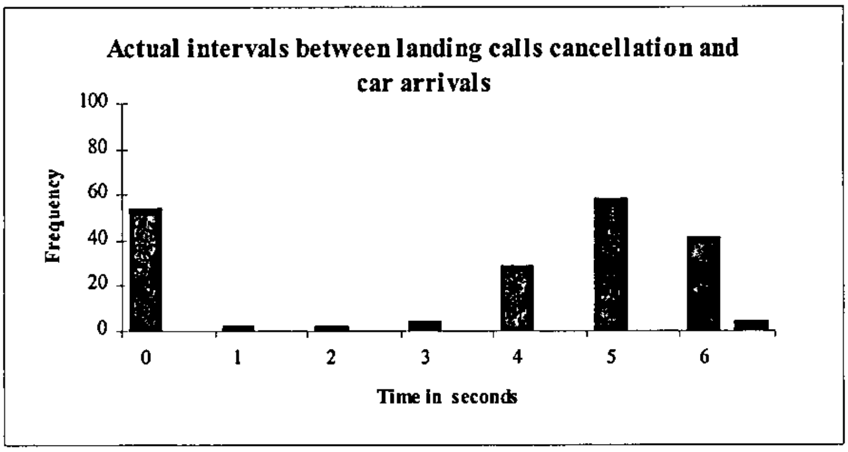


Figure 6.21 Actual and simulated intervals between landing calls and car arrivals at the terminal floor between 7am and 10am on the 3/10/94.



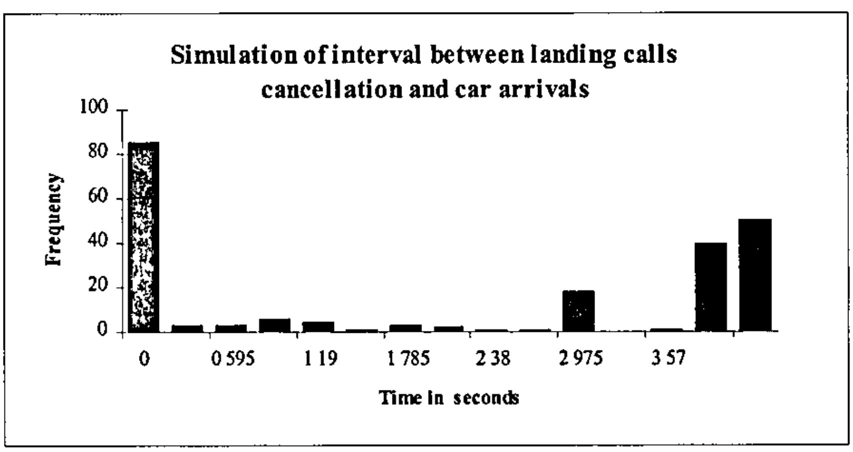
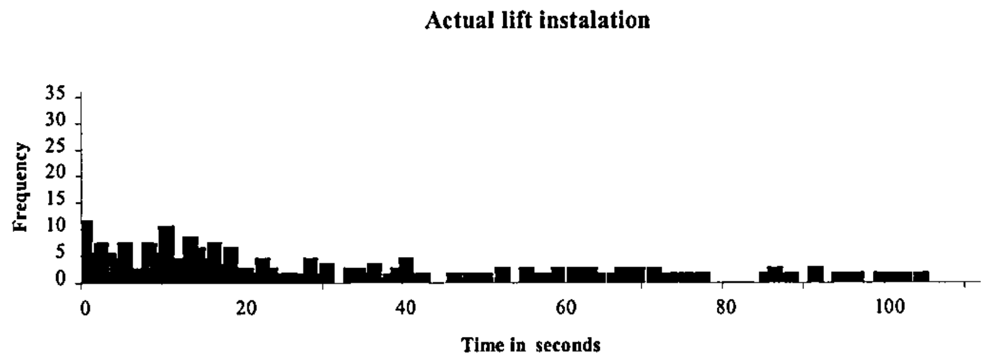


Figure 6.22 Actual and simulated intervals between landing calls cancellation and car arrivals at the terminal floor between 7am and 10am on 3/10/94.



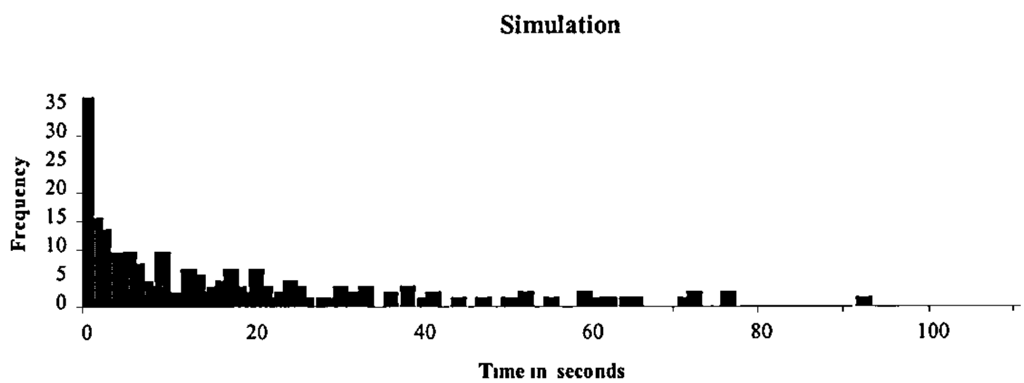
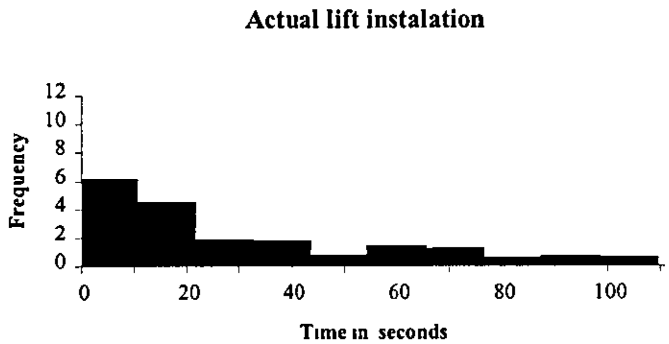


Figure 6.23 Actual and simulated distribution of intervals between car departures and next landing calls at the terminal floor between 7am and 10am on 3/10/94.



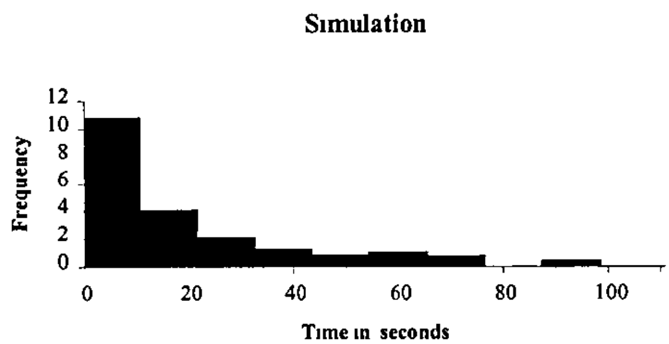


Figure 6.24 Smoothed distribution of intervals between car departures and next landing calls at the terminal floor between 7am and 10am on the 3/10/94.

Chapter 7

Intelligent real-time lift scheduler

The proposed lift control system is divided into two principal parts, namely the real-time scheduler and the monitoring and prediction system. This chapter is concerned with the design and implementation of the real-time scheduler. The scheduler needs to be able to respond within a limited calculation time in order to minimise waiting time. In the proposed solution, intelligence involves the suitable use of information about the current and predicted future states of the lift system.

In the following sections the following details are explained.

- The definition of the lift scheduling problem is given in Section 7.1
- An analysis of the use of heuristic search to solve the problem is discussed in Section 7.2.
- To demonstrate the operation of the intelligent scheduler, two representative examples are introduced in Section 7.3 whose implementation involves simplifying assumptions, especially the use of fixed delays.
- To explain the proposed search method used in the real-time lift scheduler and its implementation Section 7.4 gives the assumptions used in the implementation of the intelligent scheduler, Section 7.5 discusses the implementation of intelligent scheduler based on A* search and Section 7.6 shows the results of testing the intelligent scheduler and a third example is then given which incorporates more realistic assumptions regarding door delays and passenger entry and departure timings.
- The effect of using landing call waiting time when calculating the time the lift will take to answer a landing call is described in Section 7.7.

The scheduler developed as a result of this work is tested in Chapter 8 using the lift simulator, and the simplifying assumptions are replaced one by one by the true dynamic values which can be obtained from the monitoring system

7.1 Definition of the problem to be tackled by the intelligent scheduler

Following the discussions of Chapters 2, 3 and 4, the detailed requirements of the real-time scheduler can now be stated.

- At any current state the scheduler action is dependent on incomplete future state information For example, when calculating the path cost of a lift reaching a landing call, car call positions and car call delays can only be estimated.
- For n lifts and p landing calls the scheduler has n^p possible solutions. The scheduler has to search for the best solution, within a time duration of 250ms (Section 6.2.2). Even under light traffic conditions, an exhaustive search providing optimum results may take considerably longer than this Hence, the scheduler must implement a technique which requires the examination of a restricted set of possible solutions.
- Any future state is affected by new landing calls, car calls and the schedule produced in the current state. It is difficult to update in an incremental manner the schedule produced in the previous state in order to derive the schedule for the current state, especially if a new landing call has been issued. Such an incremental updating of a schedule may be achieved either by considering all landing calls or by only considering new landing calls. The first method would involve repair and backtracking operations, while the second solution risks the optimality of the schedule, since the previous schedule was produced without considering the new arrivals. Consequently, the approach normally taken is to perform the search from afresh in each scheduling cycle.
- Since calculating all the future consequences in response to scheduler actions generally takes more time than available, and the schedule will be recalculated in the next time interval, the scheduler need only find a sequence of actions that leads in the direction of the goal state.

- The scheduler should make the most of the statistical information available in the form of heuristic rules to aid the process of pruning undesirable solutions.
- The goal can be defined as the lowest cost assignment in term of waiting time that can be found for all landing calls within a critical time period. We can also define an immediate goal, and that is, to produce for each lift the lowest cost assignment that can be found within the time period allowed.
- A schedule should take into account the overall cost resulting from the combined movements of the lifts, rather than the smallest local cost. For example, the path with smallest cost may favour new landing calls to old landing calls, resulting in long waiting times especially during peak periods That can be solved by introducing passenger waiting time as a part of the cost.

7.2 The heuristic search problem and its solution

Chapter 3 gave a detailed overview of the use of different intelligent techniques in lift scheduling, and in Chapter 4, heuristic search techniques were found to fall naturally in the process of searching for the near optimal real-time schedule

Search techniques suffer from two main drawbacks: namely the search time and memory requirement increases rapidly with the size of the problem. Typically, heuristics are applied to avoid exhaustive search by directing the search using the available knowledge, which in turn limits the possible future states. A* search uses heuristics to calculate the path with the least expensive cost to the goal state. The least expensive cost is defined in the current system as the minimum waiting time considering all landing calls The heuristics applied help to balance the different factors affecting the assignment of landing calls to achieve the minimum waiting time. This solves the problem of which series of actions is responsible for a particular outcome; this is the credit assignment problem [Rich and Knight 1991] For example, independently choosing the shortest path to one landing call can increase the time taken to answer the remaining landing calls

The algorithm limitations explained in Chapters 3 and 4 can be summarised as below. In the attempt to find the optimum solution, the time taken will be an exponential function of the number of landing calls To minimise the effects of this problem, a number of solutions have been suggested in the literature, one of which is to consider sub-optimal solutions as the complexity of the problem increases [Korf 1988, Korf 1990, Rich and Knight 1991]. Three of these approaches are discussed below.

1 Generating only the most promising paths and choosing the best path. For example, making full use of all the rules and constraints available in order to prune unpromising paths.

2. Improving the test procedure. This depends on the quality of the heuristic function used to evaluate generated nodes. Increasing the accuracy of the evaluation improves the search, but since the heuristic evaluator may have to investigate a large number of possibilities, it needs to be fast. The selection of a suitable heuristic function that gives an accurate indication of the most suitable path yet is computationally inexpensive is vital to satisfactory solution of many search problems.

3. Limiting the search horizon to ensure the search stops within a predetermined time. This may allow more time for the application of a heuristic function which provides a better quality estimate.

By providing high data quality such as load information and predicted data based on historical and real-time information, this work investigates whether a full search can be achieved with low cost, especially under normal traffic situations. The use of a limited search horizon may still be appropriate when the number of landing calls is relatively large and the search time is greater than the calculation cycle. Under such circumstances the goal can change to one of simply providing all lifts with their first assignment. Once the initial assignment for each lift has been calculated the search can be further refined during the next calculation cycle

7.3 A simple search tree example

This section describes the skeleton of a suitable search tree technique for the lift system scheduler There are two important points to be noted Firstly, the order in which the landing calls are assigned is important. The tree structure must be able to find the sequence of assignments for all landing calls which gives the smallest cost. Secondly, the immediate goal is to give the lift system its first assignment. Hence it is important to be able to find a tree structure that provides each lift with a landing call assignment without the need to search exhaustively.

Figure 7 1 shows an example of a building with two lifts moving to answer car calls. Both lifts can decelerate to stop at the 7th floor. For the purpose of the current explanation the following simplified assumptions apply:

- answering a landing call takes 4 units of time;
- answering a car call takes 4 units of time,
- the transient time between adjacent floors is 1 unit of time;
- after answering a landing call a car call is generated and the car call floor is the last floor (top or bottom) in the direction of the landing call;
- all car calls must be answered before reversing direction.

A search tree generally consists of several levels, where each level contains a number of nodes (states) equal to the number of landing calls remaining to be answered. At each level, the node representing a lift answering a landing call with a minimum cost f is assigned and then expanded, where $f=g+h$. The time to answer all landing calls from the start of the lift trip until the present call is g and h is the cost of answering the remaining calls. h is calculated by choosing the lowest cost for answering the remaining calls by all available lifts. For N landing calls, if i is the number of an assigned landing call whose cost is g, then:

$$h = \sum_{i}^{N} \min(\mathrm{cost}_{N-i}) \qquad (7.1)$$

where $\min(\mathrm{cost}_{N-i})$ is the minimum cost of answering a landing call from all remaining landing calls by all lifts. The cost of answering a landing call is the time required by the lift to reach that landing call. This method is based on the commonly used *simplex collective* technique (Chapter 2) in which a lift travels around its zone answering the first landing call in its way. The principle of the search is that rather than generating nodes for all lifts for all landing calls, the search is directed toward the most promising paths using the least cost path for a lift to a landing call (nearest neighbour heuristic)

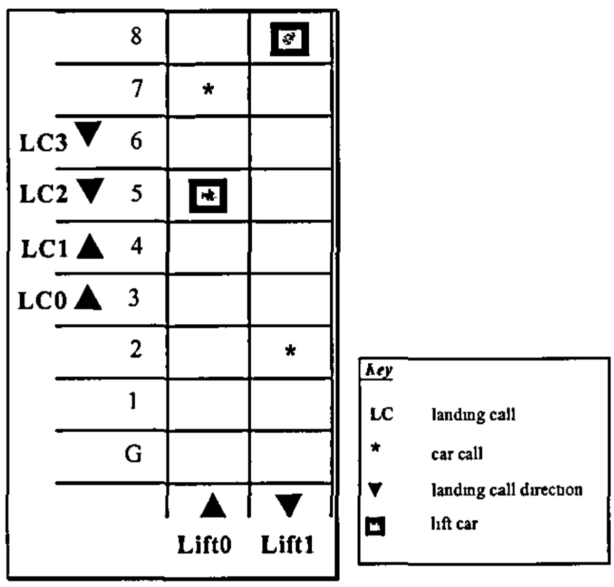


Figure 7.1 An example of the configuration of lifts and landing calls. (LC denotes landing calls).

Figure 7.2 shows an example search tree for the configuration of Figure 7.1. At each level, one node is generated for each lift. The landing call assigned to a lift is chosen according to the lowest cost of reaching a landing call, *g*. Table 7.1 gives the cost of the direct paths of each lift to each landing call from the initial state following the given rules. The first column shows landing calls which are numbered first in order of up landing calls and then in order of down landing calls.

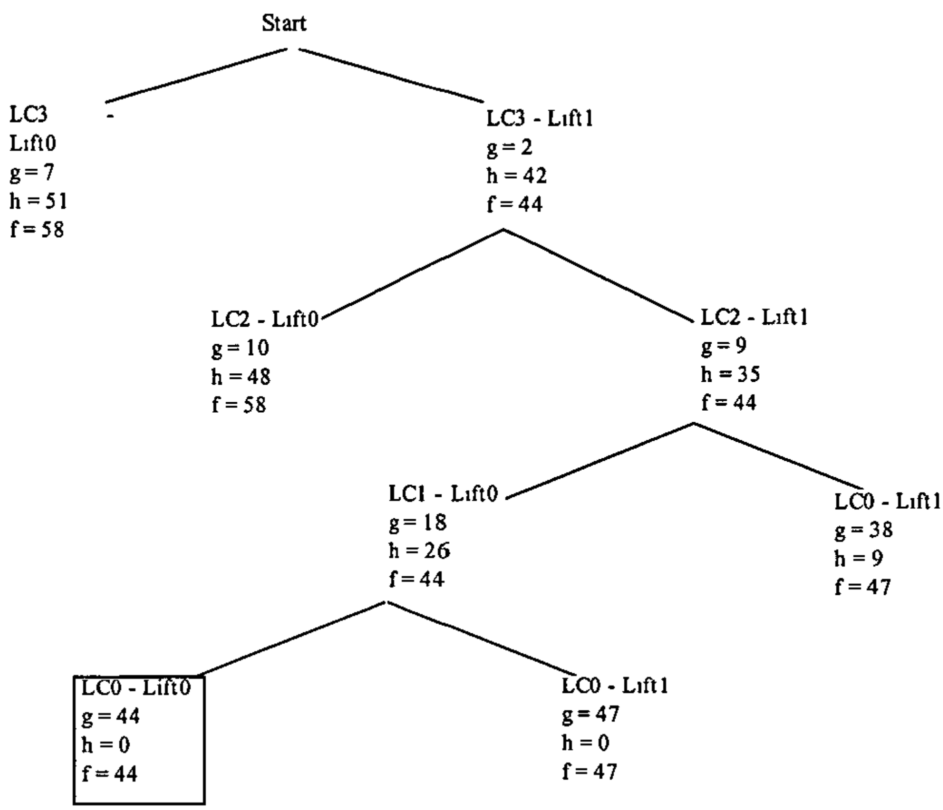


Figure 7.2 A* search tree for the example of Figure 7.1

Landing calls	Lift 0	Lift 1
LC0	10	11
LC1	9	12
LC2	8	3
LC3	7	2

Table 7.1 Costs of lift direct trips to landing calls for the example of Figure 7.1

Table 7 2 and Table 7 3 show the cost of a lift answering a landing call having already answered a previous call. For example in Table 7 2, the cost of *Lift 0* answering *LC2* after *LC1* is 15s. In the tables, *LCp* indicates landing call number *p*.

Second landing call	First landing call LC0	LC1	LC2	LC3
LC0	0	17	16	17
LC1	5	0	17	18
LC2	16	15	0	5
LC3	15	14	19	0

Table 7.2 Costs of trips between landing calls for lift 0 for the example of Figure 7.1

Second landing call	First landing call LC0	LC1	LC2	LC3
LC0	0	17	20	21
LC1	5	0	21	22
LC2	16	15	0	5
LC3	15	14	23	0

Table 7.3 Costs of trips between landing calls for lift 1 for the example of Figure 7.1

Although the nearest landing call to a parked lift or the first landing call in the way of a moving lift is the lowest cost path for that lift, this approach will not necessarily produce the best combination of paths for the lift system as a whole. The order in which landing calls are answered affect the overall cost, and this can result in an overestimation of the f value and hence an optimum solution may not be found, even for the simplest example.

In Figure 7.2, the search has *Lift 0* answering *LC1* before *LC0,* while if *Lift 0* answers *LC0* before *LC1,* a final goal node would have a value of f*=34* (the calculation example for the value of h is given in Section 7 5). An alternative approach is to modify the search to consider a greater number of alternatives at each level in the tree. Figure 7 3 shows another example of the same building but with a different landing call distribution Table 7.4, Table 7.5 and 7 6 show the cost of the trips to a landing call cost tables as explained in the previous example.

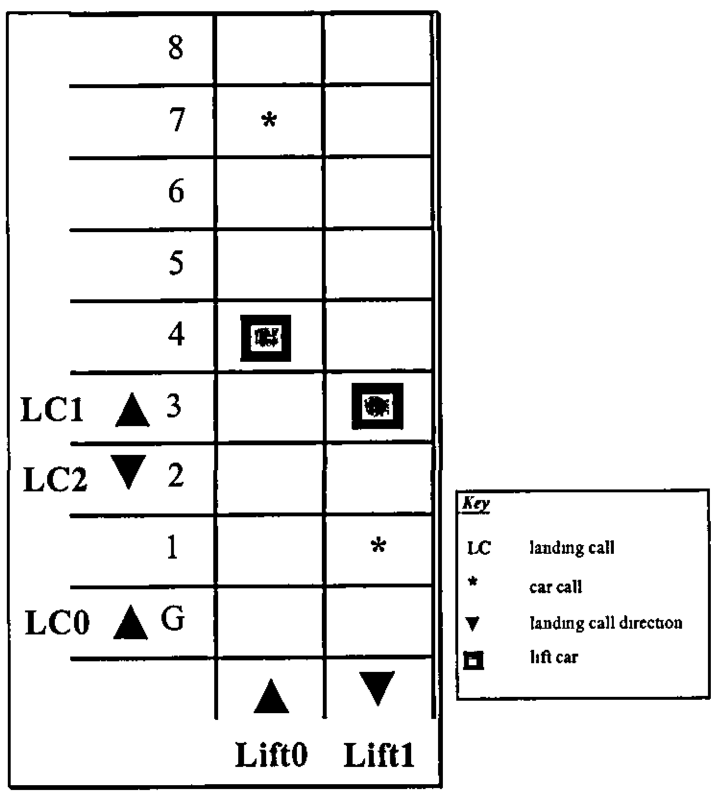


Figure 7.3 A second example of the configuration of lifts and landing calls.

Figure 7.4 shows the search tree for the example of Figure 7.3, where all unassigned landing calls are considered at each stage of the search tree. At each stage, a landing call is answered by the lift which gives the lowest cost path. The lift assignment follows the same order as the tree levels, that is, the first level is the first lift assignment. The search

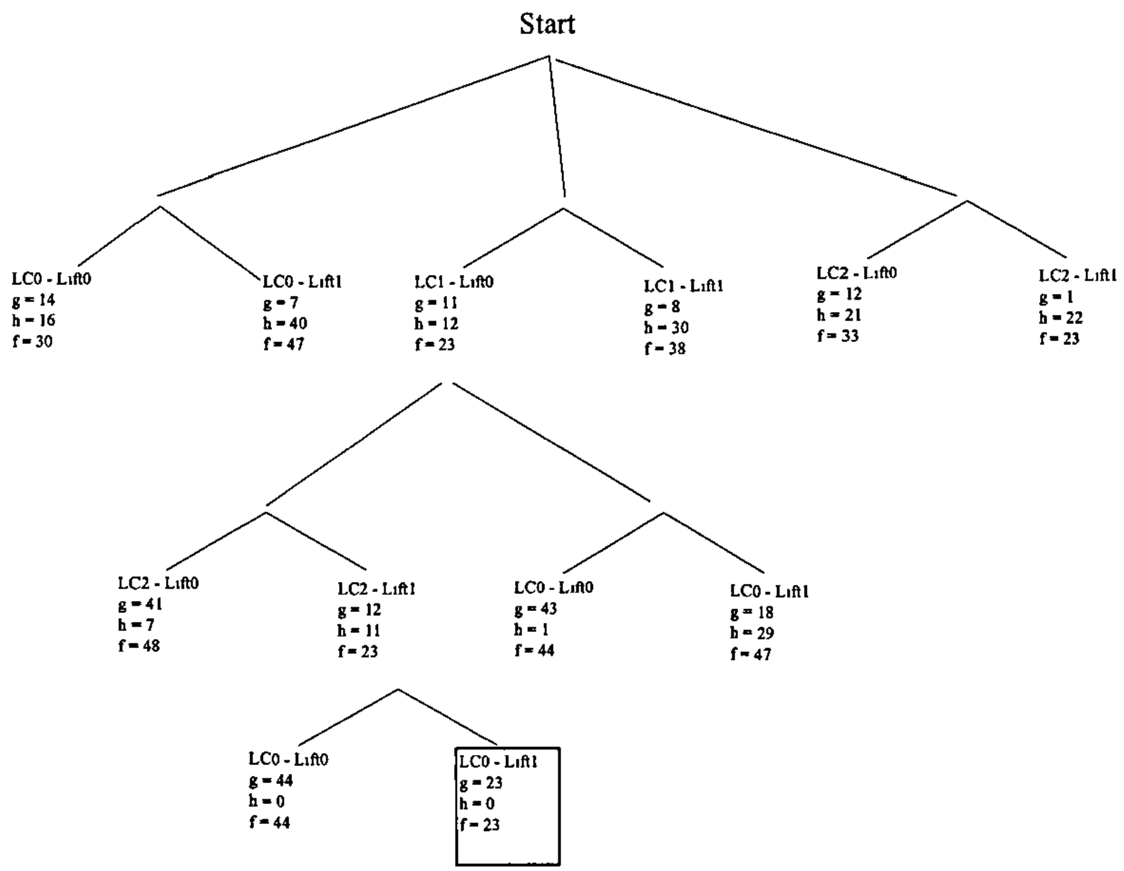


Figure 7.4 A* search tree for the example of Figure 7.3.

has given the smallest value of f and, unlike the first example, the path to answer *LC0* with *lift 1* is of lower cost than that of *lift 0*.

Landing calls	Lift 0	Lift 1
LC0	14	7
LC1	11	8
LC2	12	1

Table 7.4 Costs of lift direct trips to landing calls for the example of Figure 7.3.

Second	First landing call		
landing call	LC0	LC1	LC2
LC0	0	21	6
LC1	7	0	13
LC2	22	19	0

Table 7.5 Costs of trips between landing calls for lift 0 for the example of Figure 7.3.

Second	First landing call		
landing call	LC0	LC1	LC2
LC0	0	21	10
LC1	7	0	17
LC2	22	19	0

Table 7.6 Costs of trips between landing calls for lift 1 for the example of Figure 7.3.

The nearest neighbour heuristic is still implemented, but a larger number of nodes is considered at each level than in the previous search method, Figure 7 2. The aim at this point is to minimise the number of generated nodes As the number of landing calls increases it may take more time to calculate the value of f since the number of nodes generated will increase. When the number of landing calls exceeds the number of lifts, the immediate goal of the scheduler is to find the first lowest cost assignment for each lift. Figure 7.5 shows a search tree prioritised according to lift usage rather than landing call assignment. The full assignment is presented in Table 7.7. Each level of the tree represents a possible lift assignment to the remaining unassigned landing calls and the possibility of no assignment to that lift (that is a free node). The total number of nodes for

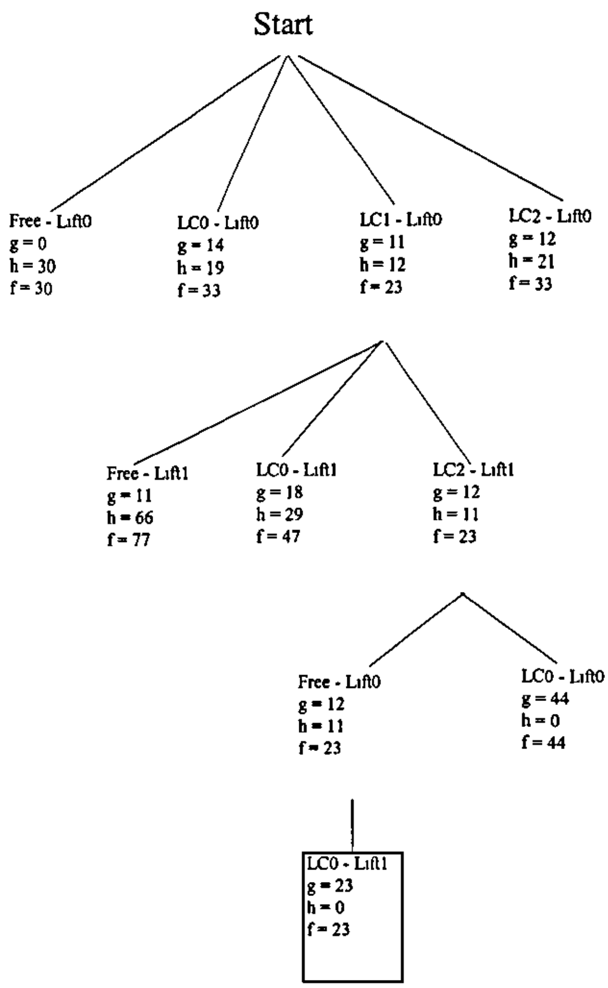


Figure 7.5 Prioritised A* search tree for the example of Figure 7.3

each level is equal to the sum of the number of landing calls left to be assigned and the free node. If a free node is chosen, the lift remains free of assignment until the end of search, as long as the search remains within that branch of the tree. If only one lift is available for assignment then no free node is generated for that lift, for example only one node is generated for *lift 1* at the last level of the tree in Figure 7.5. Note that this search tree generates the same assignment as that of Figure 7.4, but with only 10 nodes in total, while the previous search generated 12 nodes

Figure 7.6 shows the prioritised search tree of Figure 7 1, with the assignment shown in Table 7 8. The search has given the correct value of *f* with the most suitable order of assignment.

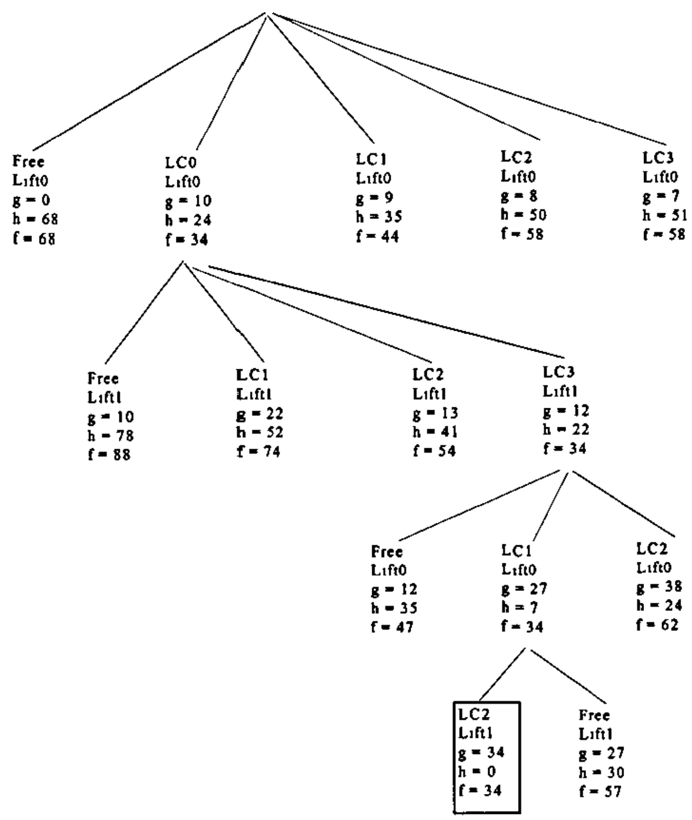


Figure 7.6 Prioritised A* search tree for the example of Figure 7.2

As the tree structure forces the assignment of lifts rather than landing calls, it is possible to stop the search when the number of tree levels is equal to the number of lifts. This technique of determining which resources should be given greatest importance in the application of the search has been termed *Prioritised A* search* by the author.

	Assignment	Lift trip total cost up to this assignment
Lift 0	LC1	11
Lift 1	LC2	1
Lift 1	LC0	11

Table 7.7 Prioritised A* scheduler assignments for the example of Figure 7.3.

	Assignment	Lift trip total cost up to this assignment
Lift 0	LC0	10
Lift 1	LC3	2
Lift 0	LC1	15
Lift 1	LC2	7

Table 7.8 Prioritised A* scheduler assignments for the example of Figure 7.1.

The search method appears to be satisfactory, but what would happen if the time interval dedicated for the scheduling cycle expires before the search is complete? What if the time expires while backtracking up the tree? In such a case the lifts' first assignments may not be valid. If this assignment is available then it might not be the optimum solution, as if the search continued, a better solution might be found. The scheduler would then have to correct its action, and this might cause instability in the lift system As explained in Section 7 2, improving the quality of the calculation of the h value may lead the search into the most suitable branch, while avoiding backtracking If this is true then truncating the search tree after first assignment would provide the required schedule

7.4 Practical assumptions for the intelligent scheduler

This section explains the rules and assumptions used in order to calculate the value of *f*. The assumptions are based on those rules generally followed by lift companies, as given in detail in Appendix D. The aim of such rules is to supply the scheduler with good heuristic functions which minimise the duration of the search. Moreover, if a schedule has to be given before a full search tree is conducted, the assignment is enhanced by improving the quality of the heuristic calculation.

The examples given in Section 7.3 show tables of lift trip times to answer landing calls The tables show values calculated using the simplified delay assumptions introduced for illustration purposes. In the following, the assumptions are replaced by information supplied by Express Evans Ltd (Figure 6.3), as follows

- One second is allocated for travelling between floors;
- 8.9s is allocated if the lift is stopping for a landing call, (one passenger entry 0.5s + 3 5s after passenger entry + doors opening and closing 4.9s);
- 6.4s is allocated if the lift is stopping for a car call, (one passenger departure 0.5s + 1s after passenger departure + doors opening and closing 4.9s);
- 9.4s is allocated if lift is stopping for both a landing call and a car call;
- 2.7s is allocated for door closing delay,
- A lift must answer all car calls before reversing direction
- When a lift is assumed to be answering a landing call, all other paths are calculated on the assumption that the lift would have a delay of 8.9s at the landing call floor. A passenger entering the lift would normally register a car call which is assumed to be at the furthest floor in the direction of the landing call. The assumed car call delay is added to all landing call paths not in the way of the lift, except for any landing call at the last floor in the direction of travel. If no passengers enter the lift, then no car calls are issued and the simulator should wait for a dwell time of 8s before accepting any new assignment. At the end of the dwell period, the lift starts closing its doors and

next direction changes to STOP. At this point, the lift is no longer assumed to have a car call and the path cost will only consider door closing time delay.

- A car call delay of 6.4s at the destination landing call floor is not included in the calculation of the cost of answering that landing call.
- The path cost of a lift moving to a landing call also depends on the number of car calls between the lift initial position and the destination landing call. Lift trip time to and from each car call is calculated rather than just using the direct trip time to the landing call floor.
- When a lift is not moving and a landing call is issued at the lift position while the door is closing, the path cost is assumed to be zero.
- If a lift is not moving then it has parked, stopped for a landing call or car call or both a landing call and car call.
- When a lift has stopped for its last car call or parked, the lift may have its next direction as STOP, PARK, UP or DOWN. The door delay may vary between zero when door is closed to the normal full delay if the lift has just stopped (no more than 6.4s).
- When the lift is moving with no car call and cannot stop for a landing call, then the trip time to that landing call is calculated as the time it takes the lift to travel to the next possible stop and reverse back to the landing call floor.
- When a lift is decelerating, the advanced lift position (passed from the monitoring system) is equal to lift destination. The time remaining to stop is added to the cost of answering landing calls. The lift next direction is changed by the simulator so that if the lift is stopping for its last car call then its next direction changes to STOP. If the lift has more car calls then the next direction remains unchanged. No assumptions for opposite direction car calls are made If the lift is stopping for a landing call then its next direction would be the landing call direction.
- The direct path array (Table 7.1 and Table 7.4) gives the cost of answering a landing call from the initial lift state.

- The path of a lift answering a landing call after its assumed assignment is calculated using the between landing call array (Table 7.2, Table 7.3, Table 7.5 and Table 7 6). If there is a car call at the level, that is, at the start of the trip path when a lift is answering the landing call, then a passenger exit time of 0 5s is added to the total cost of answering any further landing calls. When calculating the paths from any landing call not in the way of a lift, it is assumed that the lift has answered all its car calls before reaching these landing calls.

These assumptions are used in testing the prioritised A* scheduler with a number of different examples for the first stage of implementation. The second stage of implementation tests the performance of A* scheduler using the simulator as the quality of data supplied by the simulator is gradually improved.

7.5 Prioritised A* algorithm

In the prioritised A* algorithm, the tree consists of a number of levels, each of which is

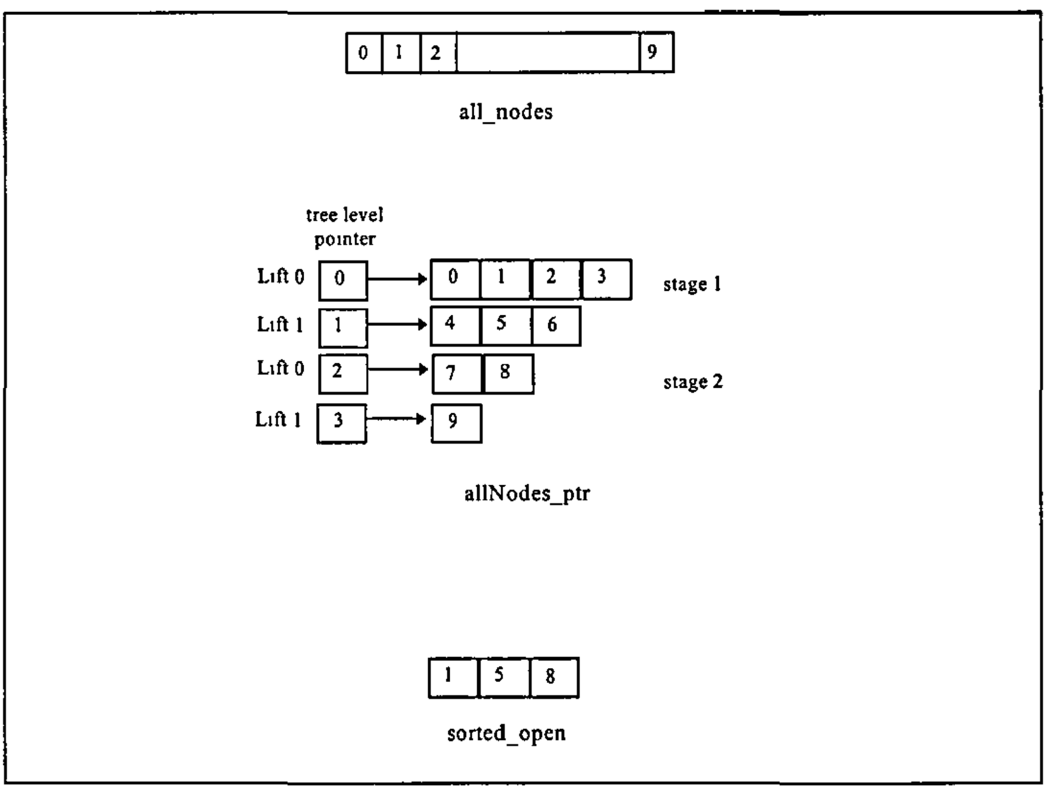


Figure 7.7 Example of arrays generated for 2 lifts and 4 landing call tree.

assigned to one lift, Figure 7.7. The number of levels depends on the number of landing calls and lift assignments. At each level of the tree, a number of nodes equal to the number of unassigned landing calls is created plus a free node. A free node is generated only if there is more than one lift available for assignment. The nodes are created and saved in an array of structures which is indexed according to a node number. Each node contains a node number, a parent node number, the tree level, the landing call assigned, *g*, *h*, and *f* values, and the state of the node (open when first generated and closed when the node is expanded and its children are generated). The nodes for each level of the tree are sorted according to their *f* value. If the *f* values are equal, the nodes are sorted according to the *h* value The sorted nodes at each level are accessed using an array which is indexed according to each level of the tree (allNodes_ptr[tree_level][node]). At each tree level, the node that follows the last closed node in the allNodes_ptr array is saved in a separate array This array points to the lowest cost opened nodes at each expanded tree level (sorted_open_array). Each time the sorted_open_array is updated, the array is sorted so that the node with the lowest *f* value is at the first location of the array. The allNodes_ptr array only saves the nodes in the current path and each level is sorted when nodes are generated. This makes the selection of the next lowest cost nodes easier and the sorting time is reduced as there is no need to sort all open nodes each time a level is generated.

The node with the lowest cost at a level is first compared with the lowest cost node in sorted_open_array. If the first node has a smaller *f* value, then it is chosen for expansion and another level is added to the tree Otherwise, the second node is the lowest cost node which is then is deleted from the sorted_open_array and the first node is added in its place. At this stage the search will backtrack either up or down the tree depending on the selected node level and the search tree is updated accordingly. Once the tree is updated, the next lowest node at the new tree level is added to sorted_open_array and the array is then sorted according to the node's *f* values

There are two types of backtracking.

1. Backtracking occurs when the parent of the chosen node is in the current tree path to a tree level which is lower down the tree or at the same level of the current tree branch.

2. If the parent node of the chosen node is not in the current tree branch, then backtracking would start from the chosen node via the root node.

The first backtracking method is useful when there is no need to backtrack to the root of the tree, since only the necessary levels are updated.

As explained in Section 3 1.1, it is important that *h* is not overestimated in order to guarantee finding the lowest cost path to the solution. Russell and Norvig [1995] explain that for most real problems, the only way to guarantee that *h* is never overestimated is by setting it to zero The search then would be admissible but not efficient. If *h* is rarely overestimated by more than δ, then the A* algorithm will rarely find a solution whose cost is more than δ greater than the cost of the optimal solution.

Since *h* resembles the distance of a node to the goal, then if it were a perfect estimation A* will converge immediately to the goal node with no search. The better the estimation of *h* the closer the solution becomes to the direct approach.

During heavy traffic periods when the number of landing calls exceeds the number of lifts and the scheduling time increases with the size of the problem, then providing lifts with their initial assignment from the bottom of the tree (the lifts' first assignments) becomes necessary if the scheduling time exceeds a predefined time interval. Hence, it is important that the A* scheduler coverages immediately to a schedule to avoid long search times, meaning that the better the estimation of *h* the more likely that the schedule produced is near the optimum solution. The algorithm used in the calculation of the values of *g* and *h* is based on both the direct path array and between landing call array. The *g* value is equal to the cost of answering all assigned landing calls, but for a free node, no cost is added to the value of *g*. The *h* value is equal to the lowest cost of answering the remaining landing calls using the available lifts, Section 7 3 and Equation 7 1. Figure 7.8 shows an example of the calculation of the value of *h* for the node in the bottom level of the A* tree chosen for expansion in Figure 7.6. The node is the first assignment for *lift 0*, calculated using the path cost tables Table 7.1, Table 7.2 and Table 7.3. The symbols (such **Lift 0: (LC0, 10) + (LC1, 5) =15** denotes that *lift 0* is answering *LC0* with a cost of *10* then answering *LC1* with a cost of *5* which results in a total cost of *15*.

Lift 0 is assigned to LC0 with g = 10

h = minimum cost of answering all remaining landing calls using the available lifts.

Search for the next landing call with smallest cost for each lift

Assign the lift with minimum cost

Lift 0: (LC0, 10) + (LC1, 5) = 15

Lift 1: (LC3, 2) = 2

Then h = 2 *LC0 and LC3 are assigned*

Lift 0: (LC0, 10) + (LC1, 5) = 15

Lift 1: (LC3, 2) + (LC2, 5) = 7

Then h = 2 + 7 = 9 *LC0, LC2 and LC3 are assigned*

Lift 0: (LC0, 10) + (LC1, 5) = 15

Lift 1: (LC3, 2) + (LC2, 5) + (LC1+21) = 28

Then h = 9 + 15 = 24 *all landing calls are assigned*

f = 10 + 24 = 34

Figure 7.8 The calculation of the node selected for assignment at the first level of the tree using the example in Figure 7.6.

7.6 Prioritised A* lift scheduler results

The prioritised A* algorithm described in the previous section is tested using several different examples. Other ways of calculating the h value were used and compared with the above method The scheduler was tested for the same examples when $h = 0$ and when h is calculated using only the direct path cost to landing calls

The results show that the tree converges directly to the goal without any backtracking when using the calculation method introduced in Figure 7.8. The chosen node f value remains the same at all levels until the goal node is reached. The results were compared with those produced when $h=0$, and can be found in Appendix E. Table 7.9 and Table 7.10 show the results for the examples in Figure 7.1 and Figure 7.3 respectively. The first column denotes h when calculated using both the direct and the between landing calls arrays; the second column shows the results of the A* search results using $h=0$ and the third column gives the results when the algorithm is tested using only the direct path to the landing call when calculating the cost. In this last approach, the lifts' previous assignments are not considered and only the direct path from the current lift position to the landing call is used. In all cases, the schedule and its cost f are the same

	$h=\sum_{i}^{N}\min(\text{cost}_{N-i})$	$h=0$	$h_d=\sum_{i}^{N}\min(\text{DirectCost}_{N-i})$
Scheduling time	0 8ms	14ms	7 7ms
Number of nodes	14	163	107
Number of backtracks	0	47	32

Table 7.9 The results of testing the prioritised A* scheduler for the example in Figure 7.1 using three different methods of calculating h.

To compare the value of h calculated using the direct path method with the calculations of h in the previous section, the value of h for the same node in Figure 7 8 is recalculated in Figure 7 9, using the direct path, Table 7 1

	$h=\sum_{i}^{N}\min(\text{cost}_{N-i})$	$h=0$	$h_d=\sum_{i}^{N}\min(\text{DirectCost}_{N-i})$
Scheduling time	0 6ms	2ms	1 7ms
Number of nodes	10	49	24
Number of backtracks	0	15	7

Table 7.10 The results of testing the prioritised A* scheduler for the example in Figure 7.3 using three different methods of calculating h.

Lift 0 is assigned to LC0 with g = 10

h = minimum cost of answering all remaining landing calls using the available lifts.

Search for the next landing call with lowest cost for each lift
Assign the lift with minimum cost

Lift 0: LC0 -> LC3, (7)

Lift 1: LC3, (2)

Then h = 2 *LC0 and LC3 are assigned*

Lift 0: LC0 -> LC2, (8)

Lift 1: LC3 -> LC2, (3)

Then h = 5 *LC0, LC2 and LC3 are assigned*

Lift 0: LC0 -> LC1, (9)

Lift 1: LC3 -> LC2->LC1, (12)

Then h = 14 *All landing calls are assigned*

f = 10 + 14 = 24

Figure 7.9 The calculation of the node selected for assignment at the first level of the tree using only direct path cost in Table 7.1 for the example in Figure 7.8.

The above results demonstrate the following

- The greater the accuracy of the *h* value the faster is the calculation of the schedule If *h* is calculated without error the schedule will converge to the goal state without backtracking, and so the search could be interrupted once an initial assignment for each lift is available, that is, when the number of levels is equal to the number of lifts.
- Using an estimated value of *h* less than the actual value will provide an optimum solution, but its calculation takes longer due to backtracking. For examples with a larger search space the algorithm may use prohibitive quantity of memory, as illustrated by example 10 in Appendix F. As the complexity of the problem increases and the time taken to calculate the schedule increases, then if the search is stopped before completion, an assignment for each lift may not be available, or the solution may not be optimal

- All three methods provide the same schedule with the same cost demonstrating that the prioritised A* search provides the optimum solution in this case.

A detailed example is now given which describes the methods of calculation adopted in the implementation of the prioritised A* search. Appendix F includes a number of selected examples for a building with 18 floors. These examples are now calculated using the assumptions defined in Section 7.3. In some examples the value of *f* reduces while searching the tree, yet the search still converges without backtracking. The reason why this occurs can be explained by considering the example shown in Figure 7.10. In this example, the lifts are stationary, lift 0 has stopped to answer a landing call and lift 1 has stopped to answer a car call.

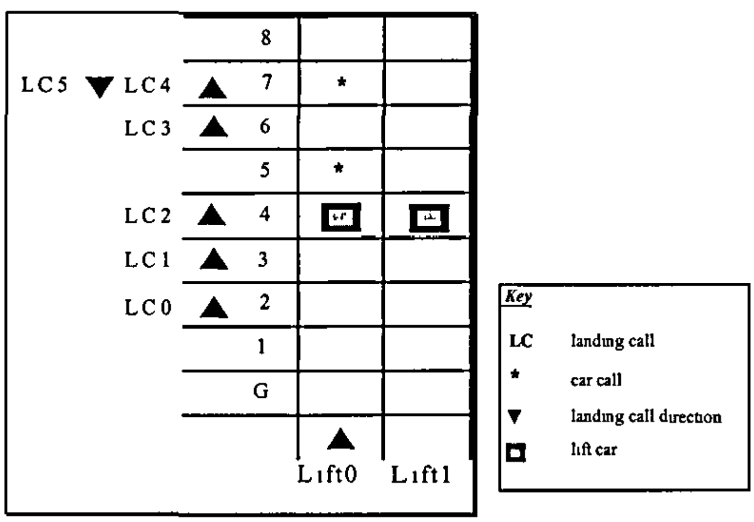


Figure 7.10 A third example of the configuration of lifts and landing calls.

The relevant values in this example can be found in Table 7.11, Table 7.12 and Table 7.13. At the first level in the search tree a node with value of *f=112 6* is expanded, Figure 7.11. While in the second level the node with value of *f=99.7* is chosen, Figure 7.12. When *h* is calculated, it follows the shortest path to a landing call. *Lift1* was first assumed to answer *LC1*. At the second level, *Lift1* answers *LC0* first and the search corrects its *f* value for the remainder of the search. The structure of the tree allows the correction of the node value without the need for backtracking. The resulting schedule is shown in Table 7.12.

Landing calls	Lift 0	Lift 1
LC0	29 7	4 7
LC1	28 7	3 7
LC2	0	0
LC3	17 3	4 7
LC4	18 3	5 7
LC5	18 3	5 7

Table 7.11 Costs in seconds of lift direct trips to landing calls for the example of Figure 7.10.

Second landing call	First landing call					
	LC0	LC1	LC2	LC3	LC4	LC5
LC0	0	26 3	38 1	29 7	22 8	24 8
LC1	9 9	0	37 1	28 7	21 8	25 8
LC2	10 9	9 9	0	27.7	20 8	26 8
LC3	12 9	11 9	17 3	0	18 8	28 8
LC4	13 9	12 9	18 3	9 9	0	29 8
LC5	22 3	21 3	33 1	24 7	17 8	0

Table 7.12. Costs in seconds of trips between landing calls for lift 0 for the example of Figure 7.10.

Second landing call	First landing call					
	LC0	LC1	LC2	LC3	LC4	LC5
LC0	0	26 3	25 8	23 3	22 3	24 3
LC1	9 9	0	24 8	22 3	21 3	25 3
LC2	10 9	9 9	0	21 3	20 3	26 3
LC3	12 9	11 9	11 4	0	18 3	28 3
LC4	13 9	12 9	12 4	9 9	0	29 3
LC5	22 3	21 3	20 8	18 3	17 3	0

Table 7.13. Costs in seconds of trips between landing calls for lift 1 for the example of Figure 7.10.

Lift 0 is assigned to LC2 with g = 0

h = minimum cost of answering all remaining landing calls using the available lifts

Search for the next landing call with smallest cost for each lift

Assign the lift with minimum cost

Lift 0: (LC2, 0) + (LC3, 17.3) = 17.3
Lift 1: (LC1, 3.7) = 3.7
Then h = 3.7 *LC1 and LC2 are assigned*

Lift 0: (LC2, 0) + (LC3, 17.3) = 17.3
Lift 1: (LC1, 3.7) + (LC3,11.9) = 15.6
Then h = 3.7+15.6 = 19.3 *LC1, LC2 and LC3 are assigned*

Lift 0: (LC2, 0) + (LC4, 18.3) = 18.3
Lift 1: (LC1, 3.7) + (LC3,11.9) + (LC4, 9.9) = 25.5
Then h = 19.3 + 18.3 = 37.6 *LC1, LC2, LC3 and LC4 are assigned*

Lift 0: (LC2, 0) + (LC4, 18.3) + (LC5, 17.8) = 36.1
Lift 1: (LC1, 3.7) + (LC3, 11.9) + (LC5, 18.3) = 33.9
Then h = 37.6 + 33.9 = 71.5 *LC1, LC2, LC3, LC4 and LC5 are assigned*

Lift 0: (LC2, 0) + (LC4, 18.3) + (LC0, 22.8) = 41.1
Lift 1: (LC1, 3.7) + (LC3, 11.9) + (LC5, 18.3) + (LC0, 24.3) = 58.2
Then h = 71.5 + 41.1 = 112.6 *All landing calls are assigned*

f = 112.6

Figure 7.11 The calculations carried out to determine *f* at the first node chosen for expansion in the search tree for the example of Figure 7.10.

Lift 0 is assigned to LC2 and Lift1 is assigned to LC0 with g = 4.7

h = minimum cost of answering all remaining landing calls using the available lifts

Search for the next landing call with smallest cost for each lift

Assign the lift with minimum cost

Lift 0: (LC2, 0) + (LC3, 17.3) = 17.3

Lift 1: (LC0, 4.7) + (LC1, 9.9) = 14.6

Then h = 14.6 *LC0, LC1 and LC2 are assigned*

Lift 0: (LC2, 0) + (LC3, 17.3) = 17.3

Lift 1: (LC0, 4.7) + (LC1, 9.9) + (LC3, 11.9) = 26.5

Then h = 14.6 + 17.3 = 31.9 *LC0, LC1, LC2 and LC3 are assigned*

Lift 0: (LC2, 0) + (LC3, 17.3) + (LC4, 9.9) = 27.2

Lift 1: (LC0, 4.7) + (LC1, 9.9) + (LC4, 12.9) = 27.5

Then h = 31.9 + 27.2 = 59.1 *LC0, LC1, LC2, LC3 and LC4 are assigned*

Lift 0: (LC2, 0) + (LC3, 17.3) + (LC4, 9.9) + (LC5, 17.8) = 45

Lift 1: (LC0, 4.7) + (LC1, 9.9) + (LC5, 21.3) = 35.9

Then h = 59.1 + 35.9 = 95 *all landing calls are assigned*

f = 4.7+95 = 99.7

Figure 7.12 The calculations carried out to determine *f* at the second node chosen for expansion in the search tree for the example of Figure 7.10.

	Assignment	Lift trip total cost up to this assignment in seconds
Lift0	LC2	0
Lift1	LC0	4 7
Lift0	LC3	17 3
Lift1	LC1	14 6
Lift0	LC4	27 2
Lift1	LC5	35 9

Table 7.14 The prioritised A* scheduler results for the example of Figure 7.10.

To calculate the schedule, the lift system needs no more than the lifts' first assignments each time the scheduler is called Since the assignment is the same whether the search is truncated or continued until completion, the search method may be useful for other scheduling problems where more than one assignment is required for each resource. For other real-time applications the search can be limited to some critical time period.

7.7 Passenger waiting time effect on the prioritised A* scheduler

The aim of the intelligent lift scheduler is to minimise passenger waiting time However, ETQ and the prioritised A* search described in the previous sections use the time it takes a lift to reach a landing call. Consequently, in order to produce a schedule which aims to minimise waiting time, it is important to restate the problem to introduce how long a passenger has been waiting. The only indication of how long a passenger has been waiting is the time elapsed since the landing call was issued, so that the passenger waiting time is equal to the landing call waiting time. For the example in Figure 7 1, the waiting times for the passengers at each landing are shown in Table 7.15. Table 7.16 and Table 7 17 show the lift path calculations in terms of waiting time.

A comparison of the scheduling results (Table 7.19) with those obtained without using landing call waiting time (Table 7 8) shows some differences. In particular, the first assignment of lift 0 as calculated in Figure 7 13 is different. Figure 7 14 shows the recalculations of the same node as in Figure 7.8 but now using waiting time The detailed results can be found in Appendix E

Landing calls	Waiting time
LC0	0 5s
LC1	0 1s
LC2	30s
LC3	20s

Table 7.15 Waiting times for the landing calls in the example in Figure 7.1.

	Lift 0	Lift 1
LC0	10 5	11 5
LC1	9 1	12 1
LC2	38	33
LC3	27	22

Table 7.16 Costs in seconds of lift direct trips to landing calls including passenger waiting time for the example in Figure 7.1.

Second	First landing call			
landing call	LC0	LC1	LC2	LC3
LC0	0	17 5	16 5	17 5
LC1	5 1	0	17 1	18 1
LC2	46	45	0	35
LC3	35	34	39	0

Table 7.17 Costs in seconds of trips between landing calls for lift 0 including passenger waiting time for the example in Figure 7.1.

Second landing call	First landing call			
	LC0	LC1	LC2	LC3
LC0	0	17 5	20 5	21 5
LC1	5 1	0	21 1	22 1
LC2	46	45	0	35
LC3	35	34	43	0

Table 7.18 Costs in seconds of trips between landing calls for lift 1 including passenger waiting time for the example in Figure 7.1.

	Assignment	Lift trip total cost up to this assignment in seconds
Lift 0	LC3	7
Lift 1	LC0	11
Lift 0	LC2	12
Lift 1	LC1	16

Table 7.19 The prioritised A* scheduler results including passenger waiting time for the example in Figure 7.1.

Lift 0 is assigned to LC3 with g = 27

h = minimum cost of answering all remaining landing calls using the available lifts

Search for the next landing call with lowest cost for each lift
Assign the lift with minimum cost

Lift 0: (LC3, 7) + (LC0, 17.5) = 24.5

Lift 1: (LC0, 11.5) = 11.5

Then h = 11.5 *LC0 and LC1 are assigned*

Lift 0: (LC3, 7) +(LC1, 18.1) =25.1

Lift 1: (LC0,11) + (LC1, 5.1) = 16.1

Then h = 11.5 + 16.1= 27.6 *LC0, LC1 and LC3 are assigned*

Lift 0: (LC3,7) + (LC2, 35) =42

Lift 1: (LC0,11) + (LC1, 5) + (LC2, 45) = 61

Then h = 27.6 + 42 = 69.6 *all landing calls are assigned*

f = 27 + 69.6 = 96.6

Figure 7.13 Recalculation of the node in Figure 7.6 for lift 0 answering LC3 at the first level of the tree.

Lift 0 is assigned to LC0 with g = 10.5

h = minimum cost of answering all remaining landing calls using the available lifts:

Search for the next landing call with lowest cost for each lift
Assign the lift with minimum cost

Lift0: (LC0, 10) + (LC1, 5.1) = 15.1

Lift1: (LC1, 12.1) = 12.1

Then h = 12.1 *LC0 and LC1 are assigned*

Lift0: (LC0, 10) + (LC3, 35) = 45

Lift1: (LC1, 12) + (LC3, 34) = 46

Then h = 12.1 + 45= 57.1 *LC0, LC1 and LC3 are assigned*

Lift0: (LC0, 10) + (LC3, 15) + (LC2, 35) = 60

Lift1: (LC1, 12) + (LC2, 45) = 57

Then h = 57.1 + 57 = 114.1 *all landing calls are assigned*

f = 10.5 + 114.1 = 124.6

Figure 7.14 Recalculation of the node in Figure 7.6 for lift 0 answering LC0 at the first level of the tree

7.8 Conclusions

A real-time intelligent scheduler has been introduced and developed in this chapter. The scheduler uses a prioritised A* search with the following characteristics

- The prioritised A* lift scheduler is operating in an environment which is continuously changing, and as shown in Chapter 8, is able to take advantage of all the available information about the correct state of the lift system, for example landing call waiting time, door delay, passenger exit and entry delay and the exact lift journey time.
- The scheduler gives priority to the assignment of lifts to landing calls, ensuring that each lift is given an assignment.

- The prioritised A* search converges to the goal state without backtracking. This means that the heuristics are providing a good estimate of the lift path cost. The result also shows that the order in which landing calls are answered is important Since the A* search gives the opportunity to investigate all the possible orderings of landing calls, the search, once terminated, would find the lowest cost schedule.
- Since the scheduler does not suffer from backtracking, it is possible to truncate the search at any stage and still produce a correct partial assignment. In the lift system, only the first assignment of lifts is required from the scheduler and so the search tree could be truncated when the number of tree levels is equal to the number of available lifts. Generally, schedulers cannot be interrupted and still be expected to give the correct assignments For the prioritised A* scheduler, it is possible to determine for a given number of lifts and landing calls how long the schedule will take to calculate.
- In the lift scheduler the main aim is to minimise the average waiting time and this is in contrast to the ETQ scheduler. Moreover, the schedule having the lowest cost for answering all landing calls is chosen rather than arbitrarily assigning priorities to some landing calls over others.

In the following chapter, investigates the practical implementation of the lift monitoring system which is responsible for providing prediction and further control actions such as up peak control. The prioritised A* scheduler is tested using the lift simulator. The effect of using data with better accuracy such as actual door delays and predicted car call floors is described.

Chapter 8

Lift monitoring system

The real-time lift control system comprises the on-line real-time scheduler described in Chapter 7 and control policies such as lift parking This chapter introduces the monitoring system which complements the real-time control system by supplying information about the current and predicted future states of the lift system

The monitoring system consists of two parts. The first part is the on-line monitoring system which is responsible for supplying detailed information about the lift system such as lift door delays and passenger current mean arrival rate. The second part of the monitoring system provides the real-time control system with predicted information about the future state based on historical statistical data.

This chapter also investigates the performance of the prioritised A* scheduler using the lift simulator introduced in Chapter 6 The intelligent scheduler performance is compared with that of the ETQ scheduler. A series of simulation runs has been conducted with each subsequent simulation using additional data provided from the monitoring system To provide a suitable benchmark, all simulation results use the same passenger arrival rate during corresponding five minute intervals In particular, the passenger traffic generated from one simulation run using the traffic data obtained from 3/10/94 is recorded and used for all simulation runs. Note that the passenger destinations are not fixed, meaning that for each simulation car calls are generated randomly using the passenger destination model obtained from 3/10/94. This chapter also shows the effects on the scheduler of using a new adaptive parking and up peak detection policy which are based on the current and historical passenger mean arrival rates

8.1 Lift monitoring system

The lift monitoring system is responsible for gathering all the information necessary for the lift control system. The monitoring system uses this information for applying constraints and to trigger control policies such as an up peak policy. There are two types of such information, namely current and historical. The current data are listed below.

1. The number of lifts available for assignment. For example, a fully loaded lift is excluded from the list of available lifts provided to the A* scheduler. Other constraints can be applied such as loaded lifts are only allowed to open their door twice in response to new arrivals.
2. The number of car calls along the path of the lift Only when a lift has answered all its car calls in the current direction of motion are car calls in the opposite direction of motion considered.
3. Whether the lift has stopped for a landing call or a car call. If the lift stops for a landing call, then the scheduler would normally assume a car call would subsequently be generated
4. Landing call floors, false landing call floor, lift position, lift status (moving, not moving and decelerating), next possible stop, next direction and how long since a lift has been travelling.
5. Lift door delay, that is, how long it will be before a lift will close its doors. The delay is updated each time a passenger enters or departs the lift
6. How long since a landing call has been issued (passenger waiting time).
7. Passenger mean arrival rate and predicted next landing call time for the terminal floor.

The current passenger mean arrival calculation is shown in equation 8.1:

$$\textit{current passenger mean arrival} = \frac{\textit{time at which the scheduler is called - landing call time}}{\textit{number of passengers arriving during the above interval}} \quad \textbf{(8.1)}$$

Landing call time is registered when a passenger issues a landing call. Passengers are detected while entering the lift using the rules in Figure 5.22. The landing call time is only updated when more than one passenger arrives in response to this landing call and the lift loading those passengers closes its doors. If only one passenger arrives or a lift reopens its doors, then.

- the previous landing call time remains the starting reference point for the duration of observation, and
- the number of passengers assuming to have arrived since the previous landing call is increased while the previous landing call time remains unchanged.

When more than one passenger has arrived and the loading lift door is closed, the landing call time is updated and only the number of passengers that have arrived since the last landing call is considered. This means that the observation period varies according to traffic intensity. This way of smoothing the calculation of the passenger mean arrival provides the lift system with a means of detecting the start of peak traffic periods, and avoids false peak period alarms resulting from sudden traffic fluctuation.

The next landing call time is calculated while the loading lift door is open and more than one passenger is entering the lift.

next landing call time = current passenger mean arrival + time interval until lift closes its door **(8.2)**

Both current passenger mean arrival and next landing call time are used to detect up peak periods as will be explained in Section 8.4.

The historical data provided by the lift monitoring system consist of the historical passenger mean arrival rate and the predicted car call floors. These are extracted by choosing from previous days a traffic pattern similar to the present day. The present day's traffic behaviour is compared with samples of other days' traffic and the best match is chosen as explained in Section 5 3. The mean arrival rate is calculated for all floors using the rules shown in Figure 5.22. Predicted car calls for the simulation runs in this chapter used the traffic data produced in the Kodak building on 4/10/94, Section 5 3.

8.2 ETQ and prioritised A* scheduler

In this section, a comparison of the ETQ scheduler and the prioritised A* scheduler is given. To understand the lift system behaviour and observe the differences in performance the following comparisons are discussed· landing call count, passenger queues and landing call waiting times for several simulation runs. The tests also include a comparison of different simulation runs using the A* scheduler. In the sequence of simulation runs, a series of modifications are made to the data provided by the monitoring system in order to observe the effect of providing the A* scheduler with gradually improving quality of data All the A* scheduler simulations in this section use the same parking policy as that used in the ETQ scheduler, Section 6.4.1. Note that ETQ and simulated ETQ always use both up peak and parking policies.

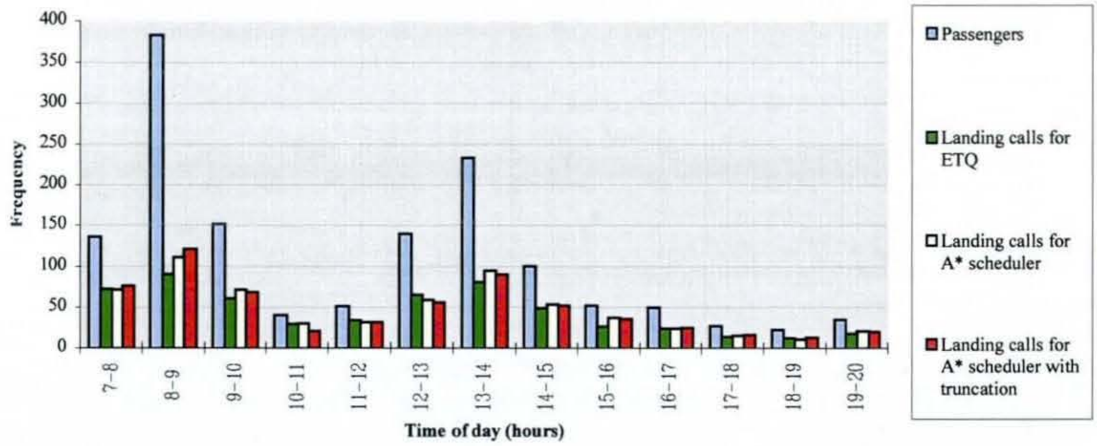
Figure 8.1 shows the numbers of landing calls for the terminal floor and the second floor. The number of landing calls shown results from three simulation runs using fixed traffic rates. The first simulation run used the ETQ scheduler, the second used the A* scheduler and the third the A* search is halted (truncated) when all available lifts have their first assignment and the number of assigned landing calls is no more than six. The first observation that can be made from Figure 8.1(a) is that the number of landing calls is slightly greater in peak periods when the A* scheduler is used This is probably due to the up peak policy during which the ETQ sends two lifts to the terminal floor where they park with doors open ready to receive passengers Consequently, passengers will enter lifts without issuing a landing call.

The A* scheduler in Figure 8.1 uses a fixed door delay, the predicted car call is assumed to be at the furthest floor in the direction of the lift, waiting time is not included in the lift trip duration to a landing call and no up peak policy is used However, all the other information listed in items (1) to (4) in Section 8.1 is used.

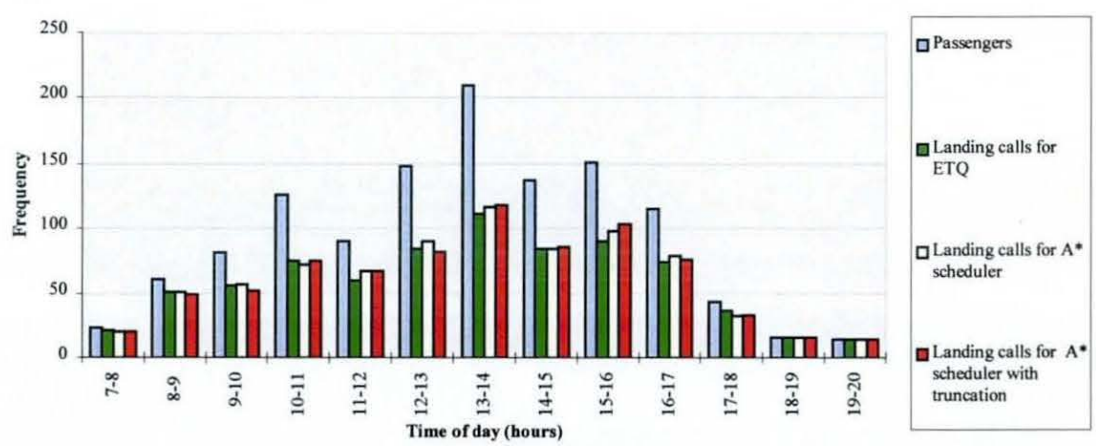
The remaining items (5) to (7) were introduced one at a time to the A* scheduler so that their effects can be observed in isolation. Figure 8.2 shows landing call frequency at the terminal floor for five simulation runs all using the truncated A* scheduler The first simulation run uses the A* scheduler as described above, the second run uses current lift door delays, the third run adds passenger waiting time and the fourth uses predicted car calls. The fifth run uses an up peak policy similar to the ETQ up peak policy explained in Section 6.4.2. The simulation runs in Figure 8.2 have a similar landing call frequency except when using the up peak policy where a reduction in the landing call frequency is evident during periods of intensive traffic. The comparison between A* with up peak policy and ETQ simulation runs shows that their performances are similar, Figure 8.3. Appendix G shows further detailed examples for actual and simulation tables of passengers and the number of landing calls for both the terminal and second floors.

Since the landing call frequency does not show a significant performance difference, the next step is to compare the passenger queues Using the ETQ scheduler, Figure 8.4 shows the average queues and maximum queue length for all floors in the building. The queues are calculated every five minutes for both up and down calls. For Figure 8.4(a), the calculation of the intervals used in estimating of the average queue length begins when a landing call is issued and ends when the lift doors are opened. When calculating the longest queue (Figure 8.4(b)), the observation of a queue starts when a passenger issues a landing call and ends when a landing call is cancelled. Since the observation period for Figure 8 4(a) is longer than Figure 8 4(b), then it is possible for a queue shown in Figure 8.4(a) to be longer than the same queue in Figure 8.4(b) within the same five minutes observation period.

Figure 8.5 shows the same queue calculations for a simulation run using the truncated A* scheduler with enhanced information and up peak policy. The comparison of Figure 8.4 with Figure 8.5 illustrates that the A* scheduler provides a reduction in the number of queues and their lengths. A comparison between ETQ and five versions of the A* scheduler is given in Figure 8.6. From these results it is clear that with each enhancement the number of long queues is reducing while the number of shorter queues is increasing At this point no difference in performance in terms of the number of queues is found between the full A* search and the truncated A* search. The details of other simulation runs can be found in Appendix H.

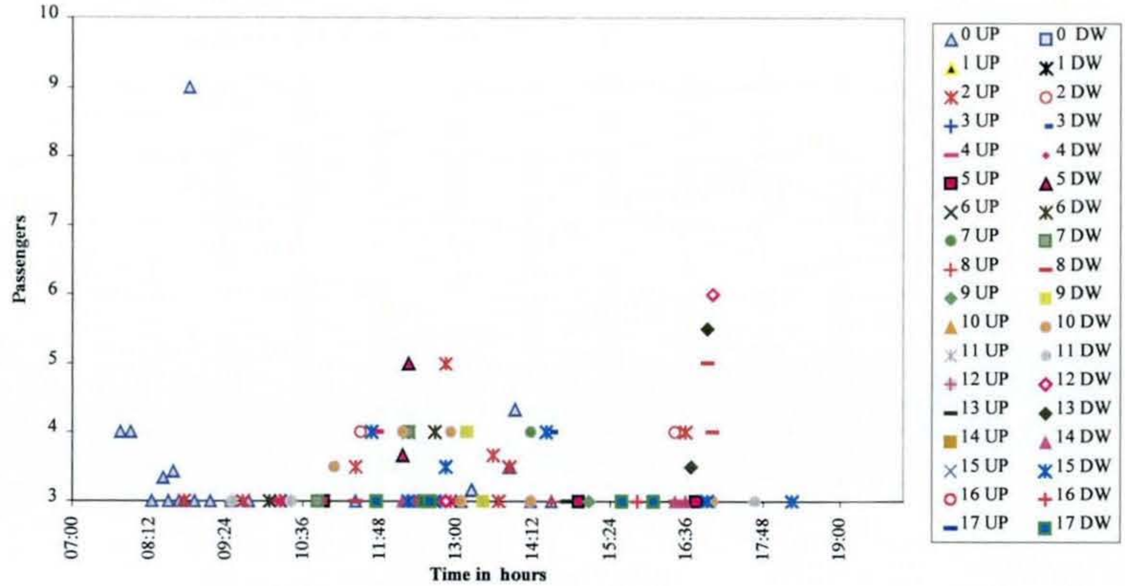


(a) Terminal floor

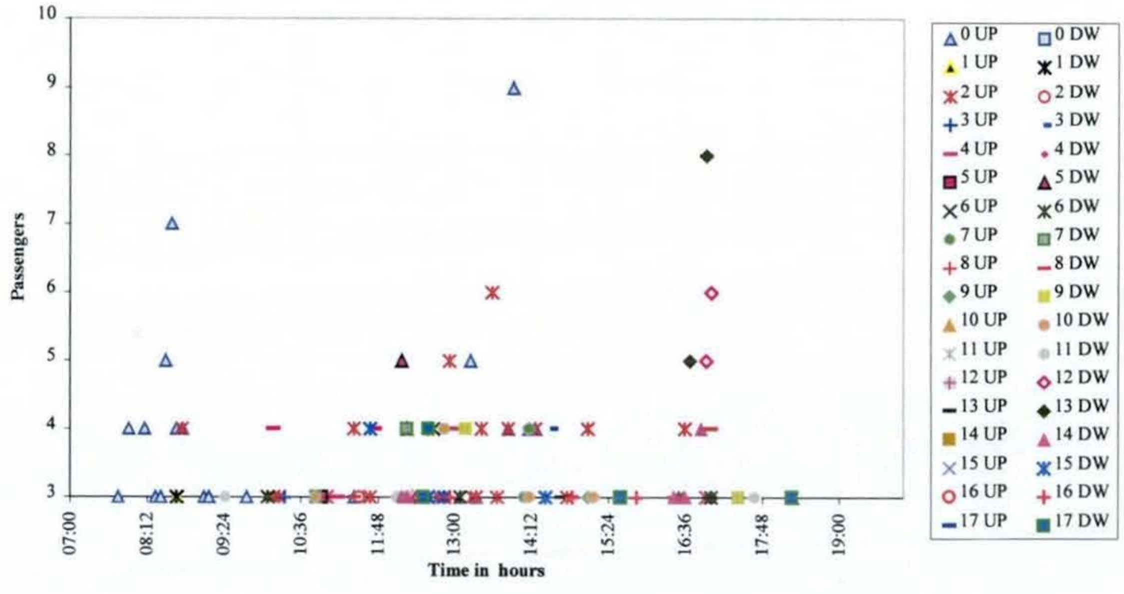


(b) Second floor

Figure 8.1 Comparison of landing call frequencies for three simulation runs using fixed passenger rates between 7am and 8pm on 3/10/94.

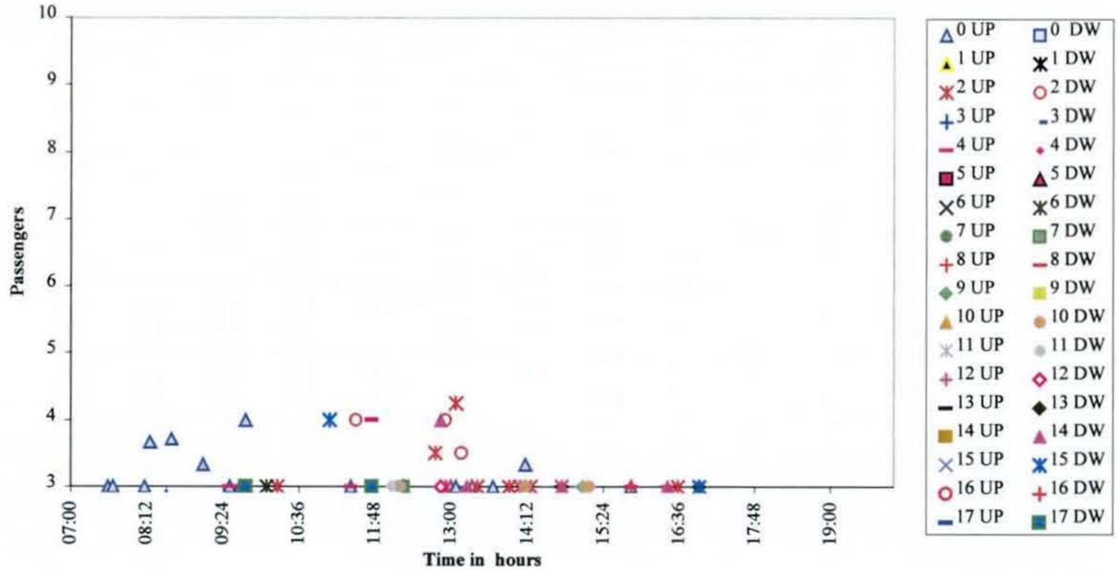


(a) Average passenger queues

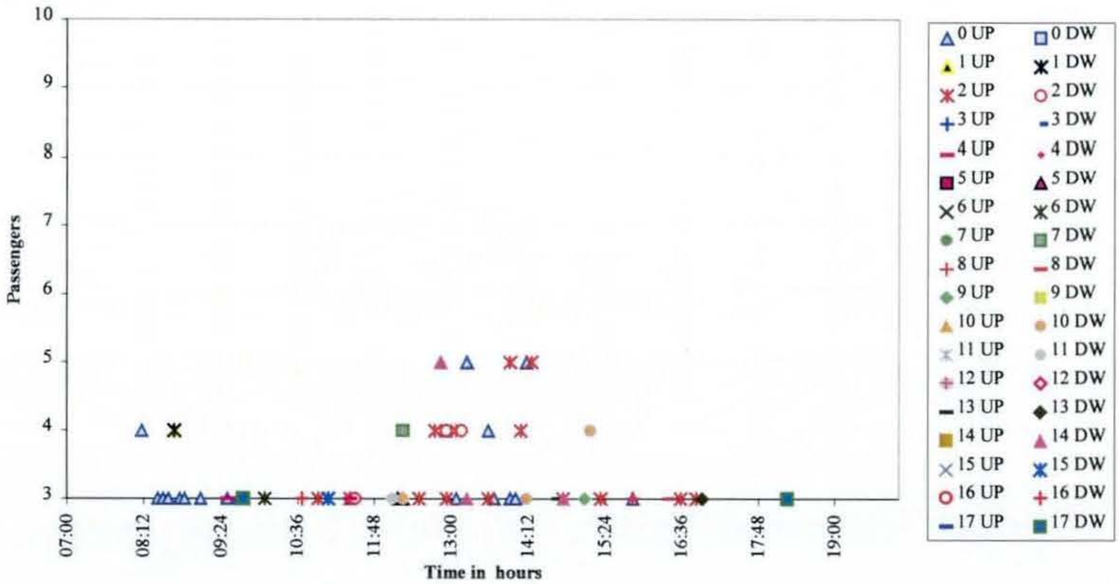


(a) Longest passenger queues

Figure 8.4 Average and longest passenger queues calculated for every 5 minute interval between 7am and 8pm on 3/10/94. The queues are measured at each floor for both UP and DOWN (DW) directions. The simulation was conducted using the ETQ scheduler.



(a) Average passenger queues



(b) Longest passenger queues

Figure 8.5 Average and longest passenger queues calculated for each 5 minute interval between 7am and 8pm on 3/10/94. The queues are measured at each floor for both UP and DOWN (DW) directions. This simulation run was conducted using an A* scheduler with truncation, enhanced door status, landing call waiting time added to lift trip time, car call floor prediction and up peak policy.

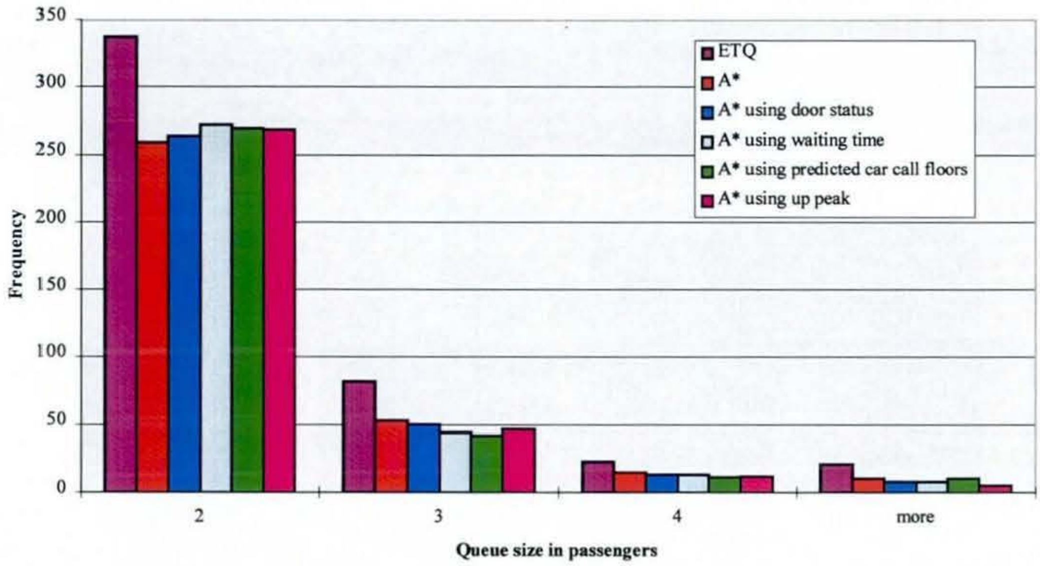


Figure 8.6 Comparison between queue frequencies for all floors between 7am and 8pm on 03/10/94. The queues are for six simulation runs, ETQ and 5 simulation runs using the truncated A* scheduler.

Since the main aim of the current work is to minimise passenger average waiting time (WT), this parameter needs to be compared for both the ETQ and A* schedulers. Passenger waiting time is the time interval between a passenger issuing a landing call and a car arriving in response. Table 8.1 shows a comparison between the actual passenger waiting time (Actual ETQ) and the simulated ETQ scheduler. As explained in Section 6.5, there is approximately a three second difference between the actual and the simulation results due to the sampling delay in the recording of the actual data. The figures in the table concentrate on up peak traffic periods when the number of passengers arriving to the terminal floor is at its maximum, but it also gives the average WT for the entire simulation period.

Similarly, Table 8.1 shows the WT when the A* scheduler is used Eight simulation runs have been conducted all showing similar performance except for the following cases.

- When up peak is switched on, lifts are sent to the ground floor before passenger arrival thereby minimising passenger waiting time. The up peak policy used is similar to the up peak policy used for ETQ (Section 6 4).
- As extra information is provided by the monitoring system, there is a gradual reduction in the number of WT intervals longer than 20s.

Simulation runs using the A* scheduler without up peak policy show a small increase in WT when compared to ETQ, while the A* scheduler with the up-peak policy enabled shows a similar performance to ETQ.

Table 8 2 shows the WT for all floors except the terminal floor. The table shows WT for the actual lift system and the simulation results using both ETQ and A* schedulers The difference in performance between the actual and simulated ETQ is probably because the simulated ETQ does not have a down peak policy, resulting in longer passenger WTs. The A* simulation runs show results similar to the actual ETQ results bearing in mind the observation delay, Section 6 5.

Terminal floor	ETQ Actual	ETQ Simulation	A* simulations 1	2 T DD	3 DD WT	4 T DD WT	5 DD WT PCC	6 T DD WT PCC	7 T DD WT PCC UP PEAK	8 T DD WT PCC UP PEAK
Average WT between 7-10 AM	8 758	5 708	7 367	6 692	7 202	6 558	6 603	6 593	5 346	5.448
Average WT between 7-8 AM	7 310	3 574	4 929	4 532	4 592	3 865	3 866	4 122	3 430	3 886
Average WT between 8-9 AM	10 294	6 111	8 202	6 847	8 449	7 373	7 746	7 531	5 879	4 781
Average WT between 9-10 AM	8 456	7 663	8 500	8 735	7 672	8 005	7 516	7 364	6 757	8 189
Average WT between 12-13 PM	6 982	6 568	7 211	10 058	8 278	9 376	7 962	7 196	8 001	6 984
Average WT between 13-14 PM	9 000	11 156	9 581	9 572	10 592	9 485	9 961	11 008	7 164	7 291
Average WT between 14-15 PM	9 192	9 620	9 749	13 038	12 141	10 158	10 648	12 714	8 391	11 591
Longest WT between 7-10 AM	32 000	42 556	44 608	42 725	30 348	33 890	35 990	48 641	36 381	55 583
Number of WT intervals between 7-10 AM	193	222	253	262	262	281	262	272	218	222
Number of WT intervals >15s 7-10 AM	20	21	39	31	35	40	34	36	22	19
Number of WT intervals >20s 7-10 AM	6	14	23	12	14	15	18	14	9	6
Number of WT intervals >25s 7-10 AM	3	9	12	6	10	8	8	4	3	4
Number of WT intervals >30s 7-10 AM	1	4	7	3	1	3	2	3	2	2
Average WT between 7AM - 8 PM	9 000	7 260	7 817	7 969	8.369	7 498	7 432	7.862	6 643	6 628

☞*Key: DD = Door Delay, WT = Waiting Time, PCC = Predicted Car Calls, T = Truncated*

Table 8.1 Waiting time (WT) for the terminal floor when using ETQ and A* schedulers. All intervals are in seconds.

Note that the A* scheduler achieved similar results to those of the actual ETQ without the need for a special down peak policy, and the up peak policy does not affect the passenger waiting time at other floors

7-8 PM	Actual ETQ	ETQ	A*0	T A*1 DD	A* 2 DD WT	T A*3 DD WT	A*4 DD WT PCC	T A*5 DD WT PCC	T A*6 DD WT PCC UP PEAK	T A*7 DD WT PCC UP PEAK
Longest WT	72	279	83	75	76	118	85	84	75	68
Number of WT intervals	3737	4609	4842	4767	4789	4790	4761	4773	4793	4795
Number of WT intervals >15s	1060	1606	844	784	766	808	806	814	755	795
Number of WT intervals >20s	561	1141	407	384	367	420	393	412	388	378
Number of WT intervals >25s	285	855	220	207	210	232	222	213	195	225
Number of WT intervals >30s	163	657	118	123	130	152	123	123	107	141
Number of WT intervals >60s	4	167	1	2	5	6	3	6	3	4
Average WT	12 827	16 396	9 850	9 690	9 703	9 867	9.780	9 823	9 646	9 882

Key · DD = Door Delay, WT = Waiting Time, PCC = Predicted Car Calls, T = Truncated

Table 8.2 Landing call waiting times for all floors except the terminal floor. The intervals are actual intervals (Actual ETQ), simulated ETQ (ETQ) and the other columns show simulation results using A* scheduler. The A* scheduler achieved a similar performance to ETQ without an up peak policy. A new adaptive parking policy and up peak policy are discussed in the following sections with the intention of further improving the performance of A* scheduler.

8.3 Adaptive parking

Parking policies are used to prevent lifts from bunching at one floor or at adjacent floors, by distributing free lifts around the building at popular floors. Section 6.4.1 explains the parking policy used in all the simulation runs previously discussed in this thesis, but, in brief, the building is divided into six fixed zones and free lifts are assigned to park in each zone. This section discusses the effect of using the historical passenger mean arrival rate gathered by the monitoring system to predict popular floors. A traffic model for 4/10/94 is used for predicting the next arrivals of passengers, and is generated using the rules of Figure 5.22, that is, they are obtained in the same manner that the traffic model for 3/10/94 is generated for the lift simulator. For the terminal floor, both historical and current passenger mean arrival rate are used to assess its popularity in comparison with other floors.

The number of zones allowed in a building should be no more than the number of lifts The order in which zones are allocated lifts is determined according to zone popularity, where the most popular are these floors with the lowest passenger mean arrival rate. As the passenger mean arrival rate is changed every five minutes in the simulator, the parking zones are updated at the same frequency. Table 8 3 shows the six most popular floors over a number of consecutive five minute intervals during a morning period. Each zone for parking is based around the popular floor and is banded by adjacent popular floors (or the terminal or top floors).

The lift parking status is updated after every scheduler run as described below.

- The lifts which are free to be parked at popular floors are identified. A free lift is a lift which has no assignments or car calls with a direction status equal to PARK. If more than one lift is parked in a zone, then the lift which is parked the furthest from the most popular floor in the zone is marked free.
- Mark the unoccupied zones as free zones. A free zone is a zone which has neither a lift parked within it nor a lift without assignments heading towards it
- Distribute free lifts amongst free zones starting with the highest priority zone currently free

- If there is a lift parked at a zone and no other free lift is heading towards this zone, then park this lift at the most popular floor of the zone.
- At up peak, if there are two parked lifts at the terminal floor and the terminal floor is the most popular floor in the building, the second lift is not moved.
- If the terminal floor is the highest priority floor with no lifts parked and no free lifts are available to send to it, then a parked lift at the lowest priority zone is sent to the terminal floor.

Table 8.4 shows the effect of using the adaptive parking on WT for four simulation runs all using the A* scheduler with no up peak policy. The first simulation run used only the historical passenger mean arrival rate for all floors while the other simulation runs used both the current and the historical passenger mean arrival for the terminal floor only (other floors used the historical passenger mean arrival rate). This was implemented in this way so as to ensure that the terminal floor is given the appropriate priority in case there is a change in the arrival pattern

Examining Table 8.4 in relation to previous results (Table 8.1), demonstrates that the WT has been reduced particularly between 7am and 10am from about 5.4s using ETQ to 4s using A*. Note that this improvement is achieved without up peak policy switched on.

8.4 New up peak detection policy

In this section, the passenger current and historical mean arrival rates for the terminal floor are used to anticipate the increase in traffic and the next landing call time. Using the passenger mean arrival for the parking policy improved the performance of the lift system in terms of passenger average waiting time at the terminal floor, giving the motive to use again the passenger mean arrival rate to trigger the up peak policy.

The current and historical mean arrival and the next landing call time (Section 8.1) are used to trigger the up peak policy if any of the following conditions apply

- The passenger current or historical mean arrival rate is less than the maximum trip cost in the last schedule produced. The time a lift may take to answer a landing call is termed the assignment cost. The maximum assignment cost is taken from the last A* scheduler assignment table, for example, Table 7.17. If the maximum cost is greater than the assumed mean arrival then this means that the anticipated landing call will affect the current schedule. That is the lift with the maximum assignment cost will not be free to answer the landing call.

- The next landing call time is less than the longest trip cost.
- The up peak policy remains in force while the number of lifts parked at the terminal floor is less than a specified target value Two lifts are required at the terminal floor at up peak for the Kodak building.

The up peak policy is switched off if none of the above conditions is satisfied. When the up peak policy is on, it is effected by the monitoring system which issues a false landing call to the scheduler which would then need to assign a lift. This means that even if all lifts have assignments (not free for parking), a lift will still be assigned to go to the terminal floor.

Table 8.5 shows the result of using the new up peak detection policy. When comparing these results with those of ETQ (Table 8.1), the results show that the WT at the terminal floor has now been reduced from 5.5s to 3.5s between 7am and 10am and the WT for the entire simulation period is reduced from about 7s to 5s. The results highlight the importance of parking lifts at the high demand floors. One drawback is that when there is no new assignment and all lifts are busy answering calls, then the lift scheduler would produce no assignment and the maximum trip cost is zero. Under such circumstances an alternative approach could be used in which the triggering of up-peak could be caused by evaluation of the longest trip time rather than the longest assigned trip cost. The comparison would then involve considering the relative values of passenger mean arrival and next landing call time

All results in Table 8 5 are from a truncated A* search, that is where the search has been interrupted when six landing calls have been assigned. This is true for all simulation runs except simulation run seven, where the condition changes to limit the search tree to the lifts' first assignments, meaning that the search tree would have only six levels. The performance of the A* scheduler does not show any significant change when truncation is applied. This may be explained by considering the simulation log of Appendix J in which the average number of nodes generated during a full simulation run is illustrated The log files show no significant difference in the average number of nodes produced. These results can be compared with example 10 in Appendix F, where 46 nodes and 17 tree levels were generated for the assignment of 6 landing calls using the Kodak building configuration.

Several tests were conducted as explained in Chapter 7 to assess the search optimality. The same examples were conducted using three different methods of calculating the heuristic value h, all of which produced the same schedule for the same example. Another test was done where two A* schedulers were running simultaneously using the simulator and the monitoring system data in calculating the paths One scheduler used the actual cost of a lift travelling to a landing call as explained in Section 7.7 while the second one used only the direct path to a landing call (including passenger waiting time) as shown in Figure 7 9. The second scheduler generates a greater number of nodes and it was found that the constraints of the computer system made it impossible to produce a complete schedule if more than four simultaneous landing calls were generated. The results show that in the 6700 schedules produced during the simulation period of 18 minutes, only two schedules were not identical Such differences occurred for a small number of consecutive scheduling cycles before the schedules once more agreed. During the periods when different schedules were produced, it was clear that the scheduler using actual cost provided the better assignment in comparison to the assignment provided by the direct path scheduler which changes to become the same as that of the total cost scheduler. Similar results were obtained when the above experiment was repeated using a $h=0$ when calculating the path cost of the second scheduler.

8.5 Conclusion

This chapter has demonstrated that a scheduler based on the A* search technique and using predicted data can give a clear improvement in performance with respect to the ETQ scheduler. The A* scheduler without an up peak policy has achieved a passenger average waiting time of about 7.8s, which compares favourably with the 15s average waiting time considered acceptable in a business type building [Barney and dos Santos 1985]. Both a full A* search and a truncated A* search were used, but the results in terms of WT were very similar, due to the low number of simultaneous landing calls produced. This may be due at least in part to the scheduler's fast response, as a good lift control system should not allow many passengers to remain waiting simultaneously. The truncated A* may prove to be useful in heavier traffic conditions.

The results also show the importance of having high quality data supplied to the A* scheduler. Using detailed data in the calculation of the heuristic function has a clear effect in improving the performance of the scheduler, as is apparent in the reduction of number of long queues Since WT is in fact the landing call waiting time, the improvement in this parameter is less obvious when comparing the scheduler performance in terms of WT.

The prediction of passenger arrival allows the generation of a false landing call in order to minimise passenger waiting time in peak traffic periods. A false landing call would force the scheduler to assign a lift for the terminal floor even if all lifts are busy. However, the A* scheduler is only able to assign one lift to a floor, that is, no more than one lift can be assigned to go to the terminal floor at up peak. Under certain circumstances it may be prudent to send a second lift to the terminal floor using a false car call. In such a case, a search may be needed to choose the lift most suitable to answer the false car call.

The detection of the up peak period may be further improved by detecting its onset using the longest lift trip as well as the longest assignment cost. This is especially useful when all lifts are busy and no new landing calls have been issued to trigger a new schedule.

The results also show the effect of the adaptive parking and up peak policy on the performance of the lift system. The use of current and historical passenger mean arrival rates to trigger parking and up peak both show a significant improvement in performance with respect to the ETQ parking and up peak policies, and the passenger average waiting time reduced to about 5s at the terminal floors for the entire simulation period. The introduction of the adaptive parking and up peak policies show no significant effect on the performance at other floors, Tables 8 5 and 8 6

Another improvement to the adaptive parking approach could be implemented when no free lifts are available. The present policy is that a lift which is parked at a lower priority zone is not moved to the highest priority free zones unless it is the terminal floor. The reason for not using the same policy for all floors is to avoid too many lifts moving within the building when there are no or few landing calls being issued. If energy consumption is considered to be of less importance than that of the reduction of WT, then the adaptive parking policy can be modified so that free lifts are always distributed to park at the highest priority zones first.

Time	Priority 1	Priority 2	Priority 3	Priority 4	Priority 5	priority 6
07 05	0	2	4	7	10	14
07 10	0	6	10	15		
07 15	0	2	9	14		
07 20	0	4	14			
07 25	0	17	14			
07 30	0	17	6	7		
07 35	0	2	4			
07 40	0	2	8			
07 45	0	2	4	7	10	14
07 50	0	2	16	8	9	
07 55	0	5	6	2	10	12
08 00	0	2	8			
08 05	0	2	8	11	1	5
08 10	0	2	4	9	12	14
08 15	0	2	4	6	9	12
08 20	0	2	9	1	5	7
08 25	0	2	10	14	16	1
08 30	0	2	15	1	10	11
08 35	0	2	16	5	7	9
08 40	0	10	2	9	3	4
08 45	0	2	8	1	11	9
08 50	0	2	11	10	5	9
08 55	0	2	4	13	5	6
09 00	0	2	10	14	1	6
09 05	0	6	14	13	15	16
09 10	0	2	1	4	5	7
09 15	0	3	17	2	4	8
09 20	0	2	1	15	9	4
09 25	0	2	3	9	5	12
09 30	0	2	16	4	9	14
09 35	0	10	1	2	9	6
09 40	0	2	17	1	15	3
09 45	0	2	10	4	7	12
09 50	0	5	1	2	3	9
09 55	0	15	2	11	8	9
10 00	2	0	7	10	15	4
10 05	2	4	11	0	3	12
10 10	2	0	17	4	15	3
10 15	2	0	10	13	3	5
10 20	2	0	15	11	5	14
10 25	10	0	2	16	4	5

Table 8.3 Popular floors selected for each 5 minutes between 7:05am and 10:25am on 3/10/94 and listed according to their priority. Priority1 is the most popular zone. Appendix I lists popular floors between 7:05am and 7:55pm.

Terminal floor	A* simulations			
	1	2	3	4
Average WT between 7-10 AM	3 760	4 235	4 188	3 954
Average WT between 7-8 AM	1 288	0 981	1 217	1 235
Average WT between 8-9 AM	4 932	5 388	5 020	4 360
Average WT between 9-10 AM	4 442	5 631	5 626	6 014
Average WT between 12-13 PM	7 715	9 370	8 913	8 150
Average WT between 13-14 PM	10 637	9 507	8 975	8 072
Average WT between 14-15 PM	10 900	9 470	10 860	11 591
Longest WT between 7-10 AM	28 238	52 624	30 491	37 707
Number of WT intervals between 7-10 AM	260	257	273	263
Number of WT intervals >15s 7-10 AM	18	21	27	19
Number of WT intervals >20s 7-10 AM	6	11	13	11
Number of WT intervals >25s 7-10 AM	2	7	6	7
Number of WT intervals >30s 7-10 AM	0	4	1	4
Number of WT intervals >30s between 7 AM -8 PM	10	21	20	13
Average waiting time between 7 AM- 8 PM	6 435	6 355	6 237	6 305

Table 8.4 WT using the A* scheduler with an adaptive parking policy, shown for four separate simulation run results. The adaptive parking policy in the first simulation run used only the historical passenger mean rate while the other simulation runs used both current and historical passenger mean arrival rate. Up peak policy was not used.

Terminal floor	A* simulations						
	1	2	3	4	5	6	7
Average WT between 7-10 AM	3 575	3 293	3 421	3 763	3 550	3 054	3 120
Average WT between 7-8 AM	1 211	0 837	1 119	1 275	1 023	0 871	0 821
Average WT between 8-9 AM	4 633	4 153	4 071	3 713	4 219	3 899	4 059
Average WT between 9-10 AM	4 028	4 584	4 559	6 960	4 893	4 338	4 169
Average WT between 12-13 PM	4 892	8 027	8 046	7 100	6 933	5 987	7 173
Average WT between 13-14 PM	6 660	7 371	7 960	5 334	7 556	5 714	7 298
Average WT between 14-15 PM	7 216	6 048	10 117	5 895	7 754	6 221	11 137
Longest WT between 7-10 AM	62 291	30 233	42 071	41 034	38 425	34 837	33 44
Number of WT intervals between 7-10 AM	232	249	248	254	253	250	245
Number of WT intervals >15s 7-10 AM	13	14	19	20	19	13	17
Number of WT intervals >20s 7-10 AM	9	7	11	12	9	6	10
Number of WT intervals >25s 7-10 AM	5	5	4	8	4	3	5
Number of WT intervals >30s 7-10 AM	5	1	4	4	4	1	1
Number of WT intervals >30s between 7 AM- 8 PM	12	8	15	11	16	7	5
Average WT between 7 AM-8 PM	5 097	5 420	5 714	4.974	5 464	4 682	5 452

Table 8.5 WT when A* scheduler is used with an adaptive parking policy and up peak parking. The first three simulation runs used only the passenger mean arrival rate, while the last four simulation runs used both passenger mean arrival rate and expected next landing call time.

All floors	A* simulations						
7-8 PM	1 no up peak	2 no up peak	3	4	5	6	7
Longest WT	78	83	83	94	96	109	77
Number of WT intervals	4773	4792	4810	4765	4785	4773	4810
Number of WT intervals >15s	784	782	812	805	795	825	829
Number of WT intervals >20s	378	375	430	443	417	444	443
Number of WT intervals >25s	223	188	233	257	230	263	257
Number of WT intervals >30s	141	98	139	159	136	169	158
Number of WT intervals >60s	5	5	8	8	8	8	6
Average WT	9 76	9 59	9 89	9 98	9 84	10 15	9 98

Table 8.6 WT for all floors except the terminal floor. The first two simulation runs used only adaptive parking, while the other simulation runs used up peak parking.

Chapter 9

Conclusions and future work

9.1 Conclusions

The aim of the current work is to assess new scheduling approaches and intelligent monitoring techniques, resulting in improvement in the performance of existing installations This requires the optimisation of lift operation and lift car assignments in a variety of different traffic modes in order to minimise passenger waiting time. To achieve this, a review was undertaken of conventional lift systems and intelligent techniques normally applied in lift systems and other real-time control systems. This resulted in the development of a new intelligent lift scheduling system which was demonstrated to improve on the existing operational performance when applied to a simulation of an installed lift system. The contributions of the research carried out in this work are summarised below and are discussed in detail in the following subsections.

- A model of passenger movements has been developed from an analysis of data gathered from installed lift systems, thereby allowing the realistic simulation of landing calls, car calls and door opening times.
- The passenger arrival pattern was shown to follow a Poisson distribution.
- A lift simulator has been implemented to allow the modular comparison of alternative scheduling and monitoring approaches and to provide an accurate model of lift dynamics
- The work has introduced and implemented a real-time prioritised A* search which was used as the real-time lift scheduler.

- An intelligent real-time lift control system was introduced and implemented. This consists of two distinct parts, namely the real-time control system and the monitoring system.

9.1.1 Modelling of passenger movements

In the current work, the availability of data from an existing lift system has allowed the extraction of a lift traffic model which does not involve the identification of specific traffic patterns such as up peak and down peak The development of the traffic model involved estimating the number of passengers entering a lift which was not directly observable using car calls and photocell activations. The passenger mean arrival rate for each time window and floor is then calculated and used to generate landing calls. It has also been possible to derive for each floor the distribution of car calls that result from landing calls

The method used for counting passengers arriving at the lift system was

- to determine the traffic model for the lift simulator for landing and car call generation and passenger movement simulation;
- to calculate the historical passenger traffic models for the prediction of popular floors;
- to generate predicted car calls.

The simulated traffic produced, within experimental error, the same number of landing calls with the simulated ETQ as those of the actual lift system. This confirmed that the method is appropriate for its purpose

The literature survey revealed that several methods of counting passengers had been implemented and that these make use of lift load information. The method of counting passengers used in this project does not require load sensor information This means that the current method can be applied without modification to the lift system and without the introduction of expensive high quality load sensors.

9.1.2 Passenger arrival pattern

Due to the availability of actual data from a real installation, a comprehensive analysis of the events that occur during the operation of a lift system could be made. This has allowed a detailed assessment of the assumption made by a number of authors, namely that the passenger arrival rates follow a Poisson distribution. The distribution of intervals between car departure and next landing calls was found to follow an exponential distribution. An appropriate exponential random number generator was developed to simulate passenger arrival using the mean arrival rate supplied by the passenger arrival model.

9.1.3 Lift simulator

A lift simulator was implemented to:

- aid the understanding of the lift system under different traffic patterns;
- test the operations of the lift scheduler, monitoring system and other lift control policies that form the lift control system;
- assess the relative performances of alternative scheduling strategies,
- assess the importance of having data of improved quality supplied to the scheduler, such as load status and lift door delay

A comprehensive lift simulator was implemented which can adapt to different buildings and lift configurations, interactively visualise a lift system with animation, make use of the passenger arrival model, accurately simulate time delays such as door opening and closing, dynamically model the motion of lifts, and be used with different scheduling algorithms.

The lift simulators found in the literature review generally use a number of simplifying assumptions such as a constant time being assumed for travel between floors regardless of trip length. In this work emphasis was given to producing a realistic dynamic simulation of the lift system thereby reducing errors which arise as a result of accumulated delays.

9.1.4 Real-time prioritised A* search

The real-time lift scheduler was implemented using a prioritised A* search which has the following characteristics.

- The search was named in this work as 'prioritised A* search' due to its tree structure which gives the priority to the lifts' first assignments. The search tree is divided into stages, in which each stage has a number of levels equal to the number of lifts The number of nodes in each level is the same as the sum of the number of unassigned landing calls and a single free node. The first stage in the tree shows the first assignment for all lifts, the next stage gives their second assignments, and so on Since the first stage has the immediate assignment required by the lift system, then it is possible to interrupt or halt the search at this stage and provide the lifts with their first assignments. The lifts' future assignments can be refined in subsequent scheduler cycles when updated information regarding the state of the lift system becomes available.

- Experimental work was performed to assess the optimality of the schedule produced by the prioritised A* search with respect to the present definition of the state space. The use of good heuristic functions allowed the search to converge immediately to the goal state without the need for backtracking Hence, the search tree can be interrupted at any level and still give an optimal schedule.

- It follows that it must be possible to know the number of nodes required to produce the first assignment of each lift and consequently the time the search will take.

- With regard to the above characteristics, the search can be described as real-time since the maximum computational time for full search is 110ms, Appendix J. This meets the requirement of the lift system in the Kodak building, since schedules produced in under 250ms are considered suitable by Express Evans Ltd.

- In the prioritised A* search there is no need to detect specific traffic patterns and alter the scheduler policy accordingly.

9.1.5 Intelligent real-time lift scheduling system

The objective to minimise passenger waiting time was achieved; the intelligent real-time scheduler reducing passenger average waiting time at the terminal floor for the entire simulation period to about 5s compared to about 7s found for ETQ. For other floors, the improvement was demonstrated in the reduction of number of passenger waiting time intervals greater than 15s.

The real-time intelligent lift system implemented in this work has the following characteristics.

- Simple, modular, easy to configure to suit building design, fully adaptable to changes in traffic and building configuration.
- There is no need to identify separately traffic modes, since the prioritised A* search provides a single algorithm for all traffic types. In many lift systems, the detection of the onset of up peak is found by determining when the load percentage and the number of car calls both increase beyond a predefined value. In this work the monitoring system is able to detect periods of heavy traffic by monitoring current and historical passenger arrival rates without the need to use arbitrary pre-set values.
- The monitoring system uses prediction based on simple statistics of passenger mean arrival and this is used to set priority floors
- The monitoring system is responsible for providing the real-time scheduler with information regarding the lift state such as available lifts, predicted car calls and door delay. The effect of improved data quality was apparent in the reduction of the number and length of the queues
- The use of the prioritised A* search technique without up peak or an adaptive parking policy (only a fixed parking policy was used) produced at the terminal floor for the whole simulation period an average waiting time of 7.8s. This is comparable with the corresponding time of 7s achieved by ETQ and the maximum of 15s average waiting time recommended for office buildings [Barney and dos Santos 1985] Note that the ETQ lift control system uses both up peak and parking policy.

- An adaptive parking policy was used to prevent bunching so that free lifts can park at the most popular floors Floor priorities were assigned every five minutes according to floor popularity, using the historical passenger arrival rate for each floor and both current and historical passenger arrival rates for the terminal floor. The parking policy showed a significant improvement with respect to the fixed parking policy, as the average waiting time for the entire simulation period for the terminal floor was reduced to about 6.3s.

- Additional lifts are assigned to the terminal floor by generating false landing calls when the predicted time until next passenger arrival is less than maximum lift assignment cost in the last schedule. The introduction of this policy further improved the performance of the intelligent scheduler. For example, the passenger average waiting time for the terminal floor during the worst up peak period from 7am until 10 am reduced from about 5.5s to 3.5s.

The adaptive parking policy is used as long as there are lifts free to park and there is at least one priority floor available. This simple approach gives greater emphasis to the individual popular floors rather than to a zone as a whole, in comparison with the parking policy described in Section 2.2.1. In the adaptive parking used in this work, no time restriction or limitation on the type of traffic was used, as the main goal is to minimise waiting time. This approach also appears to reduce bunching by distributing free lifts to popular floors without the need for pre-set values.

9.2 Future work

The current passenger model has not been tested on an actual lift installation The suitability of the method devised in the current work could be assessed by comparing the passenger traffic it predicts with that obtained from a real lift system.

The lift simulator could also be used to aid the design of new lift installations, by helping to choose the appropriate scheduling strategy and thereby both minimising the number of lifts required and maximising the usable floor space in the building The simulation can also help in choosing a suitable building configuration, for example the location of the restaurant floor affects the performance of the lift system during the lunch hour peak traffic period. The lift simulator can be also used to demonstrate to potential customers the performance of a proposed lift installation

Incorporating a lift simulator into a traffic monitoring system may allow one to simulate future events and feed useful and timely information to aid the production of a schedule.

In real-time applications it is necessary to be able to interrupt the search and still have the correct schedule The efficiency and the success of the prioritised A* method demonstrated in this work, still requires further tests to prove its optimality and functionality in different type of buildings and under different traffic conditions. The performance of the A* search could also be compared with schedules implementation using IDA*. IDA* was found to be faster than A* by Korf [Sharobe and AAAI 1988] and therefore a full exhaustive search may be possible using a value of $h=0$ and the scheduler is then guaranteed to be optimal.

The current prioritised A* scheduler is only able to assign one lift to a floor, and so no more than one lift can be assigned to the terminal floor at up peak. At present, the parking policy is used to send a free lift to the terminal floor. A future improvement may be possible either to extend the parking policy to send another lift to the terminal floor or to issue a false car call. In such a case another search may be needed to identify the lift most suitable to assign to the false car call

Determining when to send an additional lift to the terminal floor uses a comparison of the expected next passenger arrival time and the maximum trip cost in the last schedule. However, it is possible for all lifts to be busy answering car calls with no landing calls being present for input to the scheduler. When this occurs, a solution could be to use the longest lift trip in place of the assignment cost.

The effect of using improved data quality was exhibited in the reduction of the number of queues and their lengths, these being an indication of an improvement in performance. However, the improvement was not immediately apparent in the passenger waiting time, as the passenger waiting time is equivalent to the landing call waiting time and not the actual passenger arrival time. Certain passenger arrivals are not observed in lift systems, for example those who arrive during the time a lift is parked with its doors open. It is expected that the actual passenger average waiting time will show a significant improvement compared with the ETQ scheduler, but future experiments would need to be carried out to confirm this in simulation.

The prediction of the number of passengers waiting behind a landing call can improve the estimate of the door delay. Currently, the door delay is recalculated each time a passenger is detected and this involves the use of a timer to simulate how long a passenger would normally take to enter or leave a lift. This estimate could be improved if the number of passengers entering and leaving the lift is known and is included in the total door delay. In turn, the number of passengers waiting can be estimated using the current and historical mean arrival rate.

Further work could be carried out to improve the monitoring system in daily traffic pattern identification. For example, a neural network could be used to identify when a traffic pattern matches a historical pattern. The chosen traffic pattern for a day or for a certain period of a day can be then used for predicting passenger arrival rates as explained in Chapter 8.

At present, the monitoring system only specifies the terminal floor as a high priority floor needing a special lift scheduling policy. A future enhancement could be to provide false landing calls at other floors if a passenger arrival is expected. In addition, it is possible to extend the monitoring system so that it is able to select popular floors by using historical arrival rates obtained from a statistical analysis of several previous and similar days.

References

- Amano M , Yamazaki M and Ikejima H , *The latest elevator group supervisory control system*, ELEVECON 95, March, 1995, pp. 88-95.
- Barney G. C. and dos Santos S M., *Elevator traffic analysis design and control*, Peter Peregrinus, 2^{nd} edition, 1985.
- Barney G.C., *Lift simulation*, Elevcon 88, Karlsruhe, Germany, June, 1988, pp 22-29.
- Barney G. C., *Up peak revisited*, ELEVCON 4, May, 1992, Amsterdam, pp. 39-47
- Boullart L., Krijgsman A. and Vingerhoeds R A (eds), *Application of artificial intelligence in process control*, Oxford:Pergamon Press,1992
- Chan W. L. and So A.T. P., *Dynamic zoning in elevator traffic control*, Elevator World, March, 1997, pp. 136-139.
- Chenais P and Weinberger K., *New approach in the development of elevator group control algorithms*, ELEVCON 4, May, 1992, Amsterdam, pp. 48-57
- Whitehead S., Karlsson J and Tenenberg J., *Learning multiple goal behavior via task decomposition and dynamic policy merging*, pp. 45-78, in Connell J. H. and Mahadevan S. (eds), *Robot Learning*, Kluwer Academic Publishers, 1993.
- Crites R and Barto A., *Improving elevator performance using Reinforcement Learning*, Advances in neural Information Processing Systems 8 (NIPS8), Touretzky D. S , Mozer M. C and Hasslemo M. E. (eds.), Cambridge, MA: MIT Press, 1996, pp. 1017-1023.
- Crites R. and Barto A , *Elevator group control using multiple Reinforcement Learning agents,* Machine Learning 33, 1998, pp. 235-262.
- Express Evans Ltd. reports: Orland D. C. F., *Notes on channelling*, October 14^{th}, 1991.
- Express Evans Ltd. report: *Comparison between zoned up-peak and channelling*, BGS/91, 1991

- Express Evans Ltd. report: Orland D. C. F , *Control system commentary*, June, 1990.
- Express Evans Ltd. report: Orland D. C. F., *Operational parameters for control system sımulatıon*, Octobr, 1994
- Express Evans Ltd. report: Perumainar A. K. and Gallagher J M , *ETQ control system, concept level document*, November, 1988.
- Galpin V. C. and Rock S T , *A lıft sımulatıon prototype*, Software Practice and Experıence, 25(3), March 1995, pp 251-270.
- Gross D. and Harris C. M., *Fundamentals of queuıng theory*, John Wıley, 1974.
- Hadavi K., *A real tıme productıon schedulıng system from conceptıon to practıce*, in Zweben M. and Fox M S. (eds), *Intellıgent Schedulıng*, San Francısco: Morgan Kaufmann, 1994.
- Hamdi M , Mulvaney D J. and Sıllıtoe I. P. W., *An ıntellıgent real tıme lıft schedulıng system,* 14th Workshop of the UK Planning and Scheduling Special Interest Group, Coventry, UK, November, 1995
- Hamdi M. and Mulvaney D. J., *Intellıgent real time schedulıng and ıts applıcatıon to lıft systems*, IASTED International Conference on Artificial Intellıgence and Soft Computing, Banff, Canada, July, 1997.
- Hamdı M. and Mulvaney D. J., *Sımulatıng of lıft systems and modellıng of passenger movements*, The International Journal of Elevator Engıneers, Vol. 2., 1998, pp. 1-18
- Hamıdzadeh B. and Atif Y , *Tıme controlled dynamıc schedulıng of aperiodıc real-tıme tasks*, Engineerıng Complex Computer systems, Proceedıngs, 2nd IEEE ınternational conference, 1996, pp. 323-330.
- Hummet G. T., Moser T. D. and Powell B. A , *Real tıme sımulatıon of elevators*, IEEE Winter Sımulatıon Conference, part 2, Miami Beach, Florıda, December, 1978, pp.393-402.

- Imasaki N., Kiji J., and Endo T., *A Fuzzy Neural Network and its applications to elevator group control*, IFES 91, Fuzzy Engineering Towards Human Friendly Systems, 1991, pp.1126-1127.
- Imasaki N , Kubo S. and Nakai S., *Elevator group control system tuned by fuzzy neural network applied method*, Proceedings of the International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The 2nd International Fuzzy Engineering Symposium, Vol. 4, March 20-24, 1995, pp. 1719-2342.
- Kim C , Seong K., Lee-Kwang H. Kim J. and Lim Y. B., *A Fuzzy approach to elevator group control*, IEEE Transaction of Systems, Man and Cybernetics, Vol. 25, No. 6, June, 1995, pp. 985-990.
- Kim C., Seong K., Lee-Kwang H. and Kim J., *Design and implementation of Fuzzy elevator group control system*, IEEE Transaction of Systems, Man and Cybernetics-part A: Systems and Humans, Vol. 28, No. 3, May, 1998, pp. 277-287.
- Kubo S , Nakai S , Imasaki N. and Yoshitsugu, *Elevator group control system with Fuzzy Neural Network model*, ELEVCON 95, March, 1995, pp 11-20
- Korf R. E , *Search· a survay of recent results*, pp. 197-237, in Shrobe H. E. (ed), *Exploring Artificial Intelligence*, California. Morgan Kaufmann, 1988.
- Korf R. E., *Real time heuristic search*, Artificial Intelligence, 42, 1990, pp. 189-211.
- Lustig A., *Lift simulation program*, Elevator World, March, 1986, pp.74-79.
- Mehta P , Clark G. and Prowse R , *Knowledge-Based elevator controller*, Control 94, Conference Publication No 389, IEE, 21-24 March, 1994, pp 42-47.
- Mehta P., Alani A. F , Stonham J and Prowse R., *Performance optimisation of Knowledge-Based elevator group supervisory control system*, Elevator Technology 6, ELVCON 95, March, 1995, pp. 114-121.
- Montana D , Bidwell G. and Moore S , *Using Genetic Algorithms for complex real-time scheduling applications*, Network Operations and Management Symposium, NOMS 98, IEEE, Vol 1, 1998, pp 245-248.

- Mouaddib A-I and Gallone J-M, *Progressive scheduling for real-time Artificial Intelligence tasks*, Engineering Complex Computer Systems Proceeding, 2nd IEEE International Conference, 1996, pp. 95-98.
- Mulvaney D. J., personal communication, 1996.
- Mulvaney D. J and Sillitoe I. P. W, *An assessment of new techniques for lift scheduling*, report, August 1992, Loughborough University, UK
- Nakai S., Kubo S. and Imasaki N., *Elevator group control system with fuzzy neural network model*, Proceedings of the International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The 2nd International Fuzzy Engineering Symposium, Vol. 5, March 20-24, 1995, pp. 1-106.
- Nilsson N. J., *Artificial Intelligence a new synthesis*, San Francisco: Morgan kaufmann, 1998.
- Ovaska S. J., *Electronics and information technology in high range elevator systems*, Mechatronics Vol. 2, No. 1, 1991, pp. 89-99.
- Peters R. and Mehta P., *Lift passenger traffic patterns. applications, current knowledge and measurement*, Elevcon 96, Barcelona, Spain, October, 1996, pp 174-183.
- Rich E. and Knight K., *Artificial Intelligence*, New York: McGraw-Hill, 2nd edition, 1991.
- Russell S. and Norvig P., *Artificial Intelligence a modern approach*, New Jersey, Prentice Hall, 1995.
- Sattar S., Mizon J. and Sirag D., *Using Fuzzy Logic to determine the number of passengers entering and exiting an elevator car*, U.S. Pat. no. 5 219 042, June 15, 1993, Otis Elevator Company.
- Siikonen M-L., *Planning and control models for elevators in high rise buildings*, PhD thesis, October 1997, Helsinki University of Technology, Finland.

- Sıikonen M-L. and Leppala J , *Elevator pattern recognıtıon*, Proceedıng of the Fourth World Congress of the International Fuzzy Systems Association, Brussels, Belgium, 1991, pp. 195-198.
- Sıngh S., Norvıg P. and Cohn D , *Agents and Reinforcement Learnıng*, Dr. Dobb's Journal, March, 1997, pp 28-32
- Sırag D , *Usıng Fuzzy Logıc to determıne the number of passengers ın an elevator car*, U.S. Pat no 5 260 527, November 9, 1993, Otis Elevator Company.
- So A. T P., Kuok H. S., Liu S. K., Chan W. S. and Chow T. Y., New developments in elevator traffic analysis, ELEVATORI, November-December, 1994, pp. 59-74.
- Stankovic J , Spurı M., M. Natal D and Buttazzo G , *Implıcatıons of classıcal schedulıng results for real tıme systems*, Scuola Superiore, S Anna, via Carducci, 40-56100 Pisa, Italy, June 23, 1994
- Strakosch G. R., *Vertıcal transportatıon· elevators and escalators*, New York John Wıley & Sons, 2nd edition, 1983.
- Sutton R. and Barto A., *Reınforcement Learnıng. an ıntroductıon*, Cambridge, MA: MIT Press, A Bradford Book,1998.
- Thangavelu K., *Artıfıcıal ıntellıgence, based learning system predıctıng "peak-perıod" tımes for elevator dıspatchıng*, U.S. Pat. no. 5 241 142, August 31, 1993, Otis Elevator Company.
- Thangavelu K., *Up-peak elevator channellıng system wıth optımısed preferentıal servıce to hıgh ıntensıty traffic floor*, U.S. Pat. no. 5 183 981, February 2, 1993, Otis Elevator Company.
- Thangavelu K and Pullela VS, *Learnıng methodology for ımprovıng traffic predıctıon accuracy of elevator systems usıng AI*, US Pat. no. 5 168 136, December, 1992, Otıs Elevator Company, Farmıngton, Conn., USA.

- Thangavelu K , *Relative system response elevator dispatching system using Artificial Intelligence to vary bonuses and penalties*, U.S. pat. No. 5 024 295, June 18, 1991, Otis Elevator Company, Farmington, Conn., USA.
- Thangavelu K., *Queue based elevator dispatching system using up peak period traffic prediction*, U.S. pat. No. 4 838 384, June 13, 1989, Otis Elevator Company, Farmington, Conn., USA.
- Tobita T., Fujino A , Inaba H , Yoneda K. and Ueshima T., *An elevator characterised group supervisory control system*, IECON 91, Kobe, Japan, vol 3, 1991, pp. 1972-1976
- Tobita T., Fujino A , Inaba H., Segawa K., Yoneda K. and Ichikawa Y., *A parameter tuning using Genetic Algorithms for an elevator group control system*, IECON 96, 1996, pp 823-828
- Uckun S., Baachi S. and Kawamura K., *Managing Genetic Search in job shop scheduling*, IEEE Expert, October 1993, 15-24.
- Ujihara H and Amano M , *Latest elevator group control system*, Mitsubishi Electric Advance, June, 1994, pp. 10-12.

Appendix A

The Kodak building lift system specifications

Number of lifts	6	
Capacity	16 passengers	
speed	$3.5ms^{-1}$	
Jerk rate	$2ms^{-3}$	
Acceleration/deceleration rate	$1ms^{-2}$	
Number of floors	18	
Floor heights	**Floors**	**Heights**
	Terminal floor-1	4.6m
	1-4	3 5m
	4-16	3 35m
	16-17	3 5m
Door closing time	2.7s	
Door opening time	2 2s	
Door dwell when no activity	8s	
Door dwell after passenger exit	1s	
Door dwell after passenger entry	3 5s	

Appendix B

Examples of the simulation of lift trips between floors

The following is a sample of the calculation results of the trips between floors sampled at a time step of 0.05 seconds, where:

t1	**Jerk interval (s)**	**d2**	**Distance travelled by lift during t2.(m)**
t2	**Acceleration interval (s)**	**d3**	**Distance travelled by lift during t3 (m)**
t3	**Jerk interval (transition to full speed) (s)**	**v2**	**Lift velocity during t2 (ms-1)**
t4	**Full speed interval (s)**	**v3**	**Lift velocity during t3 (ms-1)**

From floor	To floor	Trip distance	Trip time	t2	d2	v2	d3	v3
0	1	4 600	4 819	1 409	1 345	1 659	0 913	1 909

Full speed time = 0 000 Time until deceleration = 1 909

Car position floor	Bypass floor time
1	4 819

From floor	To floor	Trip distance	Trip time	t2	d2	v2	d3	v3
0	2	8 100	6 214	2 107	2 746	2 357	1 262	2 607

Full speed time = t4 = 0 000 Time until deceleration = 2 607s

Car position floor	Bypass floor time
1	3 350
2	6 214

From floor	To floor	Trip distance	Trip time	t2	d2	v2	d3	v3
0	3	11 600	7 330	2 665	4 217	2 915	1 541	3 165

Full speed time=0 000 Time until deceleration=3 165

Car position floor	Bypass floor time
1	3 300
2	4 450
3	7 330

From floor	To floor	Trip distance	Trip time	t2	d2	v2	d3	v3
0	4	15 100	8 314	3 000	5 250	3 250	1 708	3 500

Full speed time=0 314 Time until deceleration=4 314

Car position floor	Bypass floor time
1	3 300
2	4 350
3	5 450
4	8 314

From floor	To floor	Trip distance	Trip time	t2	d2	v2	d3	v3
0	5	18 450	9 271	3 000	5 250	3 250	1 708	3 500

Full speed time=1 271 Time until deceleration=5 271

Car position floor	Bypass floor time
1	3 300
2	4 350
3	5 350
4	6 450
5	9 271

From floor	To floor	Trip distance	Trip time	t2	d2	v2	d3	v3
0	6	21 800	10 229	3 000	5 250	3 250	1 708	3 500

Full speed time=2 229 Time until deceleration=6 229

Car position floor	Bypass floor time
1	3 300
2	4 350
3	5 350
4	6 350
5	7 400
6	10 229

From floor	To floor	Trip distance	Trip time	t2	d2	v2	d3	v3
0	7	25 150	11 186	3 000	5 250	3 25	1 708	3 5

Full speed time=3 186 Time until deceleration=7 186

Car position floor	Bypass floor time
1	3 300
2	4 350
3	5 350
4	6 350
5	7 300
6	8 400
7	11 186

From floor	To floor	Trip distance	Trip time	t2	d2	v2	d3	v3
0	8	28 500	12 143	3 000	5 250	3 25	1 708	3 5

Full speed time=4 143 Time until deceleration=8 143

Car position floor	Bypass floor time
1	3 300
2	4 350
3	5 350
4	6 350
5	7 300
6	8 250
7	9 350
8	12 143

From floor	To floor	Trip distance	Trip time	t2	d2	v2	d3	v3
0	9	31 85	13 100	3 000	5 250	3 25	1 708	3 5

Full speed time=5 100 Time until deceleration=9 100

Car position floor	Bypass floor time
1	3 300
2	4 350
3	5 350
4	6 350
5	7 300
6	8 250
7	9 200
8	10 300
9	13 100

From floor	To floor	Trip distance	Trip time	t2	d2	v2	d3	v3
0	10	35 200	14 057	3 000	5 250	3 25	1 708	3 5

Full speed time=6 057 Time until deceleration=10 057

Car position floor	Bypass floor time
1	3 300
2	4 350
3	5 350
4	6 350
5	7 300

6	8 250
7	9 200
8	10 150
9	11 250
10	14 057

From floor	To floor	Trip distance	Trip time	t2	d2	v2	d3	v3
0	11	38 550	15 014	3 000	5 250	3 25	1 708	3 5

Full speed time=7 014 Time until deceleration=11 014

Car position floor	Bypass floor time
1	3 300
2	4 350
3	5 350
4	6 350
5	7 300
6	8 250
7	9 200
8	10 150
9	11 150
10	12 200
11	15 014

From floor	To floor	Trip distance	Trip time	t2	d2	v2	d3	v3
0	12	41 900	15 971	3 000	5 250	3 25	1 708	3 5

Full speed time=7 971 Time until deceleration=11 971

Car position floor	Bypass floor time
1	3 300
2	4 350
3	5 350
4	6 350
5	7 300
6	8 250
7	9 200
8	10 150
9	11 150

10	12 100
11	13 150
12	15 971

From floor	To floor	Trip distance	Trip time	t2	d2	v2	d3	v3
0	13	45 250	16 929	3 000	5 250	3 25	1 708	3 5

Full speed time=8 929 Time until deceleration=12 929

Car position floor	Bypass floor time
1	3 300
2	4 350
3	5 350
4	6 350
5	7 300
6	8 250
7	9 200
8	10 150
9	11 150
10	12 100
11	13 050
12	14 100
13	16 929

From floor	To floor	Trip distance	Trip time	t2	d2	v2	d3	v3
0	14	48 600	17 886	3 000	5 250	3 250	1 780	3 500

Full speed time=9 886 Time until deceleration=13 886

Car position floor	Bypass floor time
1	3 300
2	4 350
3	5 350
4	6 350
5	7 300
6	8 250
7	9 200
8	10 150

9	11 150
10	12 100
11	13 050
12	14 000
13	15 100
14	17 886

From floor	To floor	Trip distance	Trip time	t2	d2	v2	d3	v3
0	15	51 950	18 843	3 000	5 250	3 250	1 708	3 500

Full speed time=10 843 Time until deceleration=14 843

Car position floor	Bypass floor time
1	3 300
2	4 350
3	5 350
4	6 350
5	7 300
6	8 250
7	9 200
8	10 150
9	11 150
10	12 100
11	13 050
12	14 000
13	14 950
14	16 050
15	18 843

From floor	To floor	Trip distance	Trip time	t2	d2	v2	d3	v3
0	16	55 300	19 800	3 000	5 250	3 250	1 708	3 500

Full speed time=11 800 Time until deceleration=15 800

Car position floor	Bypass floor time
1	3 300
2	4 350
3	5 350
4	6 350

5	7 300
6	8 250
7	9 200
8	10 150
9	11 150
10	12 100
11	13 050
12	14 000
13	14 950
14	15 900
15	17 000
16	19 800

From floor	To floor	Trip distance	Trip time	t2	d2	v2	d3	v3
0	17	58 800	20 800	3 00	5 25	3 25	1 708	3 5

Full speed time=12 800 Time until deceleration=16 800

Car position floor	Bypass floor time
1	3 300
2	4 350
3	5 350
4	6 350
5	7 300
6	8 250
7	9 200
8	10 150
9	11 150
10	12 100
11	13 050
12	14 000
13	14 950
14	15 900
15	16 850
16	17 950
17	20 800

Appendix C

Simulated ETQ scheduling time log

Number of scheduling times of 0s = 172422

Number of scheduling times of 50s = 7532

Number of scheduling times of 60s = 7246

Number of scheduling times of 110s = 0

Number of scheduling times of >110s and <= 250s = 0

Number of scheduling times of >250s = 0

Appendix D

Examples of lift system operation rules

The Express Evans Ltd

Group control of multi-car lift systems

1 A landing call appearing in the path of a lift travelling to an assignment should not necessarily be answered by that lift. That is giving preference to the most recent call at the expense of older ones tends to defeat the aim of 'equality'. On the other hand, deliberately ignoring such calls can produce an overall longer average wait. The decision should depend upon other factors such as how soon another lift could deal with the recent call.

2 Car load actual and anticipated, should be included in assignment decisions, especially if the target landings call has been waiting for a considerable time, that is the lift system is busy. Load measurement is not good at present but will improve.

3 Distributed parking should move lifts to strategic positions according to the traffic situation. This is best done by dividing the building into bands and parking a lift in each band (one being the main entrance). The floor selected in each band will be the busiest for the time of the day unless it is already parked in another floor within that band, then the lift should not be moved. A lift leaving a parked area will not cause other lifts to reposition.

4 Car calls are only accepted for the direction of the landing call answered. If both landing calls are registered, the lift will answer one of them and if there wasn't any car calls in that direction, the lift waits for 8 seconds and then accept the other car call

5 If the assignment of a lift moving to answer a landing call is cancelled then lift would stop with door closed and if the lift is reassigned to go in the reverse direction then the lift would stop and reverse direction

Appendix E

Prioritised A* scheduler test results - simplified

Results of some of the test examples for the prioritised A* scheduler using simplified assumption as in Section 7 3

The details of the following examples are saved in the attached floppy disk

Example 1

- Lifts are moving
- Both lifts can stop at the 7th floor
- Schedule is in eg1 doc **scheduling time=0.8ms**, nodes=14, sortedOpen_count=4, tree_level=4 and Backtracks=0.
- Eg1h0 doc and Detailed-eg1h0 doc have the scheduling run when h=0

scheduling time=14ms, nodes=163, sortedOpen_count=60, tree_level=4 and backtracks=47

- Eg1d doc and Detailed-eg1d doc have the scheduling run when only direct lift path is considered when calculating h. **scheduling time=7.7ms**, nodes = 107, sortedOpen_count=38, tree_level=4 and backtracks=32.
- ***All above schedules produce the same schedule with the same value of f***
- Eg1wt doc has scheduling results when waiting time is considered. nodes=14, sortedOpen_count=4, tree_level=4 and backtracks=0

Lift	Assignment	Lift total cost at this level
0	0	10
1	3	2
0	1	15
1	2	7

	8		□
	7	*	
LC3 ▼	6		
LC2 ▼	5	□	
LC1 ▲	4		
LC0 ▲	3		
	2		*
	1		
	G		
		▲	▼

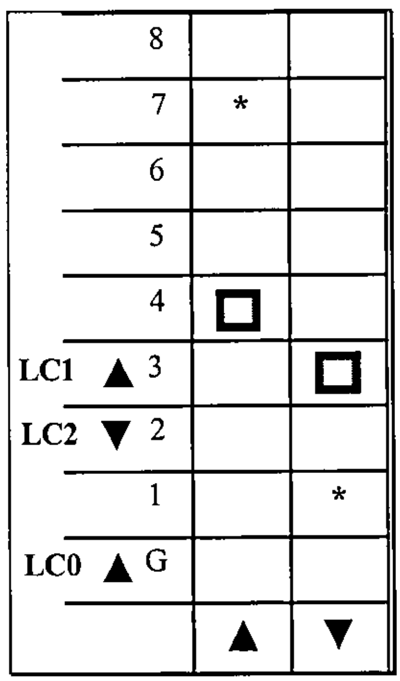
Example 2

- Lifts are moving
- Lift0 can stop at the 5th floor and lift1 can stop at the 2nd floor
- Schedule is in Eg2.doc. **scheduling time=0.6ms**, nodes=10, sortedOpen_count=4, tree_level=4, backtracks=0
- Eg2h0 doc and Detailed-eg2h0 doc have the scheduling run when h=0.

scheduling time=2ms, nodes=49, sortedOpen_count=19, tree_level=4 and backtracks=15

- Eg2d doc and Detailed-eg2d doc have the scheduling run when only direct lift path is considered when calculating h **scheduling time=1ms**, nodes = 24, sortedOpen_count=11, tree_level=4, backtracks=7
- ***All above schedules produce the same schedule with the same value of f***
- Egwt2 doc has the result when waiting time is considered nodes = 10, sortedOpen_count=4, tree_level=4, backtracks=0

Lift	Assignment	Lift total cost at this level
0	1	11
1	2	1
0	-1	0
1	0	11



Appendix F

Prioritised A* scheduler test results - detailed

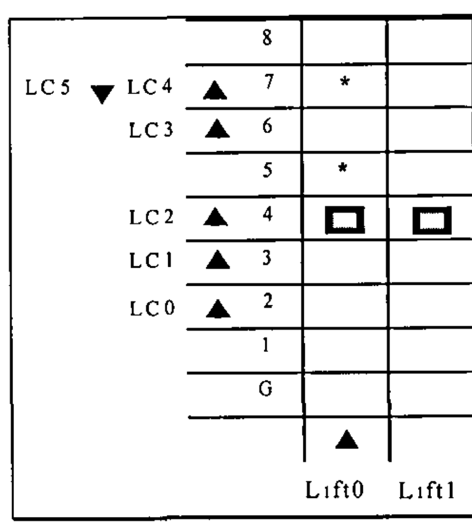
Results of some of the test examples for the prioritised A* scheduler using detailed assumption defined in Section 7 4

The details of the following examples are saved in the attached floppy disk

Example 3

- Lifts are not moving
- Lift 0 stopped to answer a landing call.
- Lift 1 stopped to answer a car call
- Schedule is in eg3.doc **scheduling time=1.8ms**, nodes=27, sortedOpen_count=6, tree_level=6 and backtracks=0

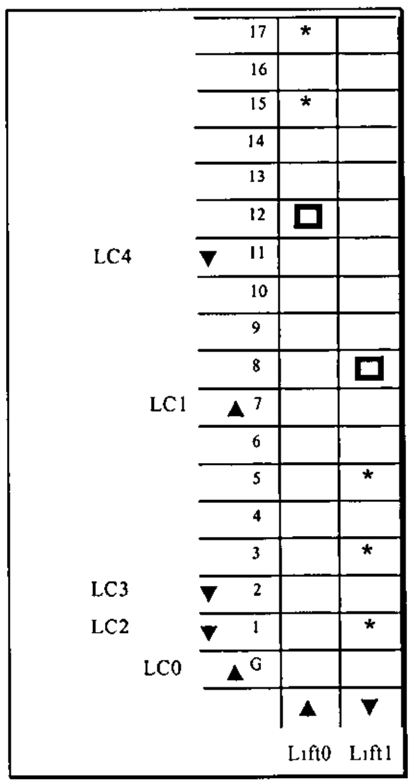
Lift	Assignment	lift total cost at this level
0	2	0
1	0	4 7
0	3	17 3
1	1	14 6
0	4	27 2
1	5	35 9



Example 4

- Lifts are moving
- Destination and next possible stop for lift 0 is floor 15 and lift 1 is the 5th floor
- Schedule is in eg4 doc **scheduling time=1.5ms**, nodes=21, sortedOpen_count=6, tree_level=6, total_LC=5 and backtracks=0
- ***Observe in Eg4.doc that f reduces within the search tree.***
- Eg4h0 doc has the scheduling run when h=0. **scheduling time=464ms**, nodes=1000, sortedOpen_count=340, tree_level=6 and backtracks=309
- Eg4d doc has the scheduling run when only direct lift path is considered when calculating h **scheduling time=84ms,** nodes=454, sortedOpen_count=140, tree_level=6 and backtracks=143.
- ***All above schedules produce the same schedule with the same value of f***

Lift	Assignment	Lift total cost at this level
0	4	23 8
1	2	19 8
0	3	41 7
1	0	30 2
0	-1	0
1	1	46 1



Example 5

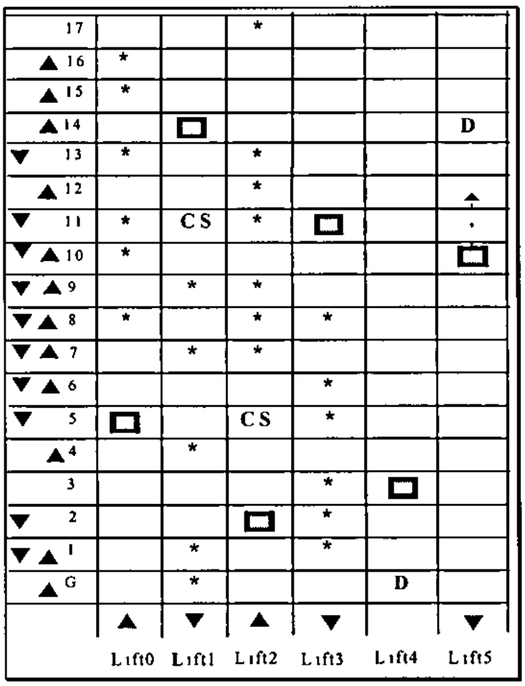
- 4 lifts are moving, lift 0 and lift 4 decelerating to answer a car call
- 24 landing calls and 25 car calls
- Lifts possible stops starting with lift0 are floors 8,11,5 and 8.
- scheduling results in Eg5 doc. **scheduling time=173ms,** nodes=331, sortedOpen_count=27, tree_level=28 and bactracks=0
- ***Observe that f have reduced within the search tree***

17			*			
▼▲ 16	*					
▼▲ 15	*					
▲ 14		□				
▼ 13	*		*			
▲ 12			*			
▼ 11	*		*	□		
▼▲ 10	*	D				
▼▲ 9		*	*			
▼▲ 8	*		*	*		
▼▲ 7		*	*			
▼▲ 6				*		
▼ 5	□			*		
▲ 4		*				
3		*		*		
▼ 2			□	*		
▼▲ 1		*		*		
▲ G		*				
	▲	▼	▲	▼		
	Lift	Lift	Lift	Lift		

Lift	Assignment	Lift total cost at this level
0	7	11.4
1	20	3
2	3	4
3	17	3
0	10	38 6
1	19	12 9
2	4	13.9
3	15	14 4
0	11	49
1	18	22 8
2	5	24.3
3	14	24 8
0	23	66.8
1	16	34 2
2	6	34.7
3	13	43 6
0	22	77.2
1	2	86 6
2	8	53.5
3	12	54
0	21	88 1
1	-1	0
2	9	71.3
3	0	64.4
0	-1	0
1	-1	0
2	-1	0
3	1	74 3

Example 6

- 24 landing calls
- Lift 0 decelerating to stop for a car call at floor 8
- Lift 1 stopping for a car call at the 9th floor It can stop at the 11th floor
- Lift 2 can stop at the 5th floor with destination at the 7th floor
- Lift 3 decelerating to answer a car call at the 8th floor
- Lift 4 decelerating to answer a car call at the G floor with next direction STOP
- Lift 5 decelerating to answer a landing call DOWN at floor 14 with next direction DOWN
- Lifts can stop at the following floors starting with lift0 :8,11,5,8,0 and 14
- Scheduling results in Eg6 doc. **scheduling time 207ms,** nodes=333, sortedOpen_count=29, tree_level=29 and backtracks=0



Lift	Assignment	Lift total cost at this level
0	7	11 4
1	20	3
2	3	4
3	17	3
4	0	3
5	21	13 9
0	8	29 2
1	19	12 9
2	4	13 9
3	15	14 4
4	1	13 4
5	12	34 8
0	10	47 5
1	18	22 8
2	5	24 3
3	14	24 8
4	2	25 3
5	22	66 1
0	11	57 9
1	16	34 2
2	6	34 7
3	13	43 6
4	9	44 2
5	-1	0
0	-1	0
1	-1	0
2	-1	0
3	-1	0
4	23	63.5

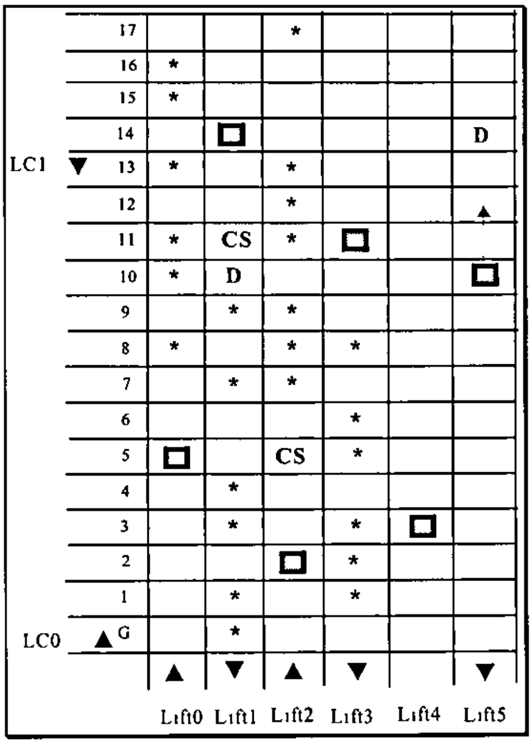
Example 7

- 6 lifts and 13 landing calls
- Lift 0 decelerating to stop for a car call at floor 8
- Lift 1 stopping at the 10th floor. It can stop at the 11th floor
- Lift 2 can stop at the 5th floor with destination at the 7th floor.
- Lift 3 decelerating to answer a car call at the 8th floor
- Lift 4 decelerating to answer a car call at the G floor with next direction STOP
- Lift 5 decelerating to answer a landing call DOWN at floor 14 with next direction DOWN.
- Scheduling results in Eg7 doc. **scheduling time=29ms,** nodes=126, sortedOpen_count=18, tree_level=29 and backtracks=0

▼ 17			*			
▲ 16	*					
▲ 15	*					
14		□				D
▼ 13	*		*			
▲ 12			*			▲
11	*	CS	*	□		⋮
▼ 10	*	D				□
9		*	*			
▲ 8	*		*	*		
7		*	*			
▲ 6				*		
5	□		CS	*		
▼ 4		*				
3		*		*	□	
▲ 2			□	*		
▼▲ 1		*		*		
▲ G		*				
	▲	▼	▲	▼		▼
	Lift0	Lift1	Lift2	Lift3	Lift4	Lift5

Lift	Assignment	Lift total cost at this level
0	4	3
1	10	4
2	3	4
3	9	26 2
4	0	3
5	11	11 4
0	6	38 1
1	-1	0
2	5	44 5
3	-1	0
4	1	12 9
5	8	32 3
0	-1	0
1	-1	0
2	-1	0
3	-1	0
4	2	22 8
5	-1	0
0	-1	0
1	-1	0
2	-1	0
3	-1	0
4	7	45 7
5	-1	0
0	-1	0
1	-1	0
2	-1	0
3	-1	0
4	12	55 6

Example 8

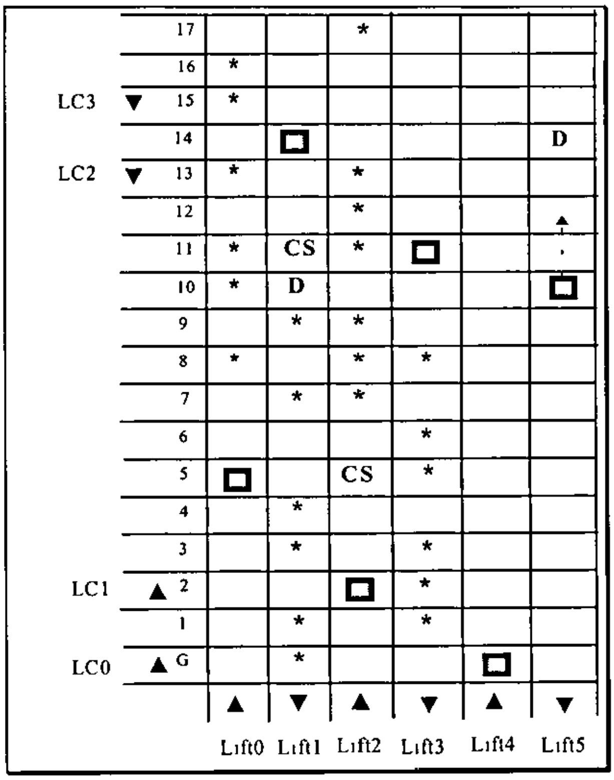


- 6 lifts and 2 landing calls
- Lift 0 decelerating to stop for a car call at floor 8.
- Lift 1 stopping at the 10th floor. It can stop at the 11th floor
- Lift 2 can stop at the 5th floor with destination at the 7th floor
- Lift 3 decelerating to answer a car call at the 8th floor.
- Lift 4 decelerating to answer a car call at the G floor with next direction STOP
- Lift 5 decelerating to answer a landing call DOWN at floor 14 with next direction DOWN
- Scheduling results in Eg8 doc **scheduling time=1msec,** node=17, sortedOpen_count=6, tree_level=6 and backtracks=0
- Eg8h0 doc has the scheduling run when h=0 **Scheduling time=1ms,** nodes=21, sortedOpen_count=8, tree_level=6 and backtracks=3.
- Eg8d doc has the scheduling run when only direct lift path is considered when calculating h **scheduling time=1ms,** node=17, sortedOpen_count=6 and backtracks=0.

Lift	Assignment	Lift total cost at this level
0	-1	0
1	-1	0
2	-1	0
3	-1	0
4	0	3
5	1	13 9

Example 9

- 6 lifts and 4 landing calls
- Lift 1 stopping at the 10th floor. It can stop at the 11th floor
- Lift 2 can stop at the 5th floor with destination at the 7th floor
- Lift 3 decelerating to answer a car call at the 8th floor
- Lift 4 stopped to answer a car call at the G floor with next direction UP Only 2.7sec delay is considered when calculating the direct path cost table However, passenger delay is added to the path costs of after answering LC0 in the Lift4 cost of a trip between landing calls table. In the future the stopping for a car call flag should change by the simulator to indicate whether the passenger has departed or not and the right delay can be chosen

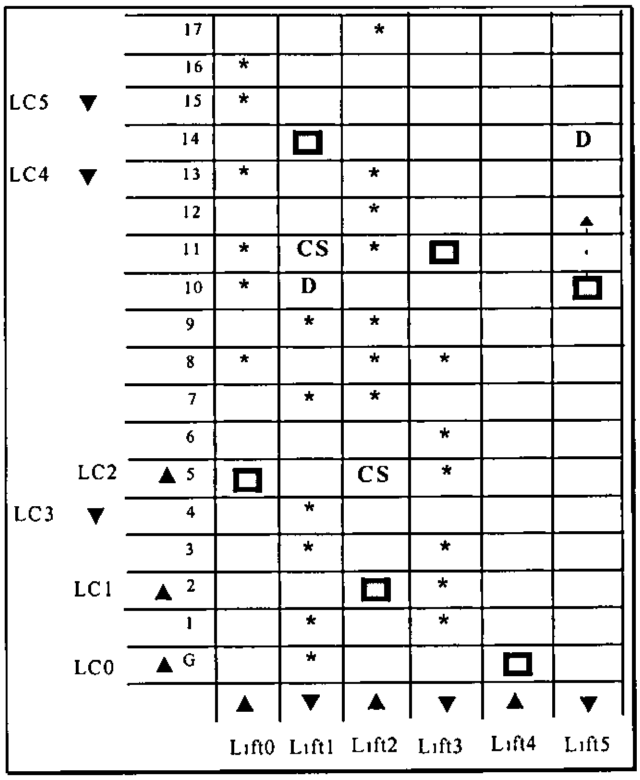


- Lift 5 decelerating to answer a landing call DOWN at floor 14 with next direction DOWN
- Scheduling results in Eg9 doc **scheduling time=3ms**, nodes=35, sortedOpen_count=9, tree_level=17 and backtracks=0
- Eg9h0 doc has the scheduling run when h=0 **scheduling time=140ms**, nodes=678, sortedOpen_count=220, tree_level=17, and backtracks=96
- Eg9d doc has the scheduling run when only direct Lift path is considered when calculating h **scheduling time=8ms**, nodes = 110, sortedOpen_count=32, tree_level=17 and backtracks=25

Lift	Assignment	Lift total cost at this level
0	-1	0
1	-1	0
2	-1	0
3	-1	0
4	0	0
5	2	13 9
0	-1	0
1	-1	0
2	-1	0
3	-1	0
4	1	11 4
5	-1	0
0	-1	0
1	-1	0
2	-1	0
3	-1	0
4	3	43 7

Example 10

- 6 Lifts and 6 landing calls
- Lift 1 stopping at the 10th floor. It can stop at the 11th floor
- Lift 2 can stop at the 5th floor with destination at the 7th floor
- Lift 3 decelerating to answer a car call at the 8th floor.
- Lift 4 stopped to answer a car call at the G floor with next direction UP
- Lift 5 decelerating to answer a landing call DOWN at floor 14 with next direction DOWN



- Scheduling results in Eg10 doc
  scheduling time=5ms, nodes=46, sortedOpen_count=11, tree_level=17 and backtracks=0
- Eg10h0 doc has the *partial* scheduling result when h=0 **scheduling time=459ms**, nodes=1891, sortedOpen_count=400, tree_level=5 and Backtracks=135
- Eg10d doc has the *partial* scheduling result when only direct Lift path is considered when calculating h **scheduling time=501ms**, node=1584, sortedOpen_count=400, tree_level=10, Backtracks=217

Lift	Assignment	Lift total cost at this level
0	-1	0
1	3	22 8
2	2	3
3	-1	0
4	0	0
5	4	13 9
0	-1	0
1	-1	0
2	-1	0
3	-1	0
4	1	11 4
5	-1	0
0	-1	0
1	-1	0
2	-1	0
3	-1	0
4	5	43 7

Appendix G

Simulation results of ETQ and prioritised A* schedulers

Results of the intelligent real-time lift scheduling system as explained in Section 8.2.

Time of day (hours)	Actual PASS0	Actual LC0	Actual PASS2	Actual LC2
7 - 8	139	60	18	12
8 - 9	376	79	59	43
9 - 10	148	78	95	58
10 - 11	58	37	125	86
11 - 12	46	36	104	62
12 - 13	131	59	166	89
13 - 14	231	94	226	127
14 - 15	111	57	119	82
15 - 16	50	35	144	98
16 - 17	66	40	105	89
17 - 18	33	16	58	44
18 - 19	24	12	12	12
19 - 20	25	17	11	12
TOTAL	**1438**	**620**	**1242**	**814**

The actual number of passengers and landing calls for both the terminal and second floor in the Kodak building on 3/10/94 between 7am and 8pm.

Time of day	ETQ							
(hours)	PASS0	LC0	LC0	LC0	PASS2	LC2	LC2	LC2
7 - 8	136	72	68	75	23	21	19	21
8 - 9	382	90	100	85	61	51	53	49
9 - 10	151	60	53	68	81	56	55	54
10 - 11	40	29	27	25	126	75	75	72
11 - 12	51	34	33	33	90	60	61	55
12 - 13	140	65	66	68	148	84	76	87
13 - 14	233	81	84	89	209	111	106	111
14 - 15	100	49	51	49	137	84	83	93
15 - 16	52	26	26	26	151	90	95	95
16 - 17	50	24	29	31	115	74	73	73
17 - 18	27	14	15	12	43	36	36	37
18 - 19	22	12	13	13	16	16	16	16
19 - 20	35	18	23	21	14	14	14	14
TOTAL	**1419**	**574**	**588**	**595**	**1214**	**772**	**762**	**777**

Landing call frequencies for both the terminal and second floor for three simulation runs using fixed passengers arrival rate in the Kodak building on 3/10/94 between 7am and 8pm.

☞ *Key: PASSn = number of passengers at floor n, LCn = number of landing calls at floor n*

Time of day (hours)	A* simulations Number of landing calls for the terminal floor FULL SEARCH		DD	DSWT	DSWTPCC	DD WT PCC UP PEAK
7 - 8	71	76	73	79	71	63
8 - 9	111	121	121	126	122	99
9 - 10	71	68	68	76	79	60
10 - 11	30	21	27	30	26	28
11 - 12	32	32	37	30	34	37
12 - 13	59	56	58	58	62	59
13 - 14	95	90	96	95	89	79
14 - 15	53	52	47	55	50	50
15 - 16	37	36	31	32	28	32
16 - 17	24	25	27	23	24	25
17 - 18	15	16	13	15	16	12
18 - 19	11	13	13	12	12	14
19 - 20	21	20	19	20	16	20
TOTAL	**630**	**626**	**630**	**651**	**629**	**578**

Time of day (hours)	A* simulations Number of landing calls for the second floor FULL SEARCH		DD	DSWT	DSWTPCC	DD WT PCC UP PEAK
7 - 8	20	20	19	20	21	19
8 - 9	51	49	51	50	49	52
9 - 10	57	52	55	58	54	53
10 - 11	72	75	74	74	68	70
11 - 12	67	67	62	65	60	59
12 - 13	90	82	86	86	87	83
13 - 14	117	118	108	111	115	121
14 - 15	84	86	87	91	88	87
15 - 16	98	103	93	91	93	99
16 - 17	79	76	80	75	78	77
17 - 18	32	33	34	34	36	33
18 - 19	16	16	16	16	16	16
19 - 20	14	14	14	14	14	14
TOTAL	**797**	**791**	**779**	**785**	**779**	**783**

Landing call frequencies for both the terminal and second floor. Simulation runs conducted using fixed passengers arrival rate in the Kodak building on 3/10/94 between 7am and 8pm.

Appendix H

Comparison between queue frequencies

Queue size	ETQ			A*	A* truncated				
passengers	1	2	3	FULL SEARCH	1	2 DD	3 DD WT	4 DD WT PCC	5 DD WT PCC UP PEAK
1	2083	2110	2144	2296	2264	2268	2256	2270	2269
2	337	324	338	259	259	264	272	269	268
3	82	75	79	44	53	50	44	42	47
4	23	28	29	7	15	13	13	11	12
more than 4	21	16	12	10	10	8	8	10	5

☞ *Key: DD = Door Delay, WT = Waiting Time, PCC = Predicted Car Calls*

Comparison between queue frequencies for all floors between 7am and 8pm on 03/10/94. The queues are for nine simulation runs, three simulation runs for ETQ, one simulation run with full A* search and 5 simulation runs using the truncated A* scheduler.

Appendix I

Popular floors in the Kodak building

Popular floors selected for each 5 minutes between 7:05am and 7:55pm on 3/10/94 and listed according to their priority. Priority1 is the most popular zone

Time	Priority 1	Priority 2	Priority 3	Priority 4	Priority 5	priority 6
07 05	0	2	4	7	10	14
07 10	0	6	10	15		
07 15	0	2	9	14		
07 20	0	4	14			
07 25	0	17	14			
07 30	0	17	6	7		
07·35	0	2	4			
07.40	0	2	8			
07 45	0	2	4	7	10	14
07 50	0	2	16	8	9	
07·55	0	5	6	2	10	12
08 00	0	2	8			
08 05	0	2	8	11	1	5
08·10	0	2	4	9	12	14
08 15	0	2	4	6	9	12
08 20	0	2	9	1	5	7
08.25	0	2	10	14	16	1
08·30	0	2	15	1	10	11
08 35	0	2	16	5	7	9
08 40	0	10	2	9	3	4
08 45	0	2	8	1	11	9
08·50	0	2	11	10	5	9
08 55	0	2	4	13	5	6
09 00	0	2	10	14	1	6
09 05	0	6	14	13	15	16
09 10	0	2	1	4	5	7
09 15	0	3	17	2	4	8
09·20	0	2	1	15	9	4
09 25	0	2	3	9	5	12
09 30	0	2	16	4	9	14
09 35	0	10	1	2	9	6
09 40	0	2	17	1	15	3
09·45	0	2	10	4	7	12
09.50	0	5	1	2	3	9
09 55	0	15	2	11	8	9
10 00	2	0	7	10	15	4

Time	Priority 1	Priority 2	Priority 3	Priority 4	Priority 5	priority 6
10 05	2	4	11	0	3	12
10 15	2	0	10	13	3	5
10 20	2	0	15	11	5	14
10:25	10	0	2	16	4	5
10.30	2	0	11	16	15	6
10:35	2	10	16	11	12	1
10:40	2	0	15	12	14	4
10·45	2	5	6	7	11	16
10 50	2	14	0	13	1	11
10.55	2	4	6	9	10	13
11.00	0	2	10	15	5	8
11 05	2	0	16	1	6	15
11·10	0	2	13	16	1	7
11:15	2	12	0	4	1	6
11:20	2	15	4	12	0	8
11:25	2	0	13	7	15	9
11·30	2	0	6	14	7	11
11.35	2	0	10	12	15	9
11 40	2	0	6	10	12	15
11:45	2	0	16	7	4	12
11·50	0	5	15	12	2	4
11 55	4	0	2	9	6	1
12 00	0	2	10	11	5	1
12.05	9	15	12	6	2	10
12.10	2	13	11	0	5	6
12·15	2	15	16	10	0	11
12 20	2	0	10	7	9	17
12.25	4	2	0	10	13	5
12 30	2	0	8	6	7	10
12·35	2	10	0	9	11	13
12 40	0	2	12	6	15	14
12.45	2	0	17	13	9	3
12 50	0	2	15	14	13	3
12 55	0	15	2	5	13	14
13 00	0	2	7	3	6	16
13 05	2	0	11	5	9	14
13.10	0	2	9	7	5	10
13 15	0	2	15	11	12	14
13·20	0	2	7	12	9	10

Time	Priority 1	Priority 2	Priority 3	Priority 4	Priority 5	priority 6
13 25	0	2	10	13	14	6
13 35	0	2	9	16	6	11
13 40	2	0	15	5	7	9
13.45	2	0	14	3	4	8
13 50	0	2	9	4	5	6
13 55	0	2	4	5	15	10
14 00	0	2	15	10	13	1
14 05	0	2	16	7	12	4
14·10	0	2	11	4	9	13
14·15	0	2	1	9	4	6
14 20	0	6	11	2	5	8
14 25	2	15	0	4	9	10
14·30	0	2	11	6	15	5
14 35	2	0	11	12	5	6
14 40	2	14	0	17	5	15
14.45	0	2	15	4	10	11
14 50	0	2	14	10	4	9
14 55	0	13	7	15	2	10
15.00	2	4	9	0	6	11
15 05	0	2	9	15	5	6
15 10	2	14	11	12	0	13
15 15	2	0	7	13	4	5
15 20	2	4	10	14	17	0
15.25	2	9	0	4	12	1
15 30	0	2	9	13	14	15
15.35	2	11	0	9	13	16
15.40	2	6	11	12	0	10
15·45	2	0	15	10	12	14
15 50	0	4	3	15	2	5
15 55	2	10	0	14	15	7
16 00	0	2	11	1	10	12
16 05	2	0	4	14	6	16
16 10	2	12	0	4	5	9
16 15	2	10	0	5	11	1
16 20	0	2	13	14	15	8
16 25	9	2	12	13	0	10
16 30	2	0	6	8	9	4
16 35	0	2	5	4	15	1
16 40	13	2	15	11	0	8

Time	Priority 1	Priority 2	Priority 3	Priority 4	Priority 5	priority 6
16·45	2	4	14	0	11	6
16 55	0	15	3	1	5	10
17:00	0	11	10	12	16	2
17 05	0	2	13	5	6	8
17 10	2	6	16	10	0	14
17:15	0	2	10	7	6	4
17·20	2	0	10	15	16	7
17 25	0	1	7	14	4	9
17 30	5	2	9	6	4	0
17 35	15	0	2	8	7	9
17 40	2	10	7	9	8	11
17 45	2	14	3	10	12	16
17 50	7	5	11	13	14	15
17 55	5	9	13	16	2	4
18 00	2	9	13	1	4	5
18 05	0	2	9	11	15	1
18 10	2	10	9	0	14	16
18 15	13	15	2	6	8	0
18 20	9	0	6	8	2	11
18 25	16	17	10	15	0	2
18·30	11	10	15	16	0	2
18 35	0	2	1	5	15	
18 40	2	4	10	13		
18 45	2	15	16	7	8	17
18 50	2	0	12	1	5	6
18 55	0	2	12	4	8	9
19 00	0	1	2	6	11	12
19 05	0	1	13	14	9	15
19 10	7	8	16	17		
19 15	0	13	6	9	14	15
19·20	4	9	2	3	12	0
19 25	5	2	7	1	8	10
19 30	2	0	1	8	4	10
19 35	9	0	3	12	14	
19.40	1	4	12	14	16	
19:45	8	2	1	4	9	
19.50	15	12	16			
19 55	0	2	4	7	10	14

Appendix J

Prioritised A* scheduler average number of nodes and scheduling time log

Average number of nodes and schedules calculation times results as explained in Section 8.4

************ Average no of nodes=0 395000
************ Backtracks every 10000 schedule=0
************ Average no of nodes=1 436100
************ Backtracks every 10000 schedule=0
************ Average no of nodes=4 225100
************ Backtracks every 10000 schedule=0
************ Average no of nodes=4 926600
************ Backtracks every 10000 schedule=0
************ Average no of nodes=6 121000
************ Backtracks every 10000 schedule=0
************ Average no of nodes=5.793900
************ Backtracks every 10000 schedule=0
************ Average no of nodes=5 546700
************ Backtracks every 10000 schedule=0
************ Average no of nodes=7 394100
************ Backtracks every 10000 schedule=0
************ Average no of nodes=10 878000
************ Backtracks every 10000 schedule=0
************ Average no of nodes=5 684100
************ Backtracks every 10000 schedule=0
************ Average no of nodes=6 137300
************ Backtracks every 10000 schedule=0
************ Average no of nodes=6 649200
************ Backtracks every 10000 schedule=0
************ Average no of nodes=5 327300
************ Backtracks every 10000 schedule=0
************ Average no of nodes=3 228500
************ Backtracks every 10000 schedule=0
************ Average no of nodes=6 641300
************ Backtracks every 10000 schedule=0
************ Average no of nodes=3 784600
************ Backtracks every 10000 schedule=0
************ Average no of nodes=1 958100
************ Backtracks every 10000 schedule=0
************ Average no of nodes=1 040800
************ Backtracks every 10000 schedule=0

Number of scheduling times of 0s 182555
Number of scheduling times of 50s 2375
Number of scheduling times of 60s 2270
Number of scheduling times of 110s 0
Number of scheduling times of >110s and <= 250s 0
Number of scheduling times of >250s 0

Average no of nodes=0 205514
Max no of nodes=53
Max no of tree levels=12

Truncated A* search

************ Average no of nodes=0 356400
************ Backtracks every 10000 schedule=0
************ Average no of nodes=1 656300
************ Backtracks every 10000 schedule=0
************ Average no of nodes=4 504200
************ Backtracks every 10000 schedule=0
************ Average no of nodes=4 759200
************ Backtracks every 10000 schedule=0
************ Average no of nodes=5 548900
************ Backtracks every 10000 schedule=0
************ Average no of nodes=5 915500
************ Backtracks every 10000 schedule=0
************ Average no of nodes=5 891000
************ Backtracks every 10000 schedule=0
************ Average no of nodes=7 888300
************ Backtracks every 10000 schedule=0
************ Average no of nodes=10 028100
************ Backtracks every 10000 schedule=0
************ Average no of nodes=6 180500
************ Backtracks every 10000 schedule=0
************ Average no of nodes=6 659000
************ Backtracks every 10000 schedule=0
************ Average no of nodes=7.150500
************ Backtracks every 10000 schedule=0
************ Average no of nodes=4 956400
************ Backtracks every 10000 schedule=0
************ Average no of nodes=3 257800
************ Backtracks every 10000 schedule=0
************ Average no of nodes=6 199400
************ Backtracks every 10000 schedule=0
************ Average no of nodes=4 413200
************ Backtracks every 10000 schedule=0
************ Average no of nodes=1 879800
************ Backtracks every 10000 schedule=0
************ Average no of nodes=0 923000
************ Backtracks every 10000 schedule=0

Number of scheduling times of 0s 182466
Number of scheduling times of 50s 2455
Number of scheduling times of 60s 2279
Number of scheduling times of 110s 0
Number of scheduling times of >110s and <= 250s 0
Number of scheduling times of >250s 0

Average no of nodes=0 167224
Max no of nodes=51
Max no of tree levels=14

Full A* search

Prioritised A* scheduler log file for full simulation run from 7am until 8pm on 3/10/94. Truncated search is interrupted when number of assigned landing calls is equal to number of available lifts

```
************ Average no of nodes=0 277300
************ Backtracks every 10000 schedule=0
************ Average no of nodes=2 382800
************ Backtracks every 10000 schedule=0
************ Average no of nodes=6 209900
************ Backtracks every 10000 schedule=0
************ Average no of nodes=6 435900
************ Backtracks every 10000 schedule=0
************ Average no of nodes=6 873900
************ Backtracks every 10000 schedule=0
************ Average no of nodes=7 478800
************ Backtracks every 10000 schedule=0
************ Average no of nodes=6 775800
************ Backtracks every 10000 schedule=0
************ Average no of nodes=7 896600
************ Backtracks every 10000 schedule=0
************ Average no of nodes=12 352000
************ Backtracks every 10000 schedule=0
************ Average no of nodes=8 089100
************ Backtracks every 10000 schedule=0
************ Average no of nodes=8 843000
************ Backtracks every 10000 schedule=0
************ Average no of nodes=7 717400
************ Backtracks every 10000 schedule=0
************ Average no of nodes=4 853700
************ Backtracks every 10000 schedule=0
************ Average no of nodes=3 195700
************ Backtracks every 10000 schedule=0
************ Average no of nodes=7 272600
************ Backtracks every 10000 schedule=0
************ Average no of nodes=3 778000
************ Backtracks every 10000 schedule=0
************ Average no of nodes=1.564500
************ Backtracks every 10000 schedule=0
************ Average no of nodes=0 912300
************ Backtracks every 10000 schedule=0

Number of scheduling times of 0s            181393
Number of scheduling times of  50s          2914
Number of scheduling times of  60s          2893
Number of scheduling times of  110s         0
Number of scheduling times of  >110s and <= 250 s  0
Number of scheduling times of  >250s               0

Average no of nodes=0 241019
Max no of nodes=44
Max no of tree levels=6
```

Truncated A* search

Search is interrupted when number of tree levels is equal to number of lifts