Aloizio Pereira da Silva

Advisor: Geraldo Robson Mateus

Location-Based Taxi Service in Wireless Communication Environment

Dissertation presented to the Graduate Program in Computer Science at the Federal University of Minas Gerais, in partial fulfillment of the requirements for the degree of Master of Computer Science.

Belo Horizonte

March 25, 2002

Abstract

The convergence of three technologies, the Internet, wireless and location technologies is creating new opportunities for mobile devices. Mobile location-based services are emerging as one of the study topics in the mobile communication area. Basically, they use both the information of mobile unit's (MU) physical position, which can be obtained by GPS (Global Positioning System) and the information contained in a user profile to answer to his needs at that moment. Many investments have been made in the field of networks in resource allocation, mainly in wireless communication networks. Recently, wireless communication has moved from voice service to data services. Wireless protocol makes it possible for commonly used devices such as cell phones, PCs and PDAs (Personal Digital Assistants) to access and to manage information anywhere and anytime. In addition, there are a lot of other objects, that make part of a particular set. For instance, cars (taxis, ambulances, police car, trucks), aeroplane, etc. Due to that, it becomes necessary the knowledge where they are physically located and most of them will not be static, but constantly moving. This shift to the mobile world presents new aspects to the problem of resource allocation. The aim of this work is to model an application of mobile location-based service taking into account the available infrastructure of a wireless communication network. The process under study is developing a more efficient strategy for transporting users of taxi service from one location to another. The strategy explored in practice tends to be either centralized as in the case of dispatchers or static as in the case of courier service trucks that have pre-assigned pick-ups and deliveries in certain geographic regions. The implemented prototype uses a new strategy to allocate a specific taxi to one MU. In this case the assignment algorithm is similar, to those used in problems studied in the area of network optimization. The proposed strategy here is based on the MU communication with the taxi via a server, which performs whole assignment processing using location data. This prototype allows, through simulation, to compare the currently used queue-type strategy with the new solution proposal. Numerical results show the benefits of allocation procedure to assign taxis to MUs and to optimize the distance traveled and travel time if applied in a real-time context.

Resumo

A convergência de três tecnologias, a Internet, comunicação sem fio e sistemas de localização está criando novas oportunidades para os usuários de dispositivos móveis. Serviços baseados na localização de unidades móveis (UMs) estão emergindo como um dos tópicos de estudo na área de comunicação móvel. Basicamente esses serviços utilizam a informação da posição física do dispositivo móvel, obtida através do GPS (Global Positioning System) por exemplo, em conjunto com a informação do perfil do usuário de forma a atender aos interesses do usuário naquele momento. Têm sido grande os investimentos de pesquisa na área de redes em alocação de recursos, principalmente em redes de comunicação sem fio. Recentemente, a communicação sem fio têm possibilitado além dos serviços de voz, os serviços de dados. Protocolos sem fio tornam possível para os dispositivos usados atualmente, tais como telefones celulares, PCs e PADs (Personal Digital Assistants) acessarem e gerenciarem a informação em qualquer lugar e a qualquer momento. Além disso, existem various outros objetos, que fazem parte de um conjunto particular. Por exemplo, carros (taxis, ambulâncias, carros de policia, caminhões) aviões, etc. Devido à isso, torna-se necessario conhecer onde eles estão localizados fisicamente, tal que muitos deles não estarão estáticos, mas movendo constantemente. Esse deslocamento para o contexto móvel apresenta novos aspectos para o problema de alocação de recursos. Este trabalho pretende modelar uma aplicação de serviços baseado na localização, levando em conta a infraestrutura de comunicação sem fio disponível. O processo em estudo é desenvolver uma estratégia mais eficciente para transpostar usuários de serviço de taxi de uma localização para outra. A estratégia em uso tende a ser centralizada, como no caso do despachante, e estática, no caso dos veículos do serviço de correio que têm busca e entrega pré-estabelecidas em certas regiões geográficas. O protótipo implementado usa uma nova estratégia para alocar um taxi especifico a uma UM. Neste caso o algoritmo de alocação é similar aqueles usados em problemas estudados na área de otimização em redes. A estratégia proposta aqui é baseada na comunicação da UM com o taxi via um servidor, o qual realiza todo o processamento de alocação usando dados de localização. Esse protótipo permite, através de simulação, comparar a estratégia do tipo fila usada atualmente com a nova solução proposta. Resultados numéricos mostram os beneficios do procedimento de alocação para atribuir taxis as UMs e para otimizar a distância percorrida e o tempo de viagem se aplicado em um contexto de tempo real.

Acknowledgments

First of all, I would like to thank Dr. Geraldo Robson, my advisor for his sincere and constant guidance, help and support during the entire process of this project.

Second, I would like to take this opportunity to thank my mother and sister for their support throughout my education, and my friends for their support during my dissertation.

Third, I would like thank my committee members, Dr. Antônio Loureiro, Dr. Claudionor Coelho and Dr. Fernando Moreira, for their useful and valuable comments on this project.

Sincere thanks all of the teaching staff at the Department of Computer Science at UFMG.

Finally, I would like to thank Jesus Christ. He is the living source of my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recently, researches into ubiquitous computing has shown that monitoring of the objects location is a feasible method of determining the state of an environment. By accurately tracking the objects location it is also possible to derive other useful information, such as their orientation and how they are moving. Location-aware systems, whose behaviour is controled by these data types, embody a practical subset of the context-aware computing paradigm, and several systems based on this principle have already been developed. The evolution of this paradigm and the communication tecnhology enables portable devices (PDA-Personal Digital Assistant, Cellular Phone, Palmtop, etc) to be equipped with a wireless interface, allowing access to a shared infrastructure, even while mobile. This combination of networking and mobility has engendered new applications and services. One such service is Location-Based Service (LBS) [Wal98] [And01] [Hay00] [Swe99] [Gra00] [ESR00] [XL] [Pat00].

These services can be described as applications that exploit knowledge about where an information device (user) is located. That is, mobile users (MUs) are able to use services based on their position or geographic location. In this context, knowing where a person or object is at any time presents a new dimension to the kinds of information services. In addition, relating location to other pertinent information it is possible to develop information systems capable of integrating and managing several data anywhere and anytime.

For example, knowing that we are one mile from a particular facility may give some small comfort that we are getting closer, but we can obtain an even more valid or reliable route to that destination. Many kinds of services, such as roadside assistance, stolen vehicle recovery, traffic alerts, driving directions, vehicle tracking, taxi fleet management, the administration of container goods and the assignment and grouping of railway repairmen are some location-based services that are emerging in the world. Given the ubiquity of location information, the increasing mobility of modern society and the anticipated availability of broadband communication and mobile interface, location services are extremely differentiated and they can take many forms.

In this way, the same set of functions could be applied in different applications to find all types of facilities or assets that are geographically dispersed. One application that has been following this idea is the vehicle tracking application.

For many years, trucks and emergency vehicles have been tracked manually by having the location from driver radio. However, if we campare a vehicle's position in relation to an origin or destination it will not provide dispatchers with information about the status of a fleet of vehicles. Now, with the advances in wireless communication technologies [Wal99] and LBS [PKK01] [Dav01] [BL01], it is possible to track any type of vehicle or mobile object. For instance, the usage of satellites to track and dispatch radio taxis, facilitates finding and locating a vacant taxi nearest a mobile user (MU) quickly and accurately, besides alerting the driver to where the passenger is waiting.

The previous description, which uses the satellite as wireless communication technology added with the LBS basic ideas are the basis of this dissertation.

Specifically, this dissertation reports on a specific LBS, namely, Location-Based Taxi Service (LBTS) in a wireless communication environment. It uses location information of both a MU and taxis, to assign the nearest taxi to the MU, referring to a local area road map. In addition, it considers the shortest distance and time to pickup and transport the MU to its destination point. The dissertation also describes the assignment problem of taxis to MUs, which is solved through a network optimization model and discusses the solution proposed here with the current approach.

This assignment problem can present different forms. From the MU viewpoint, the assignment process looks to be very simple since the best assignment is the nearest taxi of the MU. They are always satisfied. In this case, only the metrics shortest time and distance are considered.

Investigating the problem of the taxi company viewpoint and taking into account the two previous metrics (shortest time and distance), another dimension of the problem appears. The taxi companies generally want to obtain gains. In order to achieve this goal, the company needs to serve a high demand, minimizing the distance traveled by its vehicles fleet, that is the least fuel cost.

From the taxi company viewpoint is not sufficient that the taxi must be near of the user. That is, the roads direction are now considered. Thus, the assignment process besides considering the user physical position, also taking into account the roads direction in the distance calculation. It is important to note that a taxi can be near to the user in relation to the physical position of both, but when the roads direction is considered, the distance that the taxi should travel to pickup the user could be greater.

It is clear that developing assignment process which consider only physical position is not very useful for the context of this work. Concerning previous discussion the assignment process can get more complicated when the MU changes its physical position. The problem is the assignment process obtains the user physical position in the moment of the service request and it uses this position to compute the best allocation. Then, if the MU moves the allocation could not be the best.

This happens because the location information is a dynamic data. In this case the assignment process should consider mainly to real-time environments, the dynamic routing issue. The own assignment process should be dynamic, such that it to be able to obtain the current location information and to update the assignment process execution in real-time.

One way to solve the previous problem is using MU location information at the moment it makes a service request. Thus, this datum is used throughout the assignment process execution. The MU does not change its physical position until the taxi arrives and the assignment process is not updated in real-time.

This dissertation uses the previous solution which it makes the problem more simple to solve.

Another problem with the assignment process is directly related to the taxis fleet. In a normal day, one or more taxis could not be allocated to serve any service request. In this case, the assignment algorithm could be modeled to divide the workload among the taxis fleet. The disadvantage with this load balancing is that would be seldom guaranteed the best allocation. One solution would be to define a priority policy to the taxis allocation. For example, the service request arrival might to be directed to those vehicles that was stopped for a specific time without serving any service request.

Concerning the previous descriptions we can note that the problem to assign taxis to MUs based in the physical position is a generic problema, mainly when another metrics besides distance and time are considered. In this work, we choose to work with time and distance metrics since both are basic metrics in this kind of problem.

Basically, the resource assignment problem and wireless communication tecnologies and mobile computing are the base of this dissertation, which allows to meet similar needs for understanding and evaluating many options in the wireless application development. Thus, our specific interest is to predict the impact of the interaction between the assigment algorithm in the mobile environment and fleet management on the desired quality of service for the MUs. A simulator was written to arrive at some conclusions.

1.1 Review of Computer Dispatch

Computer dispatch has its origin in the taxi company, beginning in 1970. For decades before taxi companies acquired computers, taxi dispatching was performed manually using papers, voice radios, and a variety of devices. Each transport request was written on a small paper, which was transported to a dispatcher. The dispatcher would select a driver and communicate the pick-up information by voice radio to the driver.

Among the problems of manual dispatching are accuracy and speed. Accuracy was limited by the ability of the staff to avoid losing the call before they were dispatched. Speed was related to volume. During busy periods a manual dispatch operation could not keep up with the calls because of too few dispatchers and too few radio channels. Quality of service was directly a function of the speed and accuracy of the dispatch operation.

1.1.1 Taxi Computer Dispatch Operation

Computer dispatch typically operates using data communications to connect vehicles with a central dispatch computer. The calls are received by computer, which determines which vehicle is to be assigned to the call. Digital call information is transmitted to the selected vehicle and appears on a vehicle screen. If the driver accepts the call, the computer transmits the entire information about the call, and the driver proceeds to pick up the passenger.

The communication flow is illustrated in Figure 1.1. The computer sends a signal via telephone line to the Network Call Processor (NCP), which recognizes the fleet and the single vehicle to which the message is intended. The NCP directs the message by telephone line to the network of base stations, then the message is sent by radio frequency to a particular taxi. The first taxi to receive the message is the one which is first in the queue of the zone of the call's address. When the driver responds, the procedure is reversed. The base station receives the message and relays it to the NCP. The NCP recognizes the address of the taxi and routes the message to the computer.

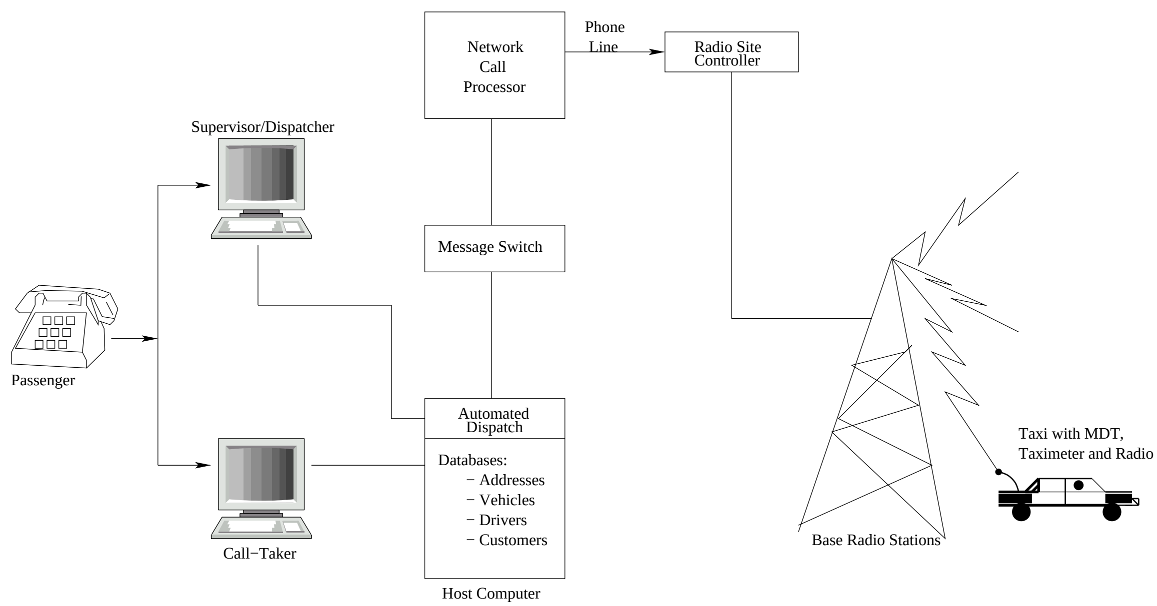


Figure 1.1: Taxi dispatch system

The system includes the following:

- Call-taker terminals in the dispatch office for call data entry,
- Backup computers with dispatching functions,
- Message switch that routes digital messages between dispatcher and drivers,
- Network call processor (NCP) to store and prioritize all messages between the computer and vehicle mobile data terminals (MDT),

- Radio site controller to code and decode digital and radio frequency information,
- One or more base radio stations for data transmission,
- Vehicle mobile data terminal to communicate with the driver, and
- Vehicle taximeters linked to the MDTs to track mileage and fares.

The computer contains an address database (database geographic) with the address ranges for all streets in the service area. The address ranges are mapped to three digit zone numbers. The zones are defined by the taxi company and are usually about four square miles in size. The major traffic generators such as a large hotel or airports, are often designated as separate zones, whereas the zones are large where taxi demand is low.

Automated dispatch can be accomplished with fast personal computers depending on fleet size and dispatching volume. For example, as shown in Table 1.1[1], a medium-sized taxi fleet with 100 vehicles can be accomodated by a personal computer system. However, several company representatives indicated that minimum fleet size for automated dispatch is usually about 200 taxis.

Company	**Location**	**Number of Cars**	**Year Installed**
Abraham Transportation	Tampa FL	200	1988
Alberta Co-op	Edmunton ALB	450	1981
Bell Radio	New York NY	350	1985
Blue Bird Cab	Hyattsville, MD	220	1992
Blue Line Taxi	Ottawa ON	700	1979
Cab Co-op	Toronto ON	420	1988
Checker	Detroit MI	475	1990
Checker and Yellow	Chicago IL	700	1988
City Cab	Orlando FL	350	1986
Communicar	Corona, NY	335	1991
DiamondTaxi	Toronto ON	600	1986
Farwest Service	Seattle WA	150	1991
Jacksonville Trans	Jacksonville FL	200	1988
San Francisco Yellow	San Francisco CA	275	1988
Unicity Taxi	Winninpeg AB	250	1981
United	Philadelphia PA	725	1995
UTOG Two-Way Radio	New York NY	350	1988
Yellow Cab	Anaheim CA	100	1985
Yellow Cab	Columbus OH	150	1988
Yellow Cab	Hamilton ON	100	1982
Yellow Cab	Hartford CT	150	1988
Yellow Cab	Indianapolis IN	360	1985

Table 1.1: Taxi Companies With Automated Computer Dispatch (Gandalf Mobile Systems)

Two parameters have been considered in [SGN92] to be indicators of increased productivity: call time and response time. Call time can be reduced if special locator identification

[1]source: U.S. Departament of Trasnportation

numbers are used. Response time measures the time between the completion of call and the driver's arrival at the passenger's location.

1.2 Thesis Organization

This dissertation is organized as follows. Chapter 2 describes some related work into three research areas: mobile computing, resource allocation and dynamic vehicle routing problem. Chapter 3 reviews the concepts regarding Mobile Location-Based Service(LBS), which we have used throughout this work. Chapter 4 describes the problem looked at in this dissertation and discusses both the current and newly proposed approaches to taxi service. In addition, it describes how the taxi assignment problem and the solution procedure are modeled. Chapter 5 discusses the simulator's architecture and implementation issues. The simulator was critical for both evaluation purposes and for gaining insight into the effect of assigment algorithm processing in action in the mobile environment. Chapter 6 presents the experiments conducted using the LBTS simulator and a discussion of the results obtained. Finally, Chapter 7, summarizes the work, presenting conclusions and suggesting future work.

Chapter 2

Related Work

2.1 Introduction

Three distinct bodies of research related directly to the focus of this dissertation. The first is mobile computing, in which users carring portable devices have access to a shared infrastructure independent of their physical location [FZ94] [MC93]. The second field is resource allocation, which has been a popular topic in both economics and software agent research and has been used to attack problems as diverse as those involved in distributed computing [AMO93] [SM]. The third body of relevant research focuses on developing solutions to the problem of dynamic vehicles routing problem (DVRP), which has been studied extensively in networks optimization [BSL] [Pub] [SG01] [JKV98].

2.2 Mobile Computing

The evolution of wireless communication networks has made the communication among people more flexible and it has considerably increased the access to service networks. Advances in such technology enable portable computers to be equipped with wireless interfaces, allowing access to a shared infrastructure, even while mobile. Wireless networking greatly enhances the utility of carrying a computing device. The combination of networking and mobility has engendered new applications and services. One such service is location-based services, which will be described in next chapter.

A survey of the fundamental software design pressures specific to mobile computing is presented by Forman and Zahorjan [FZ94]. They have examined the repercussions of three principal features of mobile computing: wireless communication, portability and mobility. Wireless communication brings challenges to network conditions, making access to remote resources often slow or sometimes temporarily unavailable. Portability entails limited resources available on board to handle the changeable mobile computing environment. Mobility causes greater dynamism of information. In addition, in their work, they discuss the issue of location dependent information. Because traditional computers do not move, information that depends on location is configured statically, such as the local name server and available printers. In this case, a challenge for mobile computing is to factor out this information intelligently and provide mechanisms to obtain configuration data appropriate to the present location. Besides this dynamic configuration problem, mobile computers need access to more location sensitive information than stationary computers if they are to serve as guides in places unfamiliar to their users, for example, to answer queries like "Where is the nearest open gas station heading north?". Whereas such queries require static location information about the world, Badrinath and Imielinski are studying a related class of queries that depends , for example on determining where the nearest taxi is [SIG93].

Xu and Lee have studied system design issues for querying wireless location-dependent data (WLDD) [XL]. In their work, they describe a server architecture for WLDD as shown in Figure 2.1. Basically, this architecture is a general model for wireless cellular systems. The geographic coverage area for information services is partitioned into wireless cells. Each wireless cell is serviced by a base station (BS), which is a leaf node in the fixed network. A mobile client can connect to a corresponding BS over the wireless network to access relevant information at any time. They define the valid scope of an item value as a set of cells where the item value is valid, and all valid scopes of an item form its scope distribution. Perhaps for simplicity, they assume valid scopes for all item values are hierarchical and thus they can be represented by a tree (as shown in Figure 2.1).

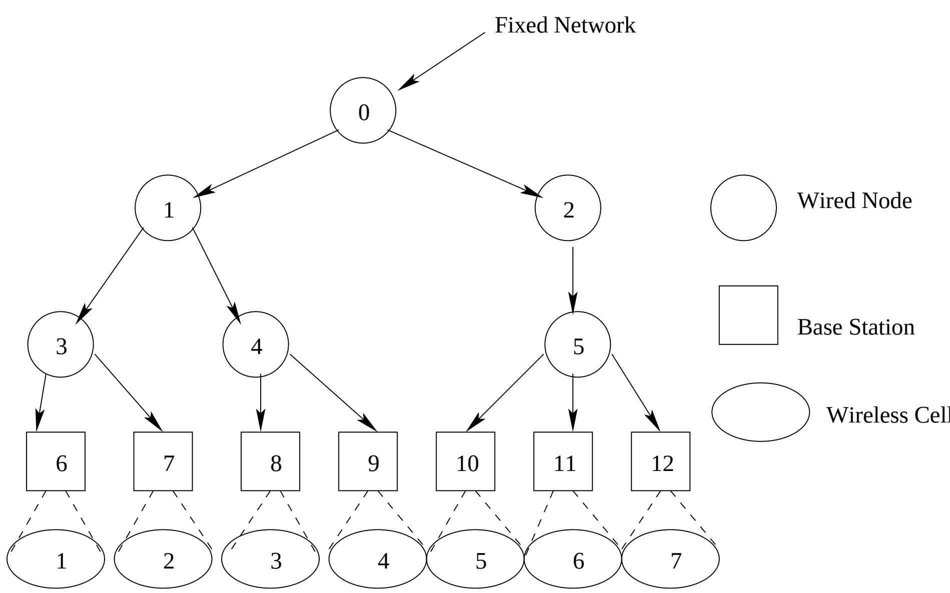


Figure 2.1: Server Architecture for WLDD querying systems

Also in the work of Xu and Lee, five query types are identified, where the query processing methods are discussed for distributed data server architecture. In addition, they discuss the QoS guarantee of queries of viewpoint data caching and replication, and partly by query scheduling. Since mobile users will move around and change off from one cell to another, a query should be responded before the client set the handoff procedure. These types of query are described below.

- **Local vs. Non-Local Queries:** Local queries refer to the queries whose results are valid in the current cell. Sample queries are like "list the local hotels", "list the local health centers", etc. To process such a query in distributed data server architecture, the system searches relevant items from the database at fixed nodes from the current BS to the root until the query is satisfied. Non-local queries refer to the queries whose results are valid in another cell. Sample queries are like "find the weather in Cell #8", "find the hotels with a room rate lower than $200 in Cell #8." To process such a query, they re-direct the query to the corresponding cell, and then follow the same method as for local queries.

- **Geographically Clustered vs. Geographically Dispersed Queries:** Geographical clustered queries refer to the queries with spatial constraints, for example, "list all hospitals within a 500 km radius." In this case the queries can be answered by a cluster of cells that are within the range, and processing at a cell can follow the same method used for local and non-local queries. Geographic dispersed queries refer to the queries that are not associated with any spatial condition, for example, "list all hospitals with a heart surgery facility."

- **Nearest Queries:** Nearest queries are another type of useful queries discussed, for example, "find the nearest gas station", "find the nearest heart hospital", "find the nearest taxi", etc. A nearest query can be processed as follows: the query is first processed at the current cell, if the result is found, the query is answered; otherwise, it will require other cells to process it from the nearest cell to the farthest until there is some result found.

A study of how location-based services can be provided to users is performed in Patel [Pat00]. In his work, he concentrates on issues regarding LBS, such as the need for LBS, the identification of requirements and technologies that enable LBS and the analysis and evaluation control of various technologies that can provide LBS. In addition, a prototype is produced to show the types of services and user interfaces that could be provided. This prototype is developed in an existing emulator phone, therefore the functionalities are not the same for different handsets and it also does not take into account like road map and geographic map.

Additional applications of location-based service to mobile environments have been discussed in [Int00] [Sie00]. In [Int00] ,a prototype service called 'Taxi!' is described , which uses location based technology to help users find cabs using their mobile phones. The taxi service allows users to call a taxi with the press of a button on their mobile phone. A central server logs the request, notes the user's location and creates an "eletronic marketplace" for local taxi companies to compete for the customer, ensuring that the users get the cheapest and quickest taxi. 'Taxi!' utilises Cambridge Positioning Systems [Cur]. Specifically, each taxi company has its own control centre and taxi drivers with mobile devices in their cars. When a passenger's request is made from a mobile phone, the mobile network searches for the closest taxi companies and creates a real-time reverse auction between them to get the best price and fastest pick-up time for the journey. In [Sie00], an intelligent network (IN) is combined with GPS to provide an efficient taxi service, which is called 'Taxi Scout'. The IN solution scenario can be described as follows in agreement with Figure 2.2.

1. The caller dials the 'Taxi Scout' service number.
2. The IN tries to determine the location of the caller.
3. If the information to locate the caller is missing or is insufficient, the INXpress will ask the caller for more detailed information, e.g. zipcode, street, etc.
4. The caller's location is used to interrogate an external database, which is supported by the satellite based GPS.
5. The GPS database identifies a suitable vacant taxi near the caller.
6. The information is returned to the IN system, which then uses this information to set up a direct connection between the caller and the appropriate driver.
7. The call center of the taxi operator will be given the calls that require manual service.

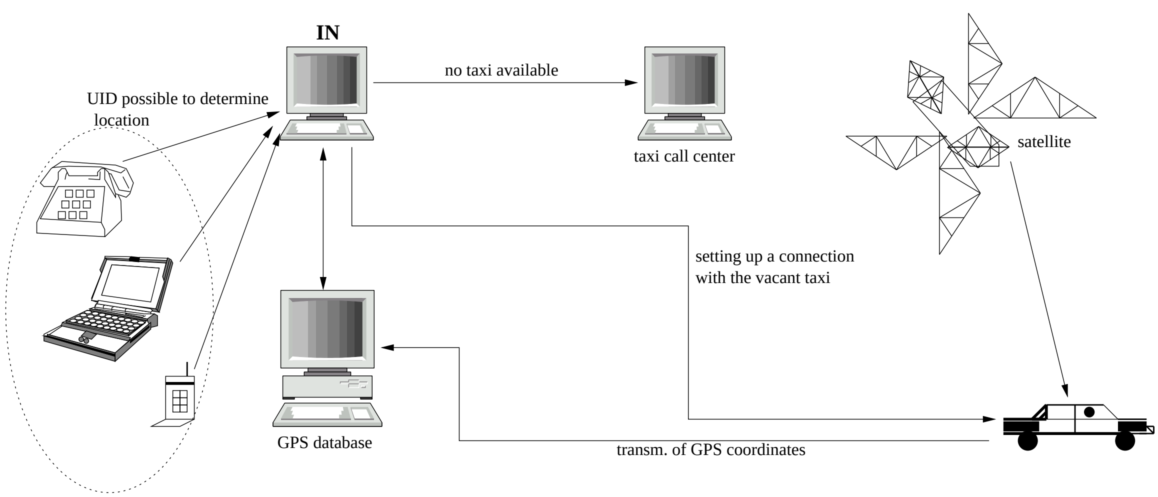


Figure 2.2: 'Taxi Scout', set-up of a direct connection between passenger

WebTrac [Web00] develops applications for wireless carriers and portals. It is a privately owned company based in Rehovot, Israel. The company has deployed its TracFlow platform (see Figure 2.3) at Telephone Communication to provide LBS to its customers. WebTrac markets three mobile service applications for the TracFlow platform:

- **TracInfo:** enables users to receive information based on their location. The service is accessible through interactive voice response.
- **TracFleet:** enables fleet managers to view and interact with their entire mobile fleet using Web, where they can track a mobile user's location and movement on a map. Managers can select any user on the map, read their details and send messages. All information is saved on a SQL database and can be used to generate a variety of reports.
- **TracMap:** can be used by any subscriber to trace someone on a map using the Web.

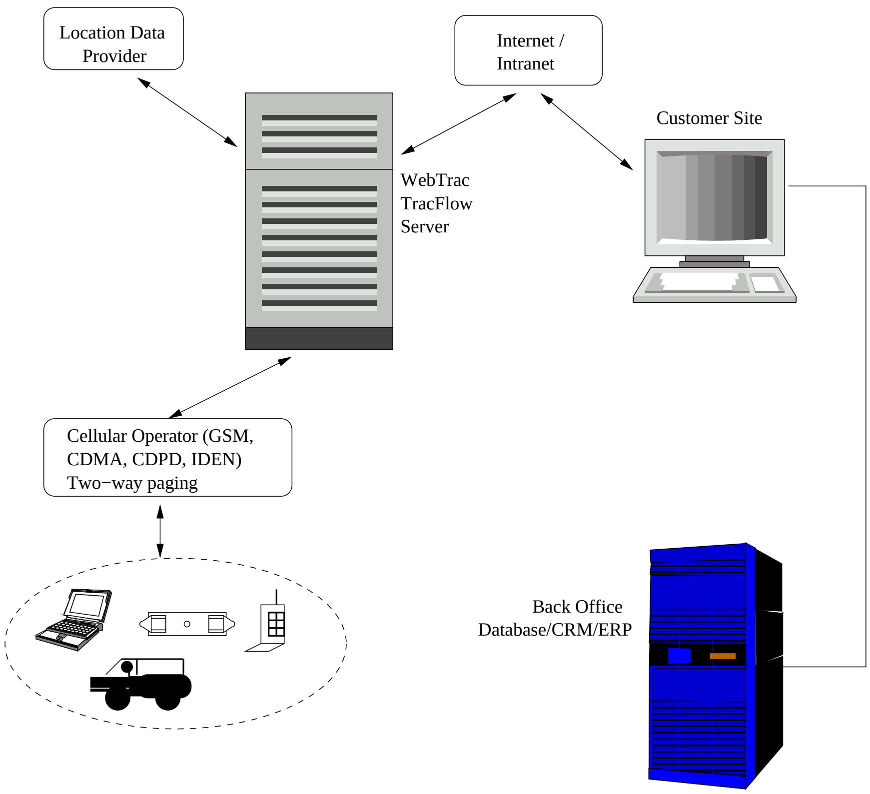


Figure 2.3: TracFlow platform for LBS

WebTrac offers some applications using the TracFlow technology, which are listed as follows:

1. Location-based taxi service - deployment and management options
2. Searching for nearest pub - using voice and SMS (Short Message Service) or WAP (Wireless Application Protocol)
3. Location-based traffic updates - traffic status request and traffic alert notification using SMS

4. Fleet management reporting

In the previous work, we encountered a similar application to the working that have been researched in this dissertation.

In [Tru00] some applications for wireless location systems are described. The primary applications include emergency management, fraud management, location sensitive billing, vehicle and fleet management, inventory/package monitoring and wireless system design. One of the oldest applications for location systems is vehicle and fleet management. Many of the original systems created in the 1970s were conceived for this purpose. For example, a carrier can provide access by a dispatcher to the location system using an electronic map workstation, and can support all of the same features and functions used on the specialized systems. This work presents a discussion of the applications that can be developed for wireless telephones considering the addition of a location system in the carriers networks.

Major applications for mobile data up to now have all been in the vertical market segment (specific to a particular industry or mode of operation), but new ones are gradually shifting to the horizontal (those applications that can be used across several fields of endeavor) as data services over analog and digital cellulars and over public data networks and LANs (Local Area Networks) permit more general interconnection among diverse users. Both horizontal and vertical applications are described in [Wal98]. Basically, the main horizontal functions of mobile data usage can be broken down into four categories: dispatch, database inquiry and update; interpersonal messaging (e-mail and paging) and remote control and reporting. The dispatch category is perhaps the most important for the scope this work, a courier or taxi company is given a job assignment or an address. This particular application verges on the vertical segment and it has been essential part of working methodology presented in [Wal98]. Moreover, a single application could be used by several industries, particularly the utility companies, in which systems have been in service for several decades. In the case of vertical application, depending on the level of mobility, applications vary between the use of in-house wireless LANs [Mul94], which are used for computer networking purposes, to in-the-field activities, which can involve long-distance communications via satellite. Most classic mobile data application are associated with data from terminals that are fitted in vehicles and transmitted over the standard PMR[1] (Private Mobile Radio) frequency allocations in the VHF (Very High Frequency) and UHF (Ultra High Frequency) bands. Many of the earlier applications were involved with vehicle tracking, or AVL (Automatic Vehicle Location). For instance, a new variation on the normal dispatcher operation has been introduced by one of London's taxi cab cooperatives. In a small dispatch operation with a few tens of vehicles, the controller is able to maintain knowledge of the taxis' status and approximate position and has little difficulty with job allocations. A simple mean of offering jobs to drivers has been devised based on the random access protocol inherent in the data system. If a customer phones the taxi company for a pickup, the dispatcher broadcasts to all cabs a bid request and the pickup location. Taxis that are free within a defined radius of, say, a quarter of a mile can

[1]PMR refers to radio trunking services that provide facilities for mobile users in company vehicles to connect to private PMR voice telephony and data networks.

make a bid by pressing one of the status buttons. The random access protocol sorts out the call contention, and one of the units succeeds in becoming the first to answer the call and automatically is awarded the job.

2.3 Resource Allocation

The history of the resource allocation problem dates back to the paper by Koopman [Koo53], in which he considered the problem of a distribution of work between two activities. Koopman mentioned some examples, such as the distribution of destructive power that is problematic in military applications. He also derived analytical solutions in some special cases. After this, Koopman and Charnes and Cooper [CC58] studied generalizations to the resource allocation problem.

Basically, the problem of resource allocation is deciding how to allocate a set of tasks among a limited amount of resources in such way as to maximize the efficiency of the system. A common example is distributed computing where programs are split up among a set of processors with the goal of minimizing the time necessary to run all the programs. Traditional solutions have used a centralized controller such as a task scheduler, which is a master process in charge of assignning the other processes to computers. Recent research has explored the use of market as an alternative solution for assignment problem based on the price parameter. Among the relevant work that has been done in this area, there are two that have explored the use of markets as an alternative solution [Mal98] [Gre00]. In 1998, Tom Malone wrote a program called 'Enterprise' [Mal98] that simulated distributing programs among processors. More recently, Gregorovic developed a market-based mechanism for allocating agents to location based tasks [Gre00], where agents bid for packages along a set of parameters such as cost and delivery speed and the package chooses the winning agent.

A brief discussion of personnel allocation is described in [DW00]. Given a set of people with various qualifications and preferences and jobs with various requirements and preferences, the task is to assign the people to the jobs intelligently. In this case, the immediate task is to assign a set of leaders to a list of subjects. Each leader candidate has a description of their academic record, their personality, and other such useful information whereas the classes gives a list of requirements that must be fulfilled by the leader and a list of preferences that they would like to see fulfilled. Each class requires one leader, and each leader can be assigned to either one or zero subjects. They do not also consider multiple assignment, that is, the task of assigning multiple leader per subject.

In terms of related work, there is significant research in the area of distributed resource allocation; for instance, [LS96] describes an environment where agents' tasks are tighly connected and require real-time scheduling and execution. An approach that consists of a standard operating procedure and a look-ahead coordination is presented. Moreover, the work extends dispatch scheduling to improve resource allocation.

In distributed resource allocation a set of agents must assign their resource to a set of tasks. This problem arises in many real-word domains such as distributed sensor networks [San01], disaster rescue [Kit96], hospital scheduling [KJ98] and others. A formalization of distributed resource allocation that represents both dynamic and distributed aspects of the problem is discussed in [MJT+01]. These two aspects presents some key difficulties. First, a distributed situation results in agents obtaining only local information, but facing global ambiguity, that is, an agent may know the result of its local operations but it may not know the global task and hence may not know what operation others should perform. Second, the situation is dynamic so a solution to the resource allocation problem at one time may become unsuccessful when the underlying tasks have changed. According to [MJT+01] the agents must continuously monitor the quality of the solution and must have a way to express such changes in the problem. Given these parameter of ambiguity and dynamism, it is defined four classes of difficulties of the problem. In this case the difficulty of the problem depends on both the ambiguity of the task from a dynamic distributed environment and the relation among tasks which may require conflicting resources. In our work, the physical position of both MUs and taxis is a dynamic parameter, that is, the mobility of them must influence in the quality of the assignment algorithm solution.

In addition, [MJT+01] also defines the notion of DDCSP (Dynamic Distributed Constraint Satisfaction Problem). This problem looks for identifying current tasks that can change over time and assign operations that are required by the current tasks. A Constraint Satisfaction Problem (CSP) is commonly defined by a set of variables, each associated with a finite domain, and a set of constraints on the values of the variables. A solution is the assignment for the variables which satisfies all the constraints. A distributed CSP is a CSP in which variables and constraints are distributed among multiple agents. Each variables belongs to an agents. A constraint defined only on the variable belonging to a single agent is called a *local constraints*. In contrast, an *external constraint* involves variables of diferent agents. Thus, solving a DCSP requires that agents not only solve their local constraints, but also communicate with other agents to satisfy external constraints. Through both theoretical analysis and experimental verifications, the DDCSP is applied to the distributed sensor network problem.

The advent of management information services and data processing greatly improved the ability of terminal managers to control the whole process, but yet raw data has to be analysed and treated to provide some insight on the performance of terminal operations. Simulation models and operations research techniques have proven to be a reliable and convenient tool to support the decision-makers in the daily operations in many cases [BS96a] [ZG]. An approach to the problem of deciding which equipment and manpower must be allocated over a sequence of work shifts is presented in [ZRGM98]. The solution of the resource allocation problem is based on a network design formulation that assumes that the loading and unloading processes can be modelled as a network of container flows between the ships and the terminal yard for all the work shifts. A discrete event simulation model is used to validade the resource allocation and the scheduling policies.

A few years ago a need for radical improvements to the allocation and dispatch of jobs to a mobile workforce was identified, and the vision of a fully automated work allocation system was proposed as part of British Telecommunications's work management programme. In [Lai95] is described such system. Basically, the paper describes the scope of work to be allocated by the system, then highlights the main algorithm design problems and describes the overall approach taken to tackle them. In addition, it is also presented a more detailed explanation of how some of the challenges were solved using operational research algorithms and other techniques. In our work, we also solved the taxi assignment problem to MU using operational research or network optmization algorithms. [Lai95] also defines that some of the main problems solved by the scheduling and allocation algorithm can be categorised into three areas as follows.

1. Theoretical
   - how to balance the many real cost (e.g. travel time, overtime) and virtual cost (e.g customer perception, risk of failure) associated with allocating a job, and produce the best overall cost of allocating many jobs,
   - how to estimate travel time to a sufficient degree of accuracy and cope with temporary changes such as rain, abnormal traffic conditions and roadworks;
2. Technical
   - how many, and which jobs and engineers to consider, and how to constrain the solution space - given finite computing resources,
   - how to integrate potentially quite different algorithm techniques within one system and get them to work together where necessary;
3. Practical
   - how to make the overall algorithms flexible (parameter or rule-driven) so as to allow the allocation objectives to be changed on, or even during, the day, e.g. in storm conditions to minimise travel, whilst in normal load conditions to minimise customer commitment failures (minimising travel as the second objective),
   - how to determine the best rules or parameter sets to optimise performance in different operational circunstances.

Properties of the assignment problem are described in [Bri97] as follows.

- for each i (resource), exactly one assignment $X_{i,j}$ is made.
- for each j (activity), exactly one assignment $X_{i,j}$ is made.

Therefore, if a number δ is added to (or subtracted from) every cost in a certain row (or column) of the matrix C (costs matrix), then every feasible set of assignments will have its cost increased (or decreased) by δ, and the optimal set of assignments remains optimal. For example, if we add δ to row 1, the total cost is increased by $\sum_{j=1}^{n} \delta X_{i,j} = \delta \sum_{j=1}^{n} X_{i,j} = \delta$ (independent of X). Moreover, if all cost $C_{i,j}$ are nonnegative, and if there is a set of assignments with total cost equal to zero, then set of assignments must be optimal. In our work, we use the properties described in [Bri97], so each of MUs must be assigned to one of n taxis, and each taxi is assigned exactly one MU.

Mateus, Loureiro, Rodrigues and Goussevskaia present the server location problem in mobile computing [MLRG00], such that given a network comprised of mobile users and a set of servers that provide services, the problem at hand is finding an association between users and servers that minimizes the cost to access the servers in the network. The main goal of this work is to study the server and service location problem in a mobile environment typically supported by a cellular infrastructure.

2.4 Vehicle Routing Problem

The common definition of the vehicle routing problem (VRP) is, given a set of customer locations and a set of vehicles, deciding how to visit each of the locations. The objective is to minimize some functions (e.g. total distance traveled) and is usually subject to some constraints (e.g. the capacity of the vehicles). Although the name implies that vehicle routing is the sole focus of the problem, there are actually two key aspects: routing and assignment. The routing problem [Sam98] is deciding in what order a vehicle should visit customer locations while the assignment problem is deciding which vehicle should visit each specific location.

Vehicle routing problems can divided along the lines of static vs. dynamic. In a static VRP the list of customer locations is known ahead of time and the problem can be solved deterministically. This is akin to the travelling salesman problem and has been extensively covered in the research literature. A dynamic VRP deals with situations where customer locations are not known in advance but appear over time. Because of the combinatorial complexity of routing decisions, it is usually unfeasible to use an optimal algorithm in practice where routing decisions must be made quickly. Instead, a wide range of heuristics has been developed that have shown to be near optimal [BC97] [BMP95].

Assignment on the other hand is extremely relevant since it is responsable for the best utilization of resources. Virtually all the research of assignment in the VRP has taken the same form. When a new request arrives, the central controller tests each vehicle with the new request to see which assignment maximizes the overall objective function. The best vehicle is assigned then to that request. The approach taken in this dissertation assumes an overall objective function and it requires a central coordinator to perform the assignment.

Stone, Gilbert and Nalenvanko describe Intelligent Vehicle Highway Systems (IVHS) technologies [SGN92]. In their work, they discuss the results of an investigation into one area of computer dispatching and scheduling technology. The central research question is whether computer dispatch technology is, or will be, capable of improving the efficiency of dispatching shared-ride vehicles on a real-time basis. They also examine current computerized dispatching capabilities of taxi companies and computerized scheduling capabilities of paratransit operations. The U.S. Departament of transportation [Gm91] has applied IVHS technologies, such as eletronic payment media, automatic vehicle locator systems, and intelligent databases, to highways and roadways. The Federal Transit Administration

(FTA), has made research and development of IVHS technologies a priority through their Advanced Public Transportation Systems (APTS) program. The process of rethinking transit service can be noted in FTA's mobility manager concept. The mobility manager is a mechanism for achieving the integration and coordination of transportation services offered by multiple providers (public, private) and involving a variety of travel models (bus, taxi, van, etc). It is a central agent who not only provide one-stop shopping for transportation information, such as schedules and fares, but also can make real time reservations for a person while on the phone. Figure 2.4 shows how the mobility manager concept might work [Jef91].

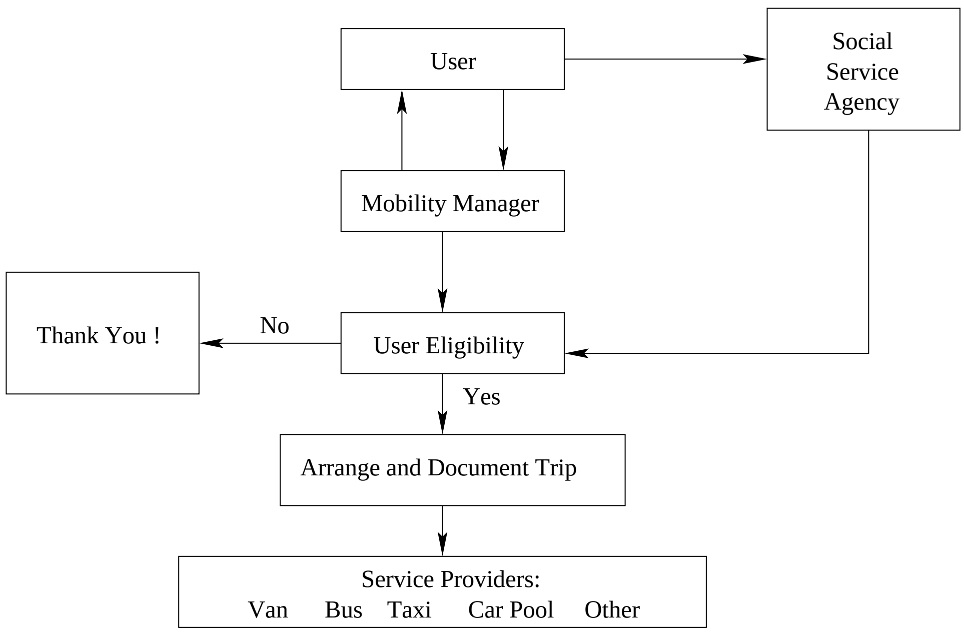


Figure 2.4: Mobility Manager Concept

The management of a fleet of taxi vehicles constitutes a particular case of the dynamic transportation problem. In agreement with [PQTC01] dynamic transportation problems are depicted by a number of complex and interrelated features, including:

- Multiple demands to transport commodities or passengers from/to origin/destination points
- Multiple vehicle resources to be routed and scheduled to meet the demands
- Multiple vehicles types with different capabilities and limitations
- Dynamically changing demands, constraints, and/or resource availability
- Dynamic planning of routes and destination/origin points according to current situation

- Multiple constraints, such as time window, vehicle capacity, trip duration, resource optimization
- Unpredictable events, such as traffic situation, weather conditions, vehicle breakdowns, and son on.

In addition, this work describes a prototype of an intelligent taxi deployment system based on the blackboard architectural approach. The Generic Blackboard Shell (GBS) [Nil98] is a framework for the development of truth-maintained blackboard systems under development at the Intelligent Systems Laboratory. The GBS offers the advantage of combining the constructive method realised by the blackboard system architecture with the iterative method supported by the truth maintenance system, while allowing the integration of heterogeneous techniques.

The main components of GBS are shown in Figure 2.5. The Blackboard Database centralises all problem information, including the solutions under construction. The Knowledge Sources are independent, specialised problem solvers that make use of the available information and contribute in turn more data for other Knowledge Sources to work on or partial solutions to the problem. The Control Structure repeatedly monitors changes in the Blackboard Database and activates Knowledge Sources accordingly until a satisfactory solution has emerged. In addition, the Assumption-based Truth Maintenance System (ATMS) works with the Control Structure to record, during the construction of a solution, the reason and impact of each decision taken. Whenever a contingency invalidates the proposed solution, this information can be used to help determine which part of the solution should be modified and how.

Specifically, the proposed Intelligent Taxi Deployment System comprises simulation models for the road network, traffic flow, vehicle locomotion and control, manoeuvres, as well as various problem solving strategies for vehicle assignment and routing. The system makes use of the GBS to integrate all these techniques transparently and provides solutions that can be adapted dynamically according to passenger request and traffic conditions.

A smart satellite-based tracking and booking system for taxi service was initiated in Singapore in 1996 [Kah96], using GPS technology. With the launch of a new dispatching system, which makes use of satellites to track and dispatch radio taxi, dispatchers can find and locate the vacant cab nearest a caller quickly and accurately and alert the driver to where the passenger is waiting. This report also discusses the advantages and limitations of the new system and other applications of GPS technology in Sigapore.

In [Wal98] the development of route guidance systems is discussed . Specifically, they calculate routes from a vehicle's current position to its destination based on digital mapping contained on CD-ROM in the vehicle. Positioning generally is based on map matching using a combination of data from magnetic field sensors, differential odometer measurement (normally derived from the sensors used by the vehicle's antilock braking system) and a GPS receiver. The driver is provided with route guidance recommedations by voice message and visual display. The first real availability of navigation systems in Europe appeared in 1995, with the launch in Germany of Philips's CARiN, initially in the BMW 7 series range of cars.

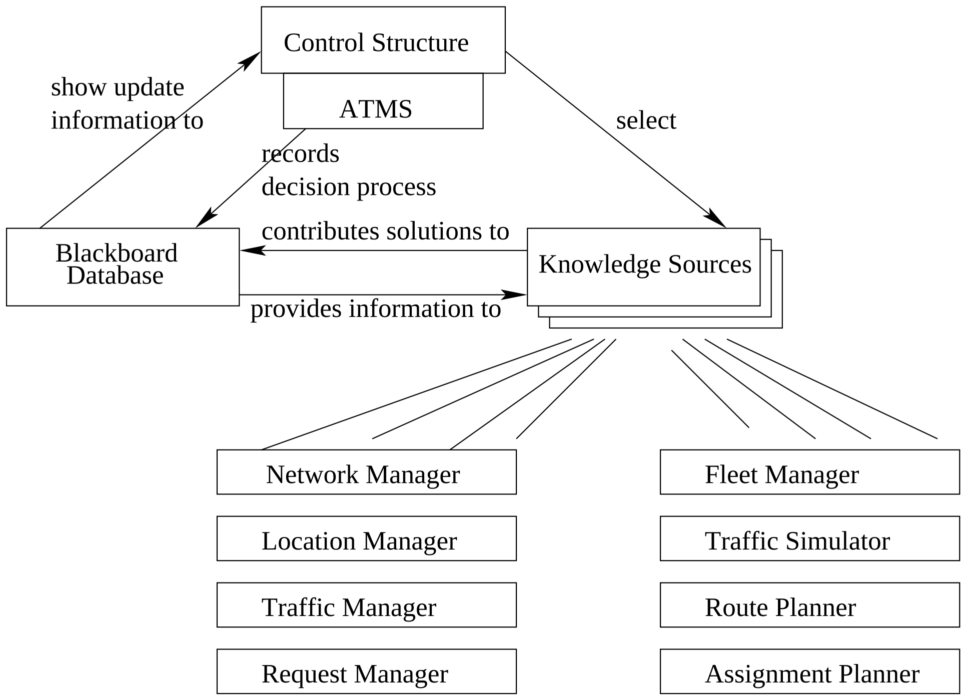


Figure 2.5: GBS-based Intelligent Taxi Deployment System

The vehicle routing problem as encountered in practice involves many restrictions on the routes that delivery vehicles can follow. Specifically, some of the more common restrictions are described in [BC97]. We can classify these restrictions to a certain extent as relating either to the vehicles or to the customers. It is important to note that in any particular case not all of these restrictions may apply, however in thinking generically about the problem it is useful to list all restrictions that potentially can apply. Below we describe these restrictions in three groups, vehicles, customers and other factors.

1. **Vehicles**

   - Each vehicle has a limit (capacity - usually weight and/or volume) on the load carried, e.g. buses have a limit on the number of people legally allowed on board.
   - Each vehicle has a total working time from departure to arrival back at the depot, typically to comply with legal restrictions on drivers' working hours.
   - Each vehicle has a time period within which it must leave the depot, typically to ensure that space is available for incoming vehicles to resupply the depot.
   - Each vehicle has a number of time periods during which it does nothing (drivers' rest periods).
   - Each vehicle has a cost associated with its use for deliveries.

2. **Customers**

- Each customer has a certain quantity which has to be delivered (or collected), typically we usually think of pure delivery operations but there are operations involving collections only and operations involving a mix of collections and deliveries (e.g. the business level express delivery/collection operations of DHL/Federal Express/UPS). Sometimes this quantity is known exactly (the deterministic case) and sometimes known with a degree of uncertainty (the stochastic case).
- Each customer has a number of time periods during which delivery can occur (time windows). For example a customer might be prepared only to accept delivery between 10:30 - 11:30 or between 14:00 and 16:15. These two periods of time are the time windows for the customer. Time windows are convenient to customers as they know when delivery is likely to occur and they can schedule deliveries to suit to the work pattern of their staff (who may need to check incoming goods, sign for them, etc). Time windows are inconvenient to delivery companies as they limit their flexibility (e.g. two customers right next door to each other might have very different time windows).
- Each customer has an associated visit time (drop time).
- Each customer has a set of vehicles which cannot be used for delivery (access restrictions)
- Each customer has a priority for delivery (if the vehicles cannot deliver to all the customers). Typically this might happen due to driver/vehicle unavailability.
- Each customer may or may not accept split visits (a delivery/collection by more than one vehicle).

3. **Other factors**

    - Multiple trips by the same vehicle on a single day, where the vehicle returns to the depot and then goes out again (e.g. post office vans).
    - Trips by same vehicle longer than one day (e.g. with overnight stops).
    - Compartmentalised vehicles with many different types of products to deliver.
    - More than one depot, where vehicles can start/visit/end at different depots.

Moreover, in [BC97] also discusses issues about design of vehicle routes. In this case to meet the above requirements there are a number of objectives that could be adopted. Three basic objectives can be distinguished:

1. Minimize the number of vehicles used (vehicles, and their associated drivers, often have a fixed cost)
2. Minimize the total distance (or time) travelled
3. Minimize some combination of (1) e (2), i.e., minimize the total of fixed and variable cost.

The cost of vehicles in the vehicle fleet is often regarded as a fixed cost so that the first objective corresponds to the minimization of fixed cost; the second objective corresponds to the minimization of variable costs; and the third objective corresponds to the minimization of total cost.

Intrinsically, the VRP is a spatial problem. During the last decades, however, temporal aspects of routing problems have also been studied. Often, users can only be served during certain hours of the day, such as office hours or hours before the opening of a shop. Furthermore, transportation companies try to compute service differentiation by delivering within prespecified time windows (VRPTW). The VRPTW is a generalization of the VRP involving the added complexity that every user should be served with a given time window. Furthermore, the objective evolves minimizing a combination of both distances travelled and the total duration of the routes. Jong, Kant and Vliet have searched for solutions for the VRPTW. In their work, they adjust the VRP to incorporate multiple time windows per users [JKV98]. In this dissertation, we do not consider VRPTW. On the other hand, we use time windows to run an assignment algorithm, which will allow the assignment of the best taxi to MU in agreement with their locations and distance.

Adaptive routing for road traffic is discussed in Fawcett and Robinson [FR00]. In their work, they describe an integrated system that uses congestion information to guide routing, both in advance and while in transit. It offers two features:

- Historic information about congestion is collected and retained for use when planning routes
- GPS tracks vehicles while they undertake journeys, and a Short Message Service (SMS) [FJG96] in the Global System for Mobile Communications (GSM) maintains communications between a moving vehicle and a central planning service to suggest revised routes avoiding congestion.

They also implemented a prototype using Java. It shows how computing, communications and location information combined in an information system can provide valuable facilities in an ubiquitous computing environment. In their prototype, they implemented Lee's algorithm [Lee61] to find the best route. In the case of our research, the Floyd-Warshall algorithm [CLR90] has been used to find the best route. The implemented prototype consists of several modules which represent the system structure as shown in Figure 2.6.

- The system collects congestion information from a variety of different network services. Converters for any eletronic data source can be written independently and the software can be asked to dynamically add data sources from the system without restarting.
- These data are collated into a single queue and copies distributed to registered data recipients, including a map manager that accumulates historic information and live data service to inform clients of unanticipated congestion.

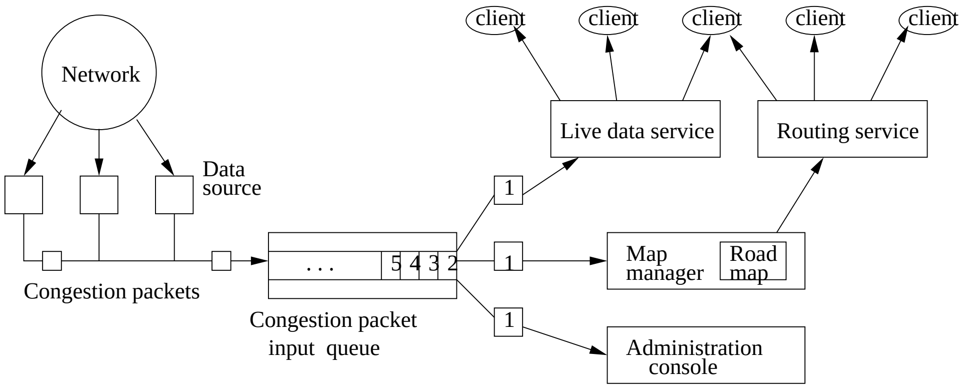


Figure 2.6: Structure of the system, which integrates congestion information

- The map manager annotates a detailed road map with congestion information for different times of the day and days of the week.
- The routing service uses the connectivity and capacity information in the map manager to find an optimal route between a pair of points for a given time of departure, in terms of minimizing the time needed to make a journey.
- The live data service monitors the current positions of clients and current congestion along their anticipated routes, and advises them of revised journey times and routes to avoid congestion.
- An administration console allows an operator to monitor and control the system.

Zhoo [Zha97b] provides a detailed description of both the principles and practices of modern vehicles location and navigation systems. He discusses issues about route planning, such that it can be classified into either for all vehicles on a particular road network, or single-vehicle route planning, which plans a single route for a single vehicle according to the current location and a given destination.

A variety of route optimization criteria may be used in route planning. The quality of a route depends on many factors such as distance, travel time, travel speed, number of turns and traffic lights, and dynamic traffic information. Specifically, these factors are refered as the travel cost in [Zha97b]. Some drivers may prefer the shortest distance. Others may prefer the shortest travel time, etc. Therefore, the evaluation function to minimize this cost depends on the system design and user preference.

These route selection criteria can be either fixed by a design or implemented via a selectable user interface. To minimize the travel distance, distance is usually stored in a digital map database, so that the route planning algorithm can use the database when performing minimization. If travel time is being minimized, the road segment lenght and speed limit are often defined as attributes of the road segment in the digital map database. In summary, determination of the best route involves using a digital map to select a route that minimizes variables such as time and distance.

2.4.0.1 Pickup and Delivery Problem

Basically, the problem discussed in this research is also related with the Pickup and Delivery Problem (PDP) [Rul95], which is a specific case of VRP. In its most basic form it consists of a fleet of vehicles and set of customer requests. Each request specifies, at least an origin location and a destination location. The vehicles must travel through the locations so that each origin is visited before the corresponding destination. This simple problem structure can be used to model package delivery or public transit.

The early origins of PDP are unclear. Wilson was the first to study the problem in a rigorous manner [Wil70]. In his work, he studied a public transit dynamic request system, which accepted transportation requests from individuals and small groups. The customer specified the origin and destination locations, number of persons in the party, and the desired arrival time. Customers were assumed ready for departure at the time of the call. This problem was dynamic because requests for service arrived as the vehicles were enroute satisfying previous requests. The objective was to satisfy customer requests while minimizing customer disatisfaction with the service and operating cost.

The PDP has a number of variations, some of which are presented as follows. These characteristics are quite important to the design of an accurate and fast algorithm.

- **Size of Vehicle Fleet:** like all vehicle routing problems, the model and algorithm depend highly on whether the request are serviced by a single vehicle or multiple. For the Travelling Salesman Problem (TSP), a multiple vehicle problem can be simply transformed to find the optimum solution. In this case, the vehicle routes are concatenated to form a single long route. No such transformation is known for the PDP because each origin and destination must be visited on the same segment of the concatenated route. Techniques have been proposed to solve multiple vehicle problems by solving a sequence of single vehicle problems. For example, customers may be clustered by proximity then each cluster assigned to a single vehicle. Other solutions have used column generation techniques to optimize the assignment of customers to vehicles.

- **Static or Dynamic:** A static problem is specified before a solution is required. This situation corresponds to request service long enough in advance so that vehicles may be routed before departing the depot. No further requests for service are accepted after the vehicles are dispatched. A dynamic problem, on the other hand, allows customers to request service after the vehicles are dispatched. In this research, we consider service requests after the taxis are dispatched in a mobile environment depending on location of both MU and taxi. The optimization of time-varying system requires sophisticated algorithms to ensure proper service [Gei90].

- **Transshipment and vehicle transfers:** Some physical systems allow cargo or persons to transfer from one vehicle to another at an intermediate place. "Transshipment" complicates the routing process because they require interaction between two separate vehicles. The two vehicles must both visit a common location with the second arriving some time after the first. Although vehicle transfers are sometimes unavoidable, they are highly discouraged.

- **Full or Split Loads:** In some problems, the requirements of each request exactly fits the capacity of the vehicle. In this case, vehicles must pick up a load and immediately transport it to its destination before they can service other requests. For other problems, the load are larger than the capacity of any single vehicle and must be split to be accomodated. This situation arises, for example, when dealing with bulk cargo consisting of pieces which can easily be divided among vehicles. In this dissertation, by nature taxis are only allowed to carry one MU at a time. In practice this MU may be one person or a group of people, but we still treat it as a single object and ignore for simplicity's sake any unusual circunstances such as multiple drop off locations.

Chapter 3

Concepts

3.1 Introduction

In this chapter we review basic concepts. Our work encompasses the main area: Mobile Location-Based Service (LBS). This concept is described in the following sections. The definition and fundamentals of LBS will provide a common conceptual ground for further study into the requirements of LBS and will keep us focused within the objective of this dissertation.

3.2 Mobile Location-Based Service

The convergence of multiple technologies, including geographic information system (GIS), the internet, wireless communication, location methods, and portable devices, has given rise to new types of information utilities that may be referred as location services. Also called mobile location services, wireless location service, or location-based services, these systems allow MUs to use services based on their position or geographic location. The MU of the LBS discussed in this dissertation is assumed to be a typical user of an Internet enabled cellular telephone, wireless Personal Digital Assistant (PDA) or any other wireless device capable of supporting a presentation layer.

Specifically, LBS is used to define and provide services based in the geographic position of a mobile device. Often, a LBS is a service in real time that can use the defined profile of a subscriber to send the most adapted information at that moment and locale. In this way, a LBS could provide the geopersonalised information helping people with routes and transit conditions, locating gas stations or restaurants, roadside assistance, fleet management, load tracking and tracing of a stolen car, and a lot of others.

A more complete definition of LBS follows. These services address so many different types of industries, research centers and applications that it is useful to consider a range of definition [ESR00].

3.2.1 The Broadest Definition

An LBS, in the broadest sense, is any service or application that extends spatial information processing, or GIS capabilities, to users via the Internet and/or wireless network. Applications accessible via desktop computer linked to the Internet, such as those provided by "COMO VOU" (www.comovou.com.br) and the "MapLink" (www.maplink.com) are LBS. Call centers where telephone operators manually enter or automatically access a user's location could also be considered LBS. Examples include emergency response services such as E.911 [Bir00] centers or the American Automobile Association's Roadside Assistance Service.

3.2.2 The E.911 Definition

By mandate of the United States Federal Communications Commission (FCC), the geographic position of all cellular phone devices must be detected so that emergency services can be dispatched to the caller's location. Phase 1 of the E.911 mandate required carriers to identify within which cell site the user was located. Phase 2 of the FCC mandate requires that carriers provide a location within a 125 meter radius in at least 67% of the cases. Beyond the FCC mandate, phase 3 emergency services include LBS authentication methods providing more control to the user to protect privacy and allow them to select which services they want to receive.

3.2.3 The Pull Definition

Services that utilize geographic position of a wireless device to derive information related to that location enable users to pull information to them whenever it is needed. Type of pull services include:

- *Travel Directions* - I am here, how do I get there? These are some of the first location services to be offered via the wired Web.
- *Taxi Hailing* - I need a taxi now. A holder of a mobile device signals that taxi is needed. Taxi services operating in that territory can automatically pass on the MU's location and phone number either to a dispatcher or directly to the nearest taxi. MUs are contacted by a respondent informing them that a taxi is nearby and will soon arrive.
- *Mobile Yellow Pages* - Where is the nearest X? Users indicate the business categories they are interested in and pull up a set of listings in order of proximity to the user's location.
- *Buying Services* - Notify me when I am near a supplier that carries the specific item I am looking for. This type of service, referred to as mobile commerce (m-commerce), connects buyers to sellers. MUs entering a shopping area with an electronic list

including the brands, features, or model numbers of the products corresponding to these shopping list. In addition, MUs could also obtain directions to the store.

- *Instant Information* - These services enable users to point their wireless device at a facility and obtain information about it, either from a central database or from the facility itself via infrared or other type of wireless data transfer. This could be also implemented as a kind of push service.

3.2.4 The Push Definition

Push refers to an LBS that utilizes the position of a wireless device to qualify the holder as a recipient of a service. Some applications and services are described as follows.

- *Mobile Advertisements* - Users can receive on their wireless devices a lot of information, such as electronic coupons or other types of discounts and promotions.

- *Friend Finders* - Allow users to find the locations of their friends or family. For example, SMS. The service automatically notifies a user when a selected person (who also has a mobile device) is nearby or has entered into a specified area.

- *Zone Alerts* - Similar in functionality to friends finder services, zone alerts can indicate when a person or vehicle has entered or exited a specified region. Early implementations were used to track the movements of Alzheimer patients. The same type of service could be used to detect when an unwanted person enters within proximity of another person or place. For example, to track people under court restraining orders.

- *Traffic Alerts* - Real time traffic service provides users of status of predefined travel routes.

3.2.5 The Telemetry Definition

Machine to machine communication enables distributed assets to automatically notify service providers of their location and status. This applies to fixed assets such as vending machines; heating, ventilation, and air conditioner systems; copiers; alarm systems; or mobile assets such as trucks, cars, or packages. For example, a company that provides commercial air conditioner maintenance can receive wireless short burst messages from a specific unit that the Freon level is becoming so critically low that the unit is about to shut down. The origin of the message can be located and a nearby service technician can be dispatched.

3.3 The Evolution of Location Services

Unpublished research by Gravitate, identifies three generations of location services [Gra00]. First generation services require the user to manually input location in the form of a street address or postal code and are typically available to stationary desktop computers or mobile units. Examples of first generation LBS are MapQuest, CitySearch or other local information services.

Second generation LBS, which are available today, have the ability to determine rough locations typically at the postal code level. Basically, they are services that have the capability to extract some location information from the underlying network without requiring user entry. For example, zip code level location information is currently available to developers of PalmOS applications through the PalmNet data network. Though the level of granularity in position is high in a second generation LBS, when coupled with a user profile, it can increase the default relevancy of data displayed on a mobile device. For example, a mobile user of a city guide may log in from his wireless PalmVII to request information on movie playtimes.

Third generation location services are more location aware, taking advantage of more precise positional information, and have the capability to initiate services proactively based on location. These trigger mode services can notify the user of relevant events or conditions without the user's active participation, such as traffic alerts that meet the user's present preferences.

Gravitate identifies three types of triggers: Object, Object-temporal and affinity. Object triggers notify the user of the mobile device when entering within a predefined distance of a facility. Object-temporal triggers add the dimension of time. Affinity triggers allow one mobile device to know of the location of another mobile device.

3.4 Requirements to Enable LBS

Some requirements are needed to provide LBS to the users. The requirements are related to the mobile device, the carrier, the wireless protocol and the method of determining the MU's location.

1. **Mobile Device:** A mobile device consists of hardware and software, including an operating system and application. With this combination, users are able to access information services and entertainment through a mobile client-server architecture. Therefore, a mobile client-server system must adapt to different environments.

2. **Carrier:** A carrier is effectively an entity with a wireless network that is able to provide a service to users. The carriers could be specific, that is, they provide services in a given segment, act in given geographic region, answer requests coming from specific networks and enable queries to be sent in agreement with the user profile.

3. **Wireless Protocol:** Wireless protocols specify the way that a mobile device can be used for the exchange of information through Internet access, including Electronic-mail (E-mail), the World Wide Web (WWW), newsgroups and Internet Relay Chat (IRC). The networks available provide access to information services to users through the use of wireless protocols as well as providing users with information of their current location. Wireless Application Protocol (WAP) [WAP], Interactive Mode (I-Mode) [IMo] and Subscriber Identity Module Application Toolkit (SIMAT) [Sim] are some examples of wireless protocols that are currently available.

4. **The Methods of Determining the MU's Location:** With the use of wireless protocols, carriers currently provide services such as information and entertainment to users. Athough the users can be provided with required services, the services supplied are not be based on the users' location, as the network or the device does not, currently, provide the location of the users. However, users could be provided with access to specific information and services through of one or more technologies that determine the location of the users. The Federal Communications Commission (FCC) regulations differentiate such technologies between terminal-based (handset-based) and network-based methods. These methods are described in Section 3.5.

3.5 Survey of Locating Technologies

LBS are enabled in part by the technologies that physically locate the MU. These technologies are usually divided into two groups, terminal-based and network-based methods [Swe99] [And01] [HB01] [BP00]. Within each group there are several subgroups.

3.5.1 Terminal-Based Methods

Terminal-based positioning relates to the positioning intelligence that is stored in the terminal or its SIM (Subscriber Identity Module) card. These kinds of positional mechanisms require a new terminal, a new SIM card. In practice, this means that once the system has been installed, users will have to replace their device or SIM cards to benefit from it.

- **GPS** Global Positioning System [Kal97] [BHE], uses a set of satellites to locate a user's position. This system has been used in vehicle navigation systems as well as dedicated devices for some time. With GPS, the terminal gets positioning information from a number of satellites (usually 3 or 4). This information can then either be processed by terminal or sent to the network for processing, in order to generate the actual position.

  Even though the GPS receiver knows the precise location of the satellites in space, it still needs to know how far away the satellites are, so it can determine its position on earth. There is a simple formula that tells the receiver how far it is from each satellite: Your distance from a given satellite object equals the velocity of the transmitted signal multiplied by the time it takes the signal to reach you ($Velocity \star TravelTime = Distance$). In this work, we use the same principle to estimate the travel time of a busy taxi. This information has used for us, in particular, to divert a vehicle away from its current destination to service a new request that just occured in the vicinity of its current position.

The main drawback of GPS is that satellite signals are relatively weak and may not always provide adequate coverage to all environments (indoor environment). However, the GSM network can provide assistance information that gives integrated GPS receivers better coverage than stand-alone GPS receivers.

- **AGPS** Network Assisted GPS uses fixed GPS receivers that are placed at regular intervals, every 200km to 400km to fetch data that can complement the reading of the terminal. The assistance data makes it possible for the receiver to make timing measurements from the satellites without having to decode the actual message. This assistance greatly reduces the time needed for a GPS receiver to calculate the location. Without the assistance information the Time-to-First-Fix (TTFF) could be in the range 2045 seconds. With assistance data the TTFF could be in the range of 18 seconds. The assistance data is broadcast around once each 1 hour.

- **EOTD** Enhanced Observed Time Difference only uses software in the terminal. To run the EOTD algorithms in the idle mode (terminal is not handling a call) and in the dedicated mode (terminal is handling a call), new phones must be designed with additional processing power and memory. The EOTD procedure uses the data received from surrounding base stations to measure the difference it takes for the data to reach the terminal. That time difference is used to calculate where the MU is located relative to the base stations. This requires that the base station positions are known and that the data sent from different sites is synchronized. The most common way of synchronizing the base stations is via the use of fixed GPS receivers. The calculation can then either be done in the terminal or the network.

3.5.2 Network-Based Methods

Network-based positioning methods do not require positioning intelligence to be built into the mobile device. These methods involve triangulating the radio emission of the phone or using RF (Radio Frequency) multipath to identify the most likely position of the radiating source. There appears to be significant performance advantages to the multipath method over triangulation in urban environments. While less accurate than GPS and perhaps more expensive for the carrier to integrate, network based methods work on existing phones which means a faster access to a massive locatable user base.

- **CGITA** Cell Global Identity (CGI) uses the identity of each cell (coverage area of a base station) to locate the user. It is often complemented with the Timing Advance (TA) information. TA is the measured time between the start of radio frame and data burst. This information is already built into the network and the accuracy is decent when the cells are small. For services where proximity is the desired information, this is an inexpensive and useful method. It works with all existing terminals, which is an advantage. The accuracy is dependent on the cell size.

- **TOA** Uplink Time of Arrival works in a very similar way as EOTD, the difference being that the uplink data is measured (the data that is sent by the terminal) [TOA00]. The base stations measure the time of arrival of data from the terminal. This requires that at least three monitoring base stations are available to perform the measurements. The base stations note the time difference and combine it with absolute time readings using GPS absolute time clocks. EOTD and TOA might look very similar, but the key difference is that TOA supports existing terminals. This is an important aspect, as it will take time to convince all terminal manufactures to implement a change in their software. The drawback of TOA is that it requires monitoring equipment to be installed at virtually all of the base stations.

3.6 Location Services

Given the ubiquity of location information, the increasing mobility and the availability of broadband communications and mobile interfaces, the applications of LBS are extremely diverse. In this way, the LBS can be categorized by type of application [Swe99] [Pat00]. Figure 3.1 shows the wireless location-based service tree in evolution according to categories discussed to follow.

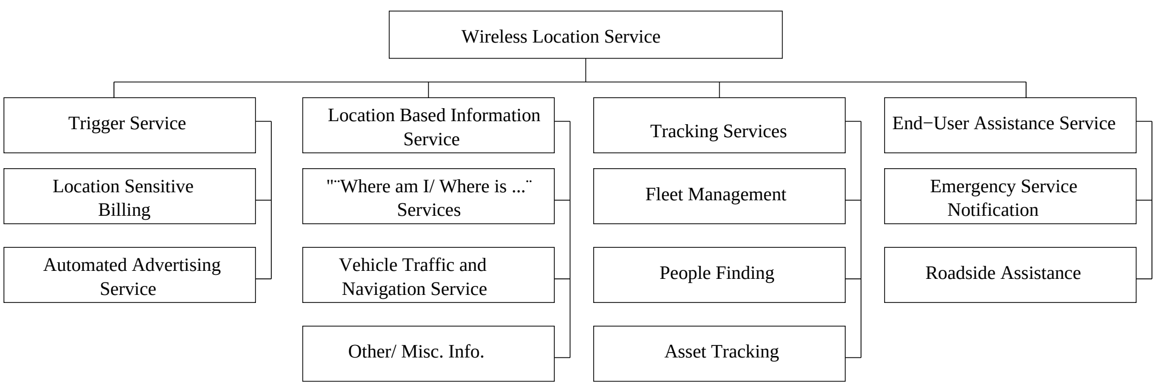


Figure 3.1: Wireless location services tree

- **Information services:** Information services make use of an information bank where information is filtered according to the relative position of a user and the applications he or she selected. Examples of information services include location-based yellow pages, events and attractions (for example, “What is happening today in town near here?”).

- **Tracking services:** Different services can use location-based information to track mobile device, to provide safety, to prevent robbery, to facilitate delivery, and so on. Examples of this kind of service include the tracking of a stolen car and giving an automobile repair shop the location of a motorist in need (out of gas, dead battery).

- **Resource management:** Resource management applications are used to manage the logistics of vehicle fleets. Examples of resource management include taxi fleet management, the administration of container goods and resource allocation.

- **Navigation:** Navigation applications are used to inform MUs how they can best move from point X to point Y. Application of this kind can be adapted to vehicle or pedestrian navigation.

In Gravitate [Gra00] the LBS are joined in four categories similar to the categories describe above. The following description of the categories shows that they convey to the same idea as those above.

- **Locators:** This category provides the basis for tracking, dispatch , and fleet management systems. For many years, trucks and emergency vehicles have been tracked manually by having the drivers radio their locations. Comparing a vehicle's position in relation to an origin or destination provides dispatchers with information about the status of a fleet of vehicles. With automated location systems, it is possible to track any type of vehicle or mobile object. Emergency response personnel can find victims of accidents, builders can track construction equipment, delivery companies can track packages, and police can find stolen property.

- **Proximity search:** Once geographic location is established, the next question to ask is, Where are the nearest facilities to my position? Electronic yellow pages and other directory services utilize proximity search algorithms to find the nearest business listings or other facilities within a specified radius of a location. Proximity searches can take many forms. Simple and sophisticated searches are supported by GIS software, which can be resident on a handheld client device or accessed remotely. The common example cited for a basic offering is a consumer oriented service that enables users to find the nearest restaurant by type of cuisine.

- **Travel directions:** These applications use a high quality geographic database, which combine multiple map area files into a seamless map coverage that enables routing across multiple geographic data files.

- **Real-Time traffic conditions:** Audio text delivery via telephone allows users to obtain reports for selected roads. This means people are able to dynamically evaluate a planned itinerary and, if needed select an alternative route to avoid problems.

Various applications described below are already in development, which utilize either cellular architecture or a specifc architecture that makes use of a mobile location center

(MLC). In the case of this dissertation, the application being studied here, it can be categorized either in the resource management category specified in [Swe99] and [Pat00] or in the locator and travel direction category specified in [Gra00]. In addition, it uses both cellular communication infrastructure and GPS to locate MUs and taxis.

3.7 LBS Architecture

The architecture being adopted today by many network operators is based upon MLC. The MLC separates the location technology to locate mobile devices from the applications, that the location information will be applied (see Figure 3.2).

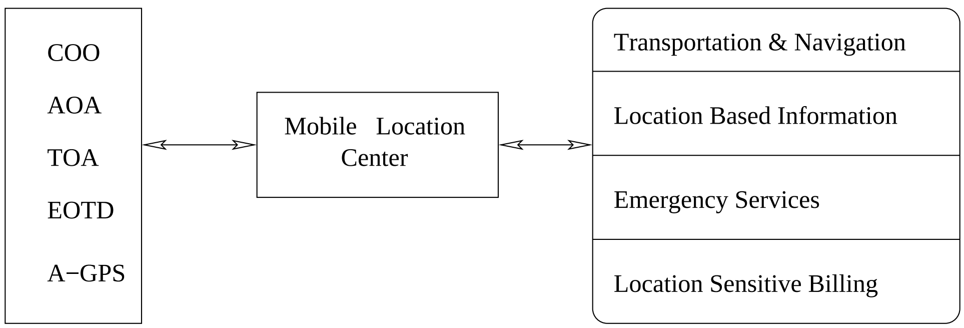


Figure 3.2: The architecture based upon Mobile Location Center

Ericsson's mobile positioning system, described in [Swe99], consists of tree subsystems:

1. The positioning subsystem can use a variety of techniques to determine geographic coordinates, such as TOA, EOTD, and AGPS.

2. The positioning gateway subsystem (MPC - Mobile Positioning Center), which functions as a mediation device between the public land mobile network (PLMN) and the location service client (LCS[1]-client), retrieves data from positioning subsystems and convert it into positioning information for the LCS-client. It still provides the operator with a GUI (Graphic User Interface based on Java platform) to manage the node. In the proposed Ericsson solution, this node is called the mobile positioning center tool (MPC-tool).

3. The LCS-client, which is a subsystem of the MLS (Mobile Location Solution, see Figure 3.3), contains applications that make use of positioning information. Internal applications are coded into the GSM system according to the GSM standard. External application are supplied to the system by system vendors and the operator.

[1] LoCation Service

In addition, the MPC monitors and registers the use of specific location applications, thereby allowing operators to charge for them via applications installed in the MPC or via interface to the billing system. This is not the case when a traditional global positioning system is used. In this system, the network solely serves as a link for transporting positioning data between the terminal and the central application management site. Therefore, operator or service providers cannot charge specifically for their services. Figure 3.3 shows the architecture of Ericsson's mobile positioning system. This architecture can work with various location methods (GPS, Cellular PS (Cellular Positioning System), etc.), depending on the application requirement.

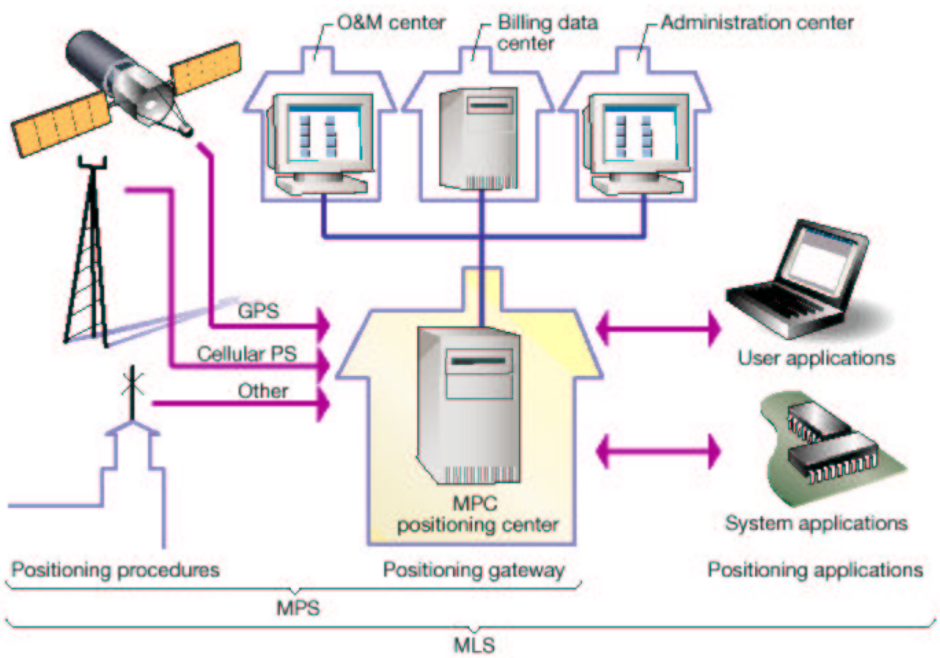


Figure 3.3: Ericsson's mobile location solution (MLS)

In contrast to 2G (second Generation) systems, 3G standardisation is not focusing on services, but on a service network architecture. Current mobile communications systems offer no standardised service platform, which would decrease service development costs. Access to network elements, like call state control function (CSCF) or home location register (HLR), is only possible within the network operator domain, by using mobile network specific concepts [Cam99] [Map99]. UMTS (Universal Mobile Telecommunications System) have been working to offer a environment for location-based multimedia services.

In [PKK01] is described a localisation architecture (LoL@ - Local Location Assistant), which uses the packet switched domain of UMTS. Its proposes an architecture for location-based multimedia services, including advanced user interface concepts with speech recognition and central user preference management. In addition, localisation and navigation is implemented in a layered structure. Basic localisation functionality is provided by service capability with a standardised interface (OSA (Open Service Architecture)/Parlay) [Van00] [3rd00a], enabling location-based services. Figure 3.4 shows LoL@ localisation architecture.

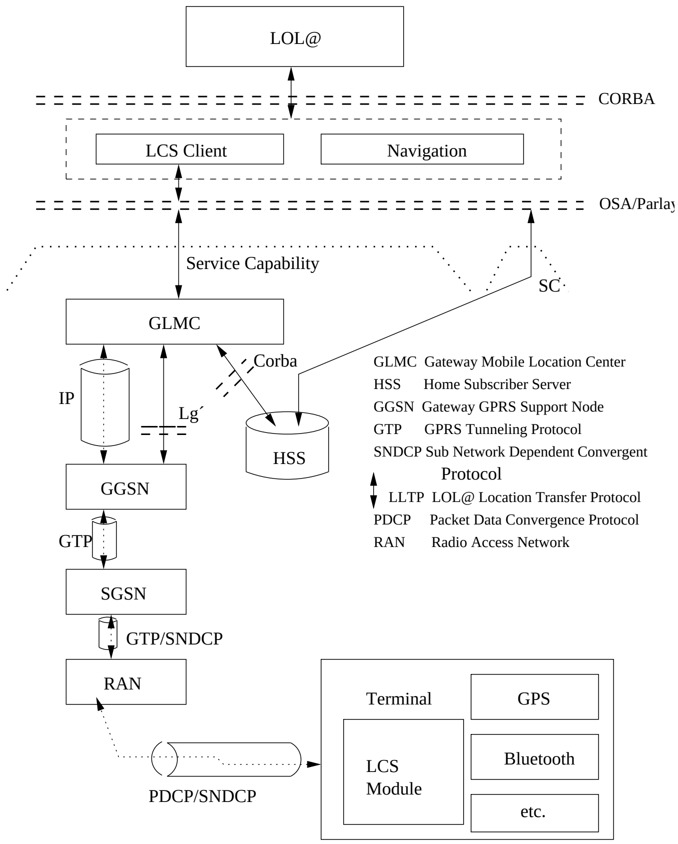
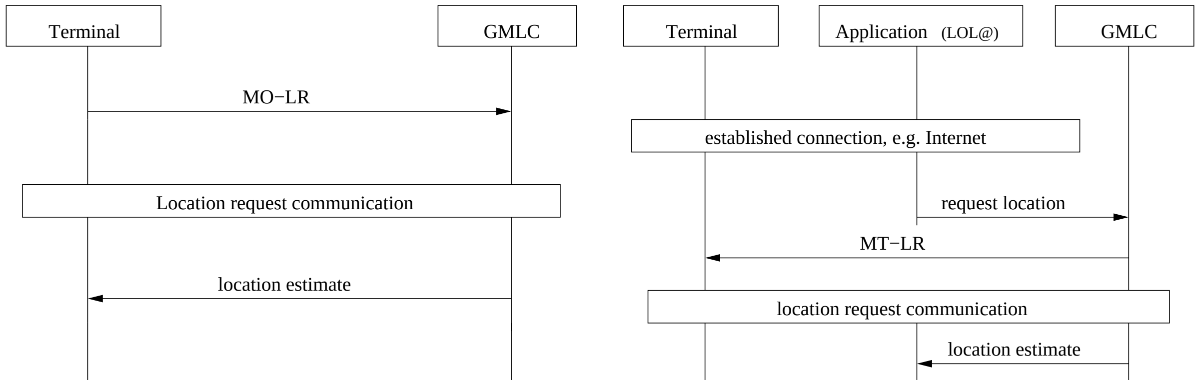


Figure 3.4: LoL@ localisation architecture

LCS enables networks provide location information in the SGSN (Sub GPRS Support Node), which is accessed by the GMLC via the standardised Lg-interface. Legancy networks are covered with an end-to-end LCS concept, extending 3GPP (3rd Generation Partnership Project) standardisation. A terminal LCS module communicates with the GLMC, using the IP-based LoL@ Location Transfer Protocol (LLTP). The terminal LCS module can see as remote part of the GMLC.

A location-based service requests the location information of a terminal. Location information may be requested directly by the terminal via the GLMC (MO-LR : Mobile Originated Location Request, see Figure 3.5(a)) or by the application Kernel residing in the server domain. In the latter case, the terminal has a connection established to this application (e.g. via internet, protocols, TCP/IP or UDP/IP). The application acts as LCS Client and requests the location (MT-LR: Mobile Terminated Location Request, see Figure 3.5(b)).



(a) Mobile originated location request (b) Mobile terminated location request

Figure 3.5: Location request possibilities

A description of cellular mobile telephone system architecture can be found in [Tei95] [Fre89]. We can state that a cellular mobile telephone system is the telecommunication system that provides the cellular mobile service, and it is the telephony service with the feature of providing to the user full mobility during the establishment of a telephone call. Figure 3.6 shows the basic elements of a simplified cellular network.

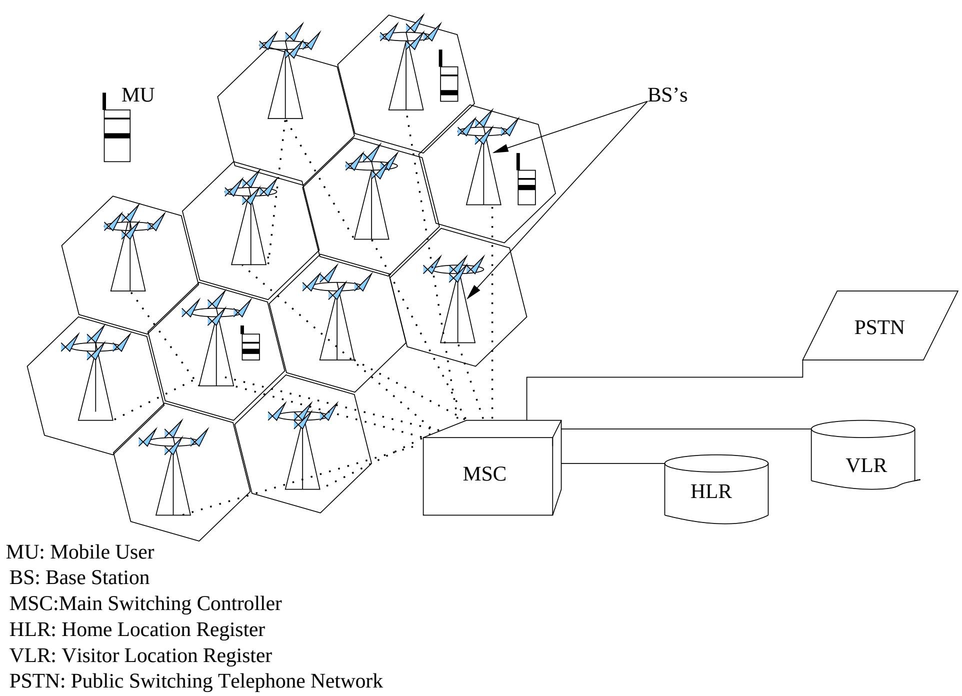


Figure 3.6: Architecture of the mobile cellular networks

- **Mobile Unit:** is a mobile device, such as cellular phone, Personal Digital Assistant (PDA), etc.

- **Base Station:** This contains a receiver which is controlled by the central processing server in order to locate a specific mobile unit.

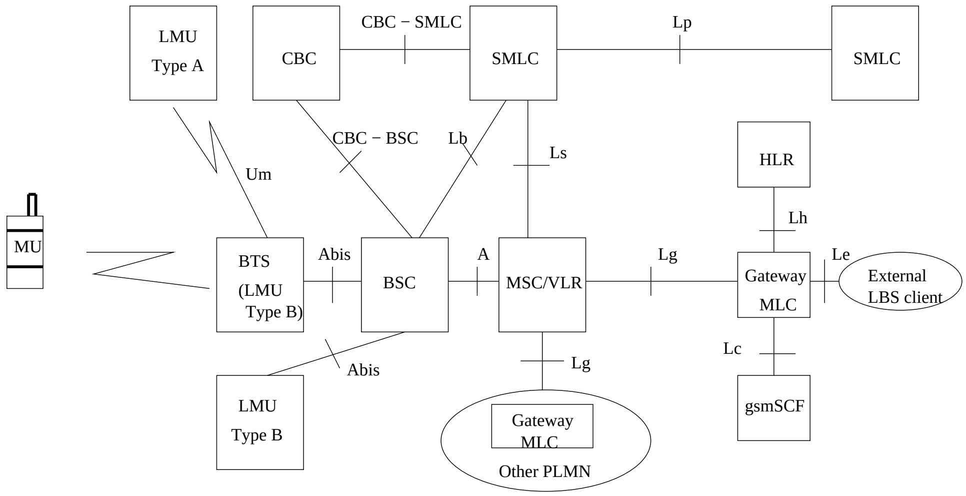
- **Main Switching Controller (MSC):** It is important in a mobile telephony system in two aspects: first, because the MSC concentrate all decisions in the context of the MSC and second because it is the interface of the MSC with the public switching telephone network (PSTN). Each MSC is responsible for a set of BSs and each BS is connected a single MSC.

- **Home Location Register (HLR):** It is a database of a service area that contains information about the registered MUs in that area. There are a lot of HLRs depending on system architecture. Some Information is stored in the HLRs such as, the mobile identification number (MIN), electronic serial number (ESN), user profile with data about billing, call forward to number, state (enable or unable), and a pointer to the last visitor location register (VLR) where the MU registered.

- **Visitor Location Register (VLR):** Working in conjunction with the HLRs to support automatic roaming. They are local and temporary databases with data of MU that is out of its service area. When a roamer subscriber registers in a service area, its profile is copied to the local VLR. This procedure reduces the signaling traffic between the service areas and decreases time needed to establish a call.

The North American T1P1.5 GSM location services working group is defining requirements for generic location services architecture [Hay00] [3rd00b]. This architecture is designed to support wireless E.911 as an initial application as well as other applications such as location sensitive billing, local information delivery and fleet tracking. New network nodes have been defined as shown in the Figure 3.7. The gateway mobile location center (GMLC) is the interface point for location applications. The serving mobile location center (SMLC) is responsible for requesting that a mobile unit be positioned. It may in some cases maintain information on network topology to allow efficient positioning of the mobile unit. There are studies that consider additional functionalities to support LBS, which will be integrated into existing nodes (MSC/VLR, BSC - Base Station Controller, HLR, etc.).

A Public Land Mobile Network (PLMN) is established and operated by an administration or Recognized Private Operating Agency (RPOA) for the specific purpose of providing land mobile telecommunications services to the public. A PLMN may be regarded as an extension of a network (e.g. ISDN - Integrated Services Digital Network); it is a colletion of MSCs areas within a common numbering plan (e.g. same National Destination Code) and a common routing plan. Basically, the MSCs are the functional interfaces between the fixed networks and a PLMN for call set-up. The implementation of the mobile service with international roaming implies the exchange of data between the equipment involved in the service. The same No. 7 signalling network should be used to transfer these data and the



CBC − Cell Broadcast Centre
gsmSCF − GSM Service Control Function
PLMN −Public Land Mobile Network
LMU − Location Measurement Unit
BTS − Base Transceiver Station

Figure 3.7: GSM network topology for LBS

call-related signalling information. In this case, the PLMN defines some basic interfaces for passing information back and forth. The main interfaces of GSM network topology for LBS are described as follows.

- **Um-interface** interface between mobile station and base station system.
- **A-interface** interface between the MSC and BSC. It is used to carry information concerning: Radio Network System (RNS) management, call handling and mobility management.
- **Abis-interface** interface between BSC and BTS.
- **Lg-interface** interface between MSC and GMLC. It is used to exchange data needed by the MSC to perform subscriber authorization and allocate network resources.
- **Lb-interface** interface between BSC and SMLC. In UMTS, it is not standardized.
- **Ls-interface** interface between SMLC and MSC/VLR. In GSM, SMLC supports positioning of a target MS (Mobile Station) via signaling on the Ls interface to the visited MSC.
- **Lp-interface** interface between peer SMLCs. In GSM, a BSS-based (Base Station System) SMLCs may support the Lp interface to enable access to information and resources owned by another SMLC.

- **Lh-interface** interface between GMLC and HLR. This interface is used by the GMLC to retrieve the VMSC (Visited MSC) location and IMSI (International Mobile Station Identity) for particular mobile.

- **Le-interface** This interface connects the PLMN to the external LCS client.

All the positioning methods rely on very precise timing measurements. The location measurement units (LMUs) provide reference information that is combined with the received measurement data or broadcast to assist in the calculations. There are two types of LMUs depending on whether they are signalled using A-bis interface[2] or over the air interface.

There is currently no unifying architecture for LBS. A number of major gaps exist between the capability to locate a MU and the software requiriments.

[2]Interface between BSC (Base Station Controller) and BTS (Base Transceiver Stations): when the BSS (Base Station System) consists of a BSC and one or more BTS, this interface is used between the BSC and BTS to support the services offered to the GSM user and subscribers. The interface also allows control of the radio equipment and radio frequency allocation in the BTS. The interface is specified in the 08.5x-series of GSM Technical Specifications and it is considered an interface internal to the access network.

Chapter 4

The Taxi Assignment Problem

4.1 Introduction

This chapter begins with the definition of the problem reported in this dissertation. The taxi service current approach is discussed in Section 4.2 and Section 4.3 describes the new proposal approach. The definition and contextualizing of the resource assignment problem is presented in Section 4.4. Section 4.5 presents the mathematical formulation for the taxi assignment problem. In the following sections we describe the assignment algorithm, which is used by the LBTS simulator to meet the challenge of which taxi will be assigned to transport each MU to his destination.

The problem being considered in this dissertation is defined around a situation in which a number of Mobile Users (MUs) appear at different locations randomly over time, and these MUs need to be transported to different destinations. There is a fleet of taxis that performs the transport of these MUs. Given this situation, the problem at hand is deciding which taxi should be assigned to transport each Mobile User (MU), and doing so in a way that minimizes the distance traveled and time of both MU and taxi.

4.2 Current Approach

The pickup and delivery problem is quite common in today's society in a number of forms. One example is the use of taxis in large cities to transport people. Courier services, like the Post Office and Federal Express [fed] is another. Even the problem of data packets being routed through the Internet can be viewed as an instance of this problem.

The taxi company has been around for a long time and especially in the EUA and Brazil tends to be very diverse. It is not only fragmented in terms of companies but also with respect to business plans. Some taxi companies own and operate all of their vehicles and use drivers as employees. Other companies own the vehicles but lease them to drivers for a percentage of what they bring in. In other cases the taxi companies do not own any vehicles but act purely as a fee-based dispatching service. Finally, some taxi drivers own their own taxi and simply ride around looking for passengers without the use of a dispatch system. One effect of this diversity in the taxi company is that it has been slow to innovate or to adopt new technology. In Brazil this is clearly evident when we compare it with United States and Singapore [Zhu01].

Most taxi services still operate through a radio-based central, or specific dispatch system. Basically, the current strategy to assign a taxi to an MU is based on a queue-type structure (see Figure 4.1). In this case, when a taxi arrives at a determined region, it sends the information of what region it is in to radio-based central. This taxi driver is put in a queue with priority FIFO (First In First Out). Therefore, when a user requests taxi service, he sends an address information to a receptionist. Then, he identifies the region where the user is and the taxi queue associated it. In this way, the receptionist calls the first taxi driver and verifies whether he could serve the request. The driver is allowed to accept or reject the request. If the driver rejects it, the driver is kept in queue for the next available request, and the request is offered to the next available driver. If the taxi driver can fill the request then he is assigned to the user, if not the receptionist verifys the next taxi driver in the queue until he reaches a taxi that can fill the request. If no taxi is found, the receptionist will begin searching nearby regions for available taxis. The algorithm used to scan other regions is subject to control by the dispatch manager and varies from region to region.

4.3 New Approach

The approach that we propose is the use of an assignment algorithm in a mobile environment. The MUs carries any wireless device to request taxi service and the assignment process determines which taxi should be assigned to MUs. Specifically, this system works as follows. When a new MU arrives and need to be transported he will send a Request For Taxi Service (RFTS). A gateway is used to take the RFTS and to obtain the location of the MU and taxis by GPS (Global Positioning System). After the locations of both are obtained, the gateway searches the shortest distance between them. The search of shortest distance is performed through the shortest distance and routes matrices had previously generated by the Floyd-Warshall [AH00] [CLR90] algorithm. Then, the assignment algorithm is run to generate an assignment matrix. Soon afterward, the gateway composes Data Packages (DPs), which are divided into two groups: Mobile User Data Package (MUDP) and Taxi Data Package (TDP). A MUDP can contain a set of parameters that represents the answer package of a determined request, such as vehicle information, pickup time, etc. On the other hand, a TDP can contain a set of parameters that allows the taxi to perform the transport of MU, such as MU information and the route to be followed. When an MU receives its MUDP, it sends a positive acknowledgement to the gateway. At this point the taxi begins to execute its task (to pickup and to transport the MU to his destination) and the assignment process is over. This process is repeated after a determined interval of time. This interval of time is called time window and it is a parameter to be specified in

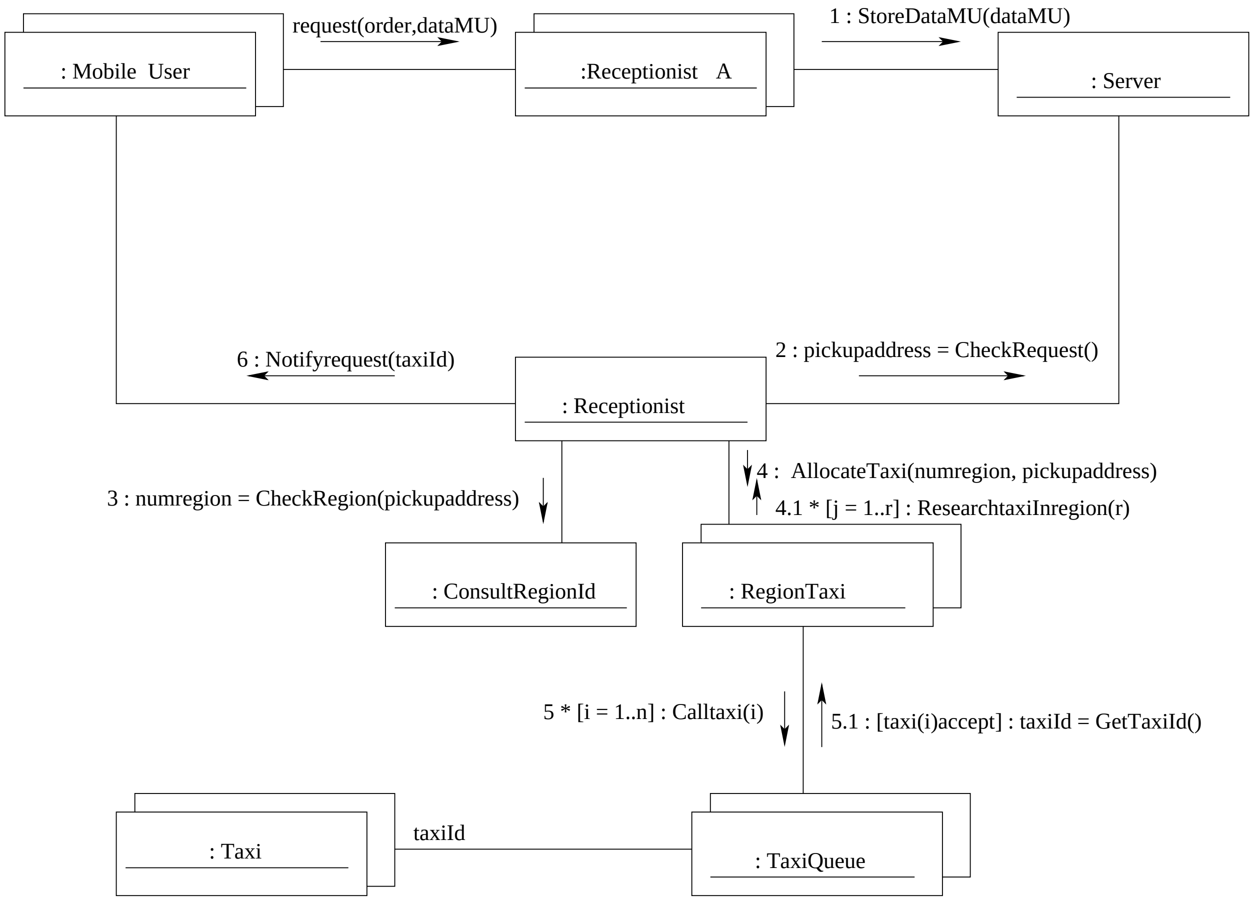


Figure 4.1: Collaboration diagram for current approach

the simulator. Figure 4.2 shows the assignment process.

This approach is different from currently used approaches, which were previously discussed, in that it uses an assignment algorithm to assign the nearest taxi to the MU instead of a queue strategy. In addition a network optimization approach is used to solve the assignment problem.

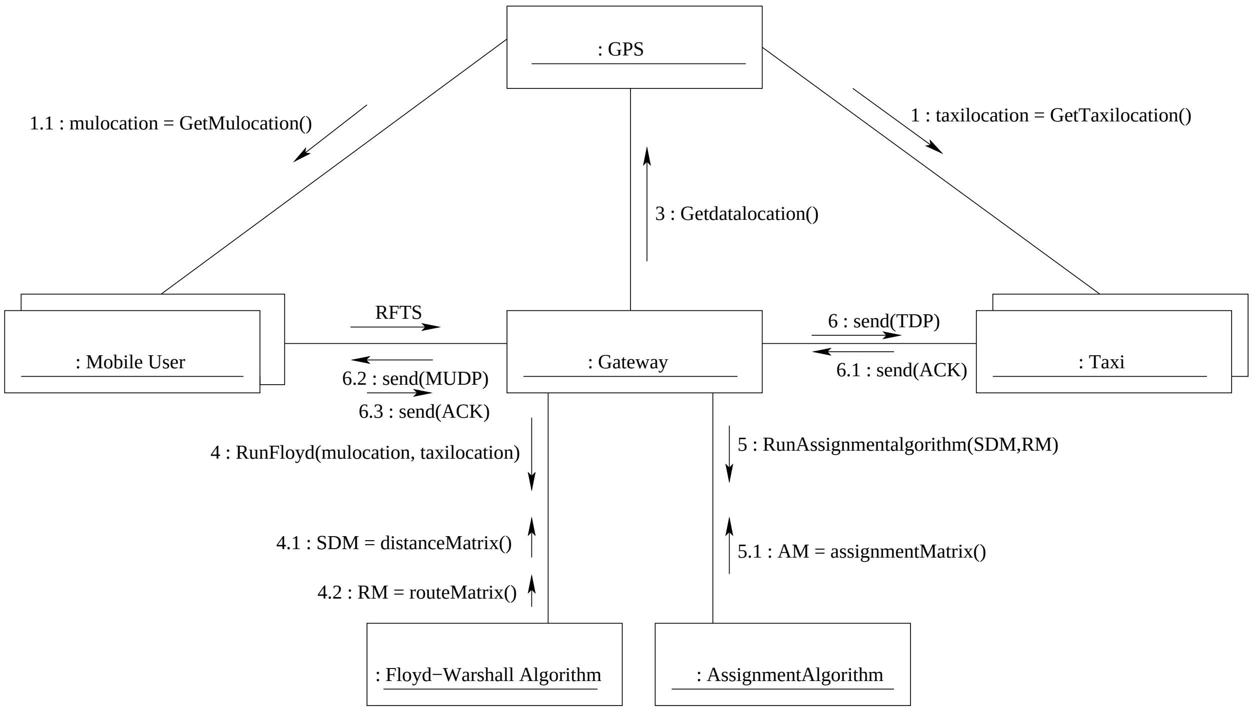


Figure 4.2: Collaboration diagram for assignment process with the use of a gateway in the mobile environment

4.4 Resource Assignment Problem

The resource assignment problem is generally formulated as shows Model 4.1 [IK88].

$$\begin{aligned} &\textit{minimize} \quad f\ (x_1,\ x_2, \dots,\ x_n) \\ &\textit{subject to} \\ &\qquad\qquad \sum_{j=1}^{n} x_j = N, \\ &\qquad\qquad x_j \geq 0, \qquad j = 1, 2, \dots, n. \end{aligned} \tag{4.1}$$

That is, given one type of resource whose total amount is equal to N, we want to allocate it to n activities so that the objective value $f(x_1, x_2, \dots, x_n)$ becomes as small as possible. The objective value may be interpreted as the cost or loss, or the profit or income, incurred by the resulting assignment. In the case of profit or income, it is natural to maximize f, and we shall sometimes consider maximization problems. The difference between maximization and minimization is not essential, however, because maximizing f is equal to minimizing $-f$. The variable x_j in 4.1 represents the amount of resources allocated to activity j. If the resource is divisible, x_j is a continuous variable that can take any not negative real value. The resource may not be divisible, however, if it represents persons, processors, trucks, taxis, and so forth. In this case, the variable x_j is a discrete variable that takes positive integer values, and the constraint shown in the Model 4.2 is added to the Model 4.1.

$$x_j \quad integer, \qquad\qquad j = 1, 2, \ldots, n. \tag{4.2}$$

Lower bounds and/or upper bounds on variables x_j are often imposed. In other words, it is sometimes necessary to allocate at least l_j but not more than u_j to activity j. These constraints are described by Equation 4.3.

$$l_j \leq x_j \leq u_j, \qquad\qquad j = 1, 2, \ldots, n. \tag{4.3}$$

4.5 Taxi Assignment Problem Model

We rarely consider the objective function $f(x_1, x_2, \ldots, x_n)$ in its general form as shown in Model 4.1, because it usually has some special structures specific to the intended applications. Specifically, in this dissertation we handle the taxi assignment to MU in a mobile environment. This problem can be described as follow. A firm owns a taxi fleet $T = 1, 2, \ldots, t$. A specific taxi is represented by $t_i \in T$. At the starts, there is a set of MU requests $M = 1, 2, \ldots, m$, that need to be transported by the fleet of taxi. Each taxi t_i has capacity 1, that is, a taxi only transports a single MU. For each request $j \in M$ has a pickup and a destination. The problem is to assign the nearest taxi i to the MU j.

The research reported in this dissertation determines which taxi should be assigned to which MU, based on optimization methods that minimize an objective function, subject to constraints. However, different functions have been proposed in the literature to capture the aims of each particular problem. Such functions are approximations of what is desired in the real word and they typically differ from one application to another. However, a common structure is often observed: a weighted sum of a variety of components related to the operation cost of the dispatching system (total distance traveled, capacity utilization, throughput) and service quality to the customers (waiting time). That is, the objective can take many forms, but is typically aimed at achieving a good trade-off between operation costs and customer satisfaction (time window compliance).

The objective function used throughout this work is to minimize the total distance travelled and total travel time of both taxi and MU. The resulting function is shown in Model 4.4.

$$\sum_{i=1}^{n} \sum_{j=1}^{m} d_{i,j} X_{i,j} \tag{4.4}$$

Where $d_{i,j}$ is the shortest distance of the taxi i to MU j and $X_{i,j}$ is the assignment variable, such that:

$$X_{i,j} \begin{cases} 1 & \text{if taxi ``i''is assigned to MU ``j'';} \\ 0 & \text{otherwise.} \end{cases} \tag{4.5}$$

Then the mathematical model of the taxi assignment problem can be given as follows.

$$Min \sum_{i=1}^{n} \sum_{j=1}^{m} d_{i,j} X_{i,j} \tag{4.6}$$

subject to

$$\sum_{i=1}^{n} X_{i,j} = 1 \qquad \text{j} \in \{1,2, \ldots ,\text{m}\} \tag{4.7}$$

$$\sum_{j=1}^{m} X_{i,j} \leq 1 \qquad \text{i} \in \{1,2, \ldots ,\text{n}\} \tag{4.8}$$

$$X_{i,j} \in \{0,1\} \tag{4.9}$$

Constraints 4.7 guarantees that each MU must be assigned to exactly one taxi. Constraints 4.8 guarantee that each taxi is assigned to no more than one MU.

Linear programming often provides a concise formulation for an optimization problem, as in this case. Recalling that each required assignment $(i \rightarrow j)$ contributes 1 to $X_{i,j}$ we have an integer linear programming problem.

Therefore, the model matrix associated to the set of constraints meets the unimodularity property. In this case, the optimal solution for the linear formulation is also integer.

4.6 Taxi Assignment Algorithm Description

We shall briefly explain here how we solve the taxi assignment problem based on the mathematical formulation described in Section 4.5.

Our problem is originated with local area (Belo Horizonte) taxi services. In this case, we choose a region of Belo Horizonte which is represented by a topographical map, which includes the road map. It is clear that a directed graph is the data structure able to represent efficiently the structure of road map. For each intersection between streets or avenues a node is created and connections between them enable definitions of edges. One-way or two-way streets are represented by a single or double directed edge. Such a graph is a collection of edges $(label, (i, j), d, v)$, where "label" is an identifier for the edge, (i, j) is a directed edge (from node i to j), and "d" and "v" are respectively the distance and speed associated with the edge.

The taxi assignment algorithm receives three parameters: the set of MUs that make service requests in the last time window, the set of taxis that should transport the MUs and the corresponding shortest distance matrix. The result of the assignment algorithm tells which taxi to assign to each MU. In spite of, the assignment problem is considering all the taxi requests, it can happen that in a determined time window we have more MU requests than the number of taxis available. In this case a set of MUs could not be served.

4.6.1 Find Shortest Distance Matrix

The Floyd-Warshall algorithm [AMOT90] can efficiently find and record all shortest paths at the same time as establishing the distance, d, by building a matrix D such that the first edge of the least distance path ($u \to v$) from node u to node v is $D_{u,v}$ (subsequent edges along the path are found similarly; if $D_{u,v}$ is an edge (u, u') to u'≠v then $D_{u',v}$ is the next edge to take towards v, and so on.). In addtion, the algorithm generates a second matrix, describing the step-by-step path (routes matrix). The algorithm runs in $O(|V|^3)$ time.

Given the graph of the road map, we use the Floyd-Warshall algorithm to generate the shortest distance matrix and routes matrix.

4.6.2 Algorithm Description

Initial datum is the shortest distance matrix, which is computed by the Floyd-Warshall algorithm. Such that $d_{t,m} \in D$, where $t \in T$ (set of taxis) and $m \in M$ (set of MUs). Since the mathematical formulation to taxi assignment problem is an integer linear programming problem, we decide to solve the problem according to a well known numerical procedure called the Simplex method [SHH92]. In this way, the LBTS simulator described in the next chapter, has used the GLPK (GNU Linear Programming Kit) [Mak01a] (see Appendix A) to process the mathematical model (taxi assignment algorithm). Specifically, we use GLPK/L [Mak01b] to write the mathematical programming model (see Appendix A), as shown in Figure 4.3.

Both Floyd-Warshall and the taxi assignment algorithm are integrated in the simulation environment, such that the taxi assignment algorithm is run every time that the time window is reached. Where the time window is a parameter configurable by the user interface. Consequently, every time that the time window is reached the standard model is modified regarding the number of variables and the distance values. This happens because the assignment process depends on the location of both the taxi and MU, which are objects that have mobility and also depends on the free or occupied state of the taxi.

The time is an important datum due to the fact that it will determine satisfaction of the MU. Let L be the length of the time window. At each time window n, there is a time interval $[l_n, u_n]$, where l_n and u_n are the lower and upper bounds of the desired service. Therefore, the time interval is $[n * L - L, n * L - 1]$.

The new approach to taxi service reported in this dissertation can be described by the following algorithm:

1. begin simulation
2. inicialize max-mobile-users[1], stop time, time window, max-taxis[2], simulation clock
3. load the graph of road network;

[1]Number of mobile user that we intend to simulate.

[2]Number of taxis that we intend to simulate.

```
model TAXI ;   /* taxi  assignment  algorithm*/
sets   I      /* taxis */              = ( T1, T2, ..., Tt ) ;
       J      /* mobile user */  =  ( UM1,  UM2,  ..., UMm );

parameter D[I, J]    /* distance between taxi and mobile users */      ;

D[I, J]   :=  table (i  in  I, j  in  J  :
                 UM1          UM2        . . .   UMm :
          T1      d11            d12                 d1m
          T2      d21            d22                 d2m
           .
           :
          Tt       dt1            dt2                 dtm         );

variables
       X[I, J]     /* assignment variable 1 or 0 */   ;

constraints
      Z            /* total distance traveled */     ,
      taxi[I]     /*taxi  constraints*/  ,
      user[J]   /*mobile  user  constraints*/ ;

Z := sum ((i, j), D[i, j]*X[i, j]);
taxi[i]    := sum(j, X[i, j]) = 1;
user[j]   :=  sum(i, X[i,j]) <= 1;

minimize   Z;
end.
```

Figure 4.3: Taxi assignment algorithm model description

4. load the shortest distance and routes matrices[3];
5. while *number of taxi* $<$ *max-taxis* do
   (i) create taxi;
   (ii) update the number of taxi;
6. end while
7. while *simulation clock* $<$ *stop time* do
   (a) if *number of request* $<$ *max-mobile-users* and *next arrival event* then
      i. create new mobile users;
      ii. start the generated mobile user process;
      iii. update the number of request;
   (b) if *simulation clock* $=$ *time window* then
      i. get taxis and requests information;
      ii. create the corresponding shortest distance matrix;

[3]These matrices have been generated in a previous step before starting simulation using the Floyd-Warshall algorithm

iii. run taxi assignment algorithm;
   A. create new GLPK/L model description;
   B. run GLPK (that is, obtain a numerical solution for the model);
   C. create an assignment matrix;
iv. assign taxis to mobile users;
v. assign routes to each taxi;

8. end while

9. generate taxi movement;

10. create output file;

11. end simulation

Thus, each new request is stored until the next time window is reached. When this happens (step 7b), the free taxis and busy taxis that should perform dropoff before reaching the lower bound of the next time window are included in the solution of the assignment algorithm. The algorithm takes into account the concept of multithreaded programming[4]. In this case, in the step 7b while the assignment algorithm is running, new requests can arrive and the taxi movements continue normally.

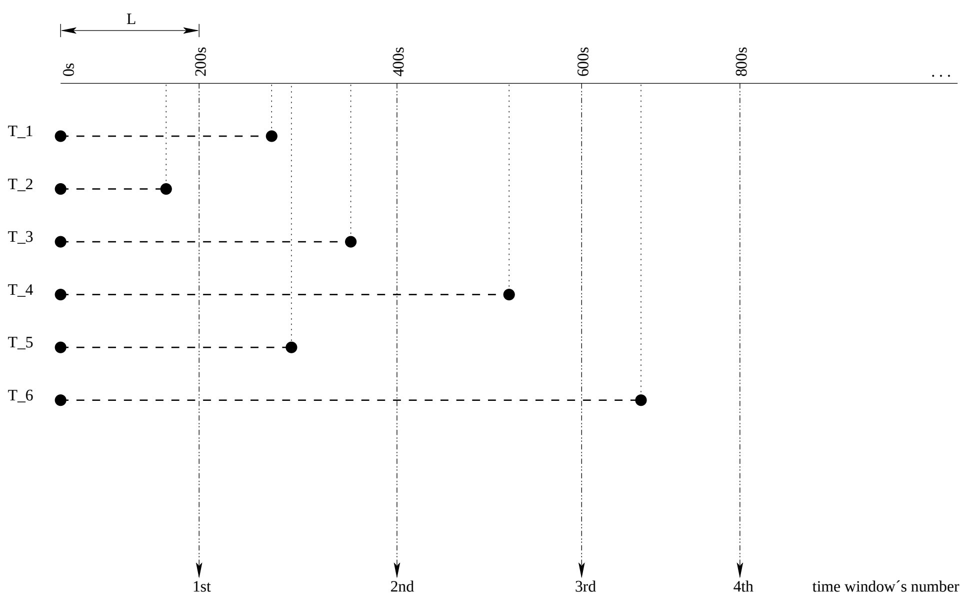


Figure 4.4: Assignment process

[4]A multithreaded program contains two or more parts that can run concurrently. Each part of such a program is called a thread and each thread defines a separate path of execution.

We already know that the assignment process takes into account the free and busy taxis. But it is still interesting to see how the assignment process happens. Suppose we have 6 taxis ($T_1, T_2, T_3, T_4, T_5, T_6$) as shown in the Figure 4.4, and let time window lenght $L = 200$ seconds. When the first time window is reached (200 seconds), the set of free taxis are included in the algorithm solution. According Figure 4.4, the taxi T_2 should be included in the solution, because it had been serving a MU and it had ended before $200s$ (1st time window). On the other hand, the busy taxis that should end their job before the next time window (400 seconds or 2nd time window) should be also included in the solution. Therefore, the algorithm solution should include the taxis T_1, T_2, T_3, T_5 for the first time window.

In the case that the number of MUs is different from the number of taxis, the algorithm generates either virtual MUs or virtual taxis. This is necessary because the standard Simplex method requires that the number of MUs must be equal to the number of taxis. In this way the algorithm assigns distance equal zero to these virtual variables, which are also called artificial variables. Thus, they are not considered in the assignment matrix. Therefore, in a given solution it can happen that MUs may not be served.

When the assignment matrix is completed, the algorithm queries each taxi, belonging to the assignment matrix, to get the best route from the route matrix that it should follow to pick up the MU at his origin and transport him to his destination. The query to the route matrix returns a set of nodes. Then, it gives a set of nodes for each taxi.

Chapter 5

The Simulator

A simulator was developed in this work in order to produce different operating environments that reflect as closely as possible what is observed in the real world. This chapter begins with the definition of the importance of a simulation environment and the motivation for using the Java programming language to build the simulator. Following this, the different aspects and functioning of the simulator are described.

5.1 Introduction

Computer simulation is used to model a physical system, to execute the model, and to analyze the results of the execution [Fis96].

Simulation is important mainly because "simulation can tell us things we do not already know" [Sim69] and can help us to make decisions and recommendations. It is expensive and sometimes impossible to experiment on real systems. Therefore, simulation can assist us to make decisions by modeling real systems and by analyzing the simulation results [Gor78] [Sei88].

The emergence of Java programming language together with the Internet makes us possible develop dynamic simulation models more interactive and more widely available. The reasons to use Java to build a simulation environment have been discussed in several recent papers [BS96b] [Fis96] [NMZ96] [MRZZ97] [Zha97a].

1. Simulation models implemented as Java applets can be downloaded and executed on local machines with Java-enabled web browsers.

2. Java is capable of developing sophisticated animations using its built-in Thread, Applet and AWT classes. Java applets running on Java enabled web browsers can combine text, sound, graphics and images to provide a dynamic effect.

3. Java is an object-oriented programming language, small and easy to learn. It has a high degree of code re-usability.

5.2 LBTS Simulator

The LBTS or Mobile Location-Based Taxi Service simulator developed in this work allows us simulate both current and new proposal approach, where the main scenario is taxi service. This simulator is responsible for:

- generating new requests according to statistical distribution,
- running the taxi assignment algorithm,
- assigning taxis to MUs,
- updating the current dispatching situation (e.g. monitoring the movement of taxis and the actual pick-up and transport times).

The program takes the form of an applet and a stand-alone application and it was written in Java 2 SDK 1.3.1 on a SuSe Linux 7.1 platform. It has a user interface that shows the simulation in action and some results of the simulation. In addition, it allows the user to select the simulation scenario using either current taxi service, which we call "BasicTaxi" or the newly proposed approach, which we call "SmartTaxi" and adjust some parameters discussed in this chapter.

5.2.1 Map

In the user interface of the simulator, the area in which all of the transport action takes place is a Euclidean area represented by a rectangular grid of 300 pixels X 300 pixels, where we projected a road map of Belo Horizonte's central region. The map scale is 1:12000 and it represents a area of 3.36 km X 3.36 km square. Obviously, each pixel corresponds to 0.0112 km.

We represented the road network using a directed graph with each edge depicting a one-way road and each node corresponding to an intersection. Two-ways roads were represented as a pair of edges, one in each direction.

Weights (or cost) associated to the edges in the directed graph represent the distance in km between two adjacent nodes.

The road network is composed of avenues and streets. Therefore, each edge in the directed graph is labeled either with "A" (Avenue) or "S" (Street). In addition, all of the edges also have an associated speed limit. In this case, all of the avenues have the same speed limit, as well as the streets. Both avenue and street speed limits are parameters that can be configured by the user interface. The speed default values are 80 km/h and 60 km/h for avenues and streets respectively.

The calculation of the shortest paths for the road network is initiated by selecting the 'run' button, if the user changes the speed default values. During this calculation the adjacency matrix values are updated. This happens because the distance values are weighted by speed limit.

Figure 5.1 shows the directed graph, which represents the road network of Belo Horizonte's central region.

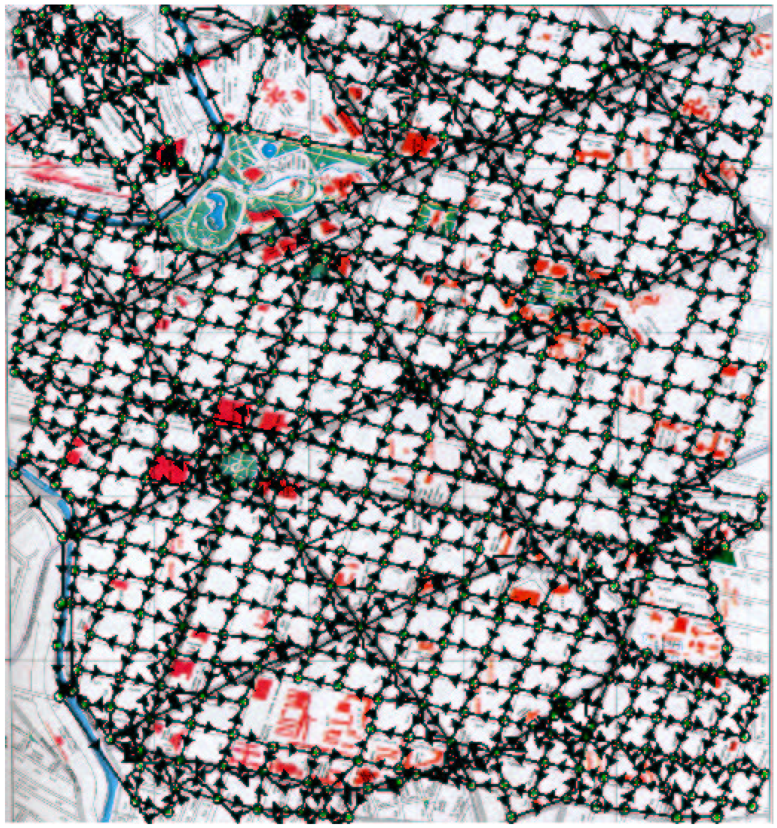


Figure 5.1: The road network using directed graph

Given the directed graph, the system retrieves the avenue or street angle for the taxi movement at the current time, every time it moves from one intersection to another. This angle, corresponds to the "θ" angle associated on each edge reported by Model 5.1.

$$\theta = \arctan \frac{node_2.location.y - node_1.location.y}{node_2.location.x - node_1.location.x} \tag{5.1}$$

Therefore, each edge has its own θ angle, which we define to be either the avenue or street angle.

5.2.2 Simulator Object Model

The taxi service model implemented consists of some objects. The relationship between these objects is represented in Figure 5.2 using Object Modeling Technique (OMT) diagram [RBP+91].

All of the objects and another data structures needed to the simulation are created in the begining of simulation and kept throughout its execution.

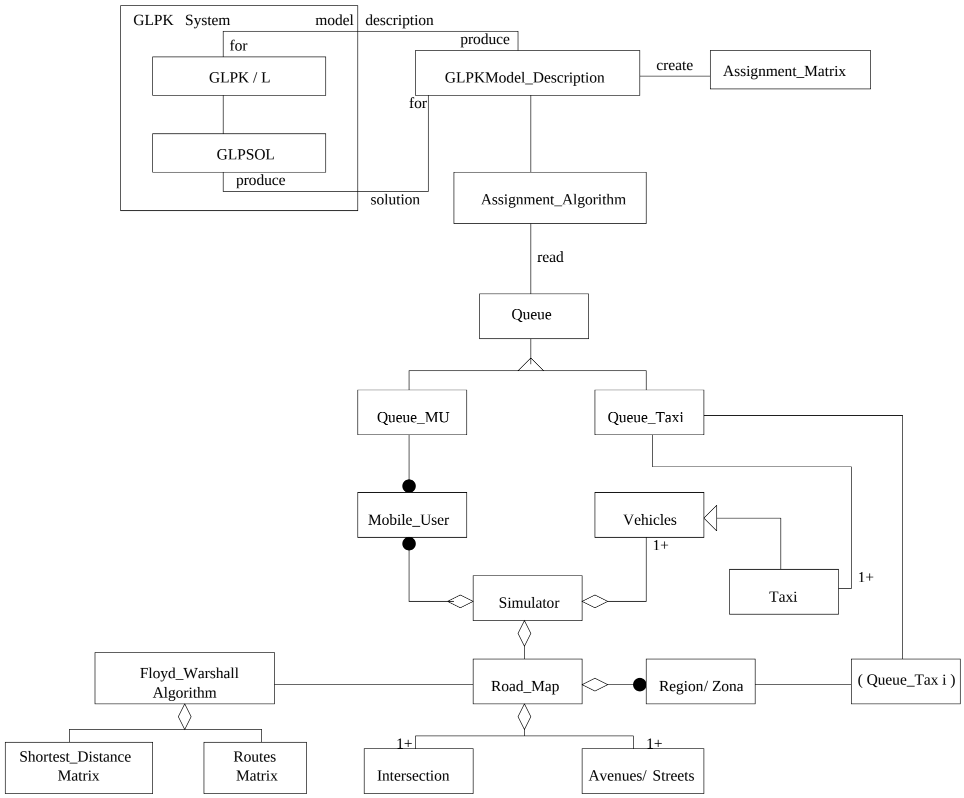


Figure 5.2: Simulator object model

5.2.3 Scenarios

There are two scenarios in the simulator, namely, Basic-Taxi and Smart-Taxi. Each simulation is only allowed one of them.

The Basic-Taxi scenario models the current approach used to manage a taxi fleet. In this case, there is a queue-type mechanism, which associates a taxi queue to each region or zone.

The zones have been modeling as shown in Figure 5.3. The local area was divided into the four regions, depicted by quadrants 1, 2, 3 and 4. Each quadrant has a taxi queue associated. The procedure of a taxi moving from one quadrant to another, while serving a MU, we should call handoff. When a taxi performs the handoff procedure, it can be put in the queue of the destination zone or not. It will be put in the queue of the destination zone if this zone contains the MU's destination point. In this case, after the dropoff event, it will put in the queue. Otherwise, nothing happens and it continues its movement.

Let's, consider the case of a taxi that moves from zone one to zone four and from zone four to zone three, respectively (see Figure 5.4). Suppose that it has been serving the same MU. In the first handoff ($1 \rightarrow 4$), the taxi will not be put in the queue of zone four. But, it will be put in the queue of zone three, where it should perform the dropoff.

The queue-type mechanism includes the following:

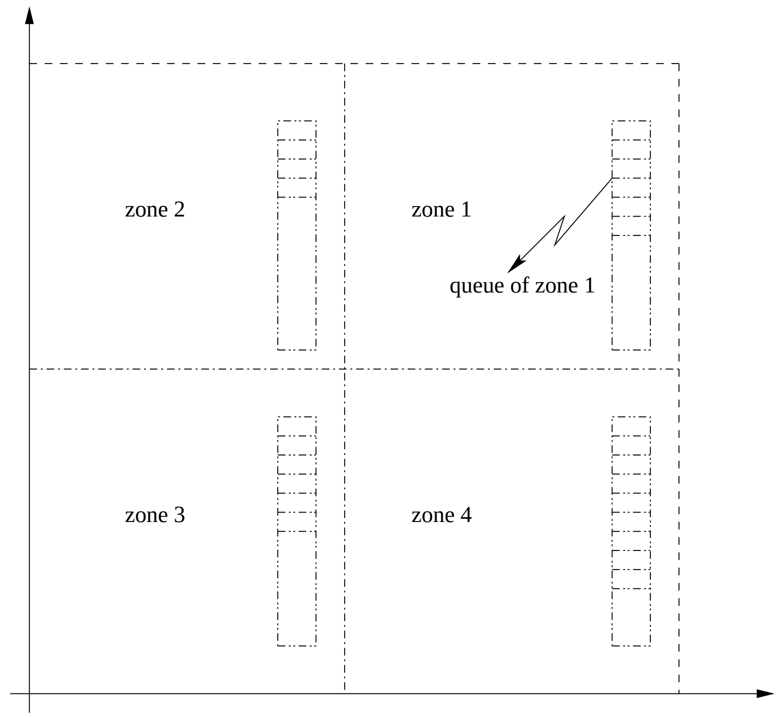


Figure 5.3: Zone modeling process

- The calls are received by the gateway[1], which determines which taxi is to be assigned the call.

- Call information is transmitted to the first taxi in the queue of zone associated to the call's address.

- The driver is allowed to accept or reject the request. The probability of him accept or reject a service request is specified by the 'Service Refusing Probability' - a parameter configurable by the user in the simulator.

- If the driver rejects it, the driver is kept in the queue for the next available request, and the request is offered to the next available driver.

- If the taxi driver can fill the request, then he is assigned to the user.

- If not, the gateway verifies the next taxi driver in the queue until reaches a taxi that can fill the request.

- If no taxi is found, the gateway will begin searching nearby regions for available taxis.

Currently, the algorithm used to scan other regions or zones is subject to control by the dispatch manager and varies from region to region.

[1]In the simulator the receptionist is represented by a gateway

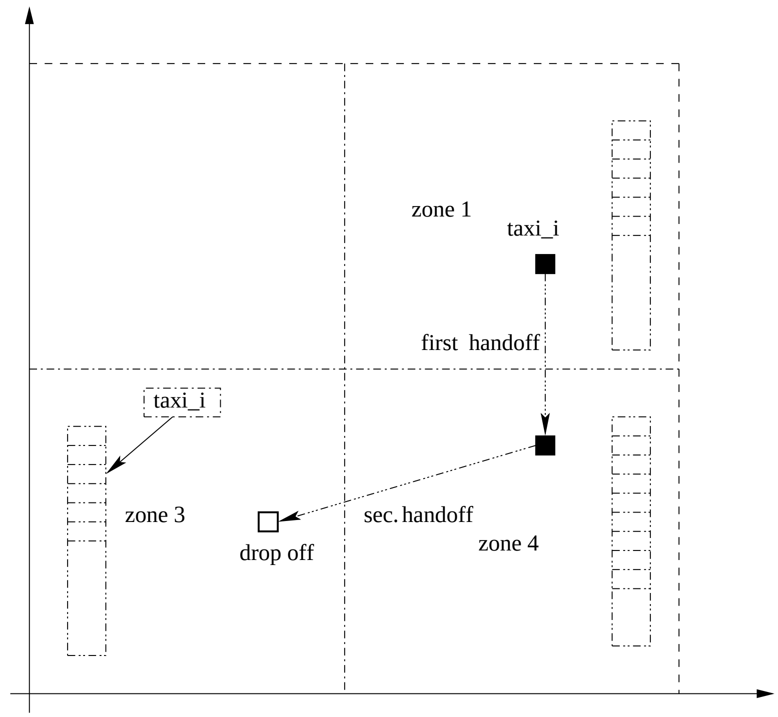


Figure 5.4: Example of handoff process

On the other hand, the Smart-Taxi scenario models the new approach proposal, where the assignment process of a taxi to one MU is made by an assignment algorithm. In this case, the location data of both MU and taxi is very important. The algorithm assigns the nearest taxi to the MU minimizing the distance and time. The new approach is delineated in more details in the Section 4.3.

Both scenarios are fundamental in the simulator, once they make us possible through a lot of simulations, to compare the obtained results among them.

5.2.4 Time Management

The basic time advance mechanism used in the simulator is the constant increment. In other words, the simulator works with discrete time steps instead of real-time calculations. At each time step all objects are notified and move exactly one space per time step.

The simulator first sets up a time horizon that spans the entire day. For example, an operation that runs from 6:00 AM to 12:00 AM (midday), has a time horizon of 21600 seconds (or 360 minutes or 6 hours). Within this horizon, discrete time events are produced by the simulator to account for the occurrence of new service requests.

Specifically, the generation of a new taxi service request is dependent on this discrete clock. The MUs arrival process are considered to follow the Poisson model. The time interval between arrivals are considered to have a Exponential distribution. The new request proceeds as follows: given the current time t, a Exponential distribution with mean μ[2] is applied to determine the time of occurrence of the next request.

The day is divided into five periods: early morning, late morning, lunch time, early afternoon and late afternoon. A set of probabilities (which means the service request probability happens in a given unit of time) and time intervals in hours, is used for these periods. Both are configured in a file named *timetable.txt*[3], as shown in Figure 5.5, before running a simulation.

periods	time interval		Probability (%.)
EARLY−MORNING	00:00	06:00	10
LATE−MORNING	06:00	12:00	38
LUNCH−TIME	12:00	14:00	05
EARLY−AFTERNOON	14:00	17:00	40
LATE−AFTERNOON	17:00	24:00	07

Figure 5.5: Configuration file of day's periods

Therefore, the standard time is from 00:00 AM to 24:00 PM (midnight), which can vary with the user's need.

5.2.5 Mobile Users

In the simulator the MUs appear at random locations with random destinations. The MUs appear one at a time following a Poisson process [Knu81] with a mean that is specified by periods of day (see Figure 5.5).

Each MU can be one of the modes shown in the Table 5.1. It is important to note that the modes transition order is from NEW-USER to DROPOFF. That is, one MU must pass by each one of previous modes before arriving at final mode (DROPOFF).

The simulator generates the characteristics of each new MU, namely, its time of occurrence as well as the pick up and destination locations. Generated this characteristics the simulator then starts to produce the life history of MU. This history must contain the modes transition in the order shown at Figure 5.6.

[2] Rate of service requests per time unit

[3] The standard values in this file were obtained by averaging of service request time values of five selected taxi companies scattered geographically throughout Belo Horizonte. The purpose of interviewing them was to learn about the current dispatching system and to identify the average of request in a normal day.

MU Mode	Description
NEW-USER	A new mobile user was created.
REQUIRING-SERVICE	The mobile user requests the taxi service. In this case, he remains in REQUIRING-SERVICE mode until the time window is reached and a taxi can be assigned to serve this request. This happens in the 'Smart taxi' scenario. On the other hand, in the 'Basic Taxi' scenario serving the MU does not depend on a time window, but on a free taxi standing in the queue.
WAITING	A taxi was assigned to the MU and he is waiting to be transported to his destination.
IN-TRANSIT	The MU is being transported to his destination.
DROPOFF	The MU was dropped-off at his destination. End of service.

Table 5.1: Mobile user modes

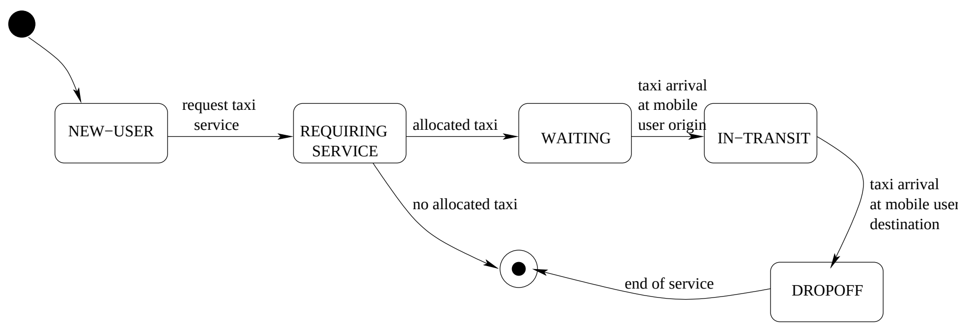


Figure 5.6: Mobile users states transition diagram

5.2.6 Taxis

In the begining of the simulation the taxis appear at random locations. Each taxi is only allowed to carry one MU at a time. In practice this MU may be one person or a group of people, but we treat it as a single object for simplicity and to avoid multiple destinations. In the assignment algorithm the capacity of the taxi is represented as a constraint.

Each taxi has its own speed in km/h[4], which depends on the maximum speed allowed on the road map. The driving method for deciding how to get from one point to another does not follow any standard movement. The directions are defined by the movement angle, which is updated in each intersection.

The taxis can use one of the two procedures described below to communicate with the gateway (or service center) when the event is the occurrence of a new request. These procedures were implemented as evidence of taxi moviments using the different scenarios.

Simple-Queue this procedure is used in the 'Basic Taxi' scenario (that represents the current approach). If the taxi is in the queue, it is free, then a positive acknowledgement is returned. Otherwise, a negative acknowledgement is returned. In this case, the taxi can be occupied picking up or transporting a MU.

[4]In the simulator code the taxis's speed is converted as follows: $km/h \rightarrow m/s \rightarrow pixel/s$. Therefore, 1 km/h is the same as 0.0247 pixels/s

Taxi Mode	Description
WAITING	WAITING mode means that the taxi is free, that is, it is waiting for MU service request.
PICKUP	PICKUP mode means that the taxi was in the WAITING mode and it has been assigned to an MU service request. The taxi was dispatched and routed so that, MU is picked up at his origin before being transported to his destination.
DROPOFF	When the taxi is in this mode, it is busy, that is, it is transporting the MU to his destination. The taxi gets out this mode when the MU is dropped off at his destination.

Table 5.2: Taxi modes

Smart-Queue this procedure is applied in the 'Smart Taxi' scenario. In the Smart-Queue procedure, the taxi always returns a positive acknowledgement to the gateway. This happens, because if the taxi is currently free, it is automatically included in the process to solve the assignment problem. On the other hand, if the taxi is busy, then the pick-up and drop-off times are calculated based on the taxi's current trip. In this case, the taxi is only included in the assignment process if the time calculated was less than the lower bound of the next time window. That is, if the taxi will finish its current trip before the beginning of processing the assignment matrix.

During the simulation process a taxi can assume one of the modes or states described in the Table 5.2. When a taxi is either in the PICKUP or DROPOFF mode, it is busy. All of the possible transitions are shown in the Figure 5.7.

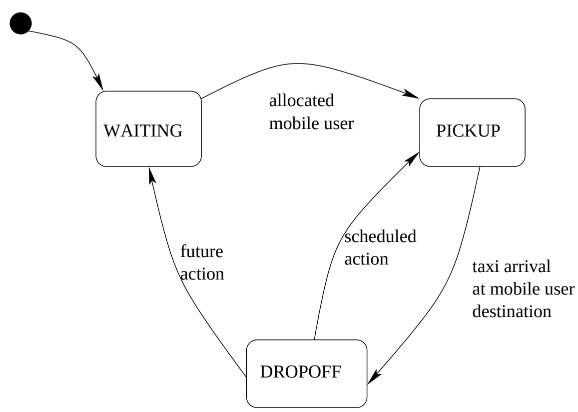


Figure 5.7: Taxi states transition diagram

In the beginning of the simulation all of the taxis are in the WAITING mode, they start an action when they are assigned to transport a MU. As a result, the taxis should make a transition to the following modes: PICKUP and DROPOFF respectively. When the DROPOFF mode is reached, the taxi verifies if the queue attached to it has a scheduled request, then the transition to PICKUP mode occurs. Otherwise, a transition to WAITING mode occurs, where the taxi remains waiting for a future request.

In addition, a route is associated to each taxi that contains the shortest path to pick up the MU and to drop him off at his destination. This route is defined when the taxi is assigned to serve a service request.

5.2.7 User Interface

The graphical user interface was developed to enable the simulation to display some results and to allow the user to configure some parameters during simulation.

This interface consists essentially of three sections (see Figure 5.8). The left portion of the applet allows the user to set some simulation parameters and to start, stop, and resume a simulation. The central portion shows the space on the road map where both MUs and taxis move, that is, it shows the taxi service in action. The last and right portion displays information about the current simulation. In addition, this portion also shows specific information about any taxi or MU that is selected with a mouse click.

Figure 5.9, shows the user interface at the moment that a simulation is running. The scenario under simulation is the "Smart Taxi". In this case, we can note that there is a selected taxi, which has its data (taxi number, mode, current location) displayed and also its pickup and dropoff route drawn. Moreover, a histogram depicts the average pickup time for the simulation scenario.

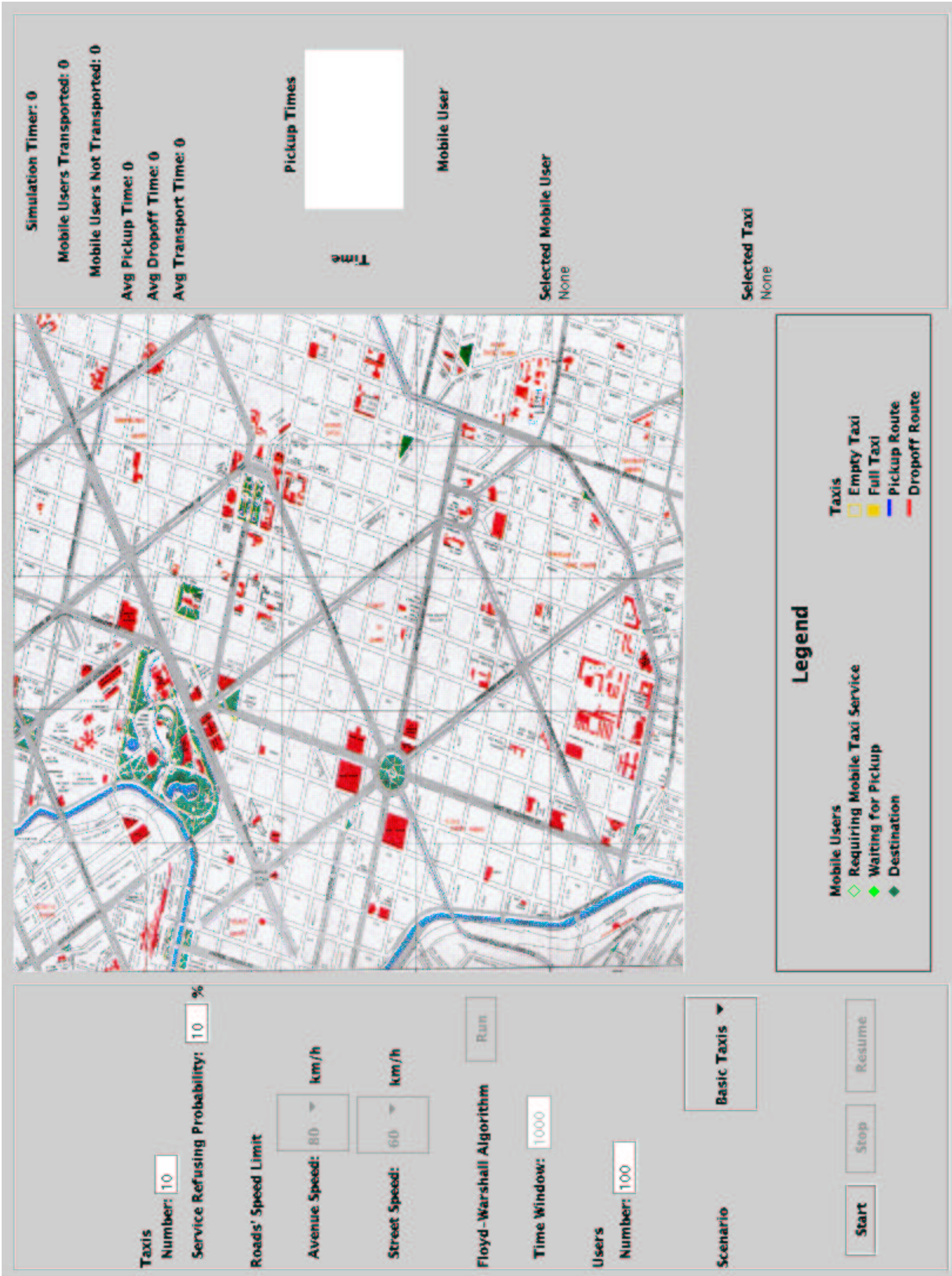


Figure 5.8: LBTS simulator user interface

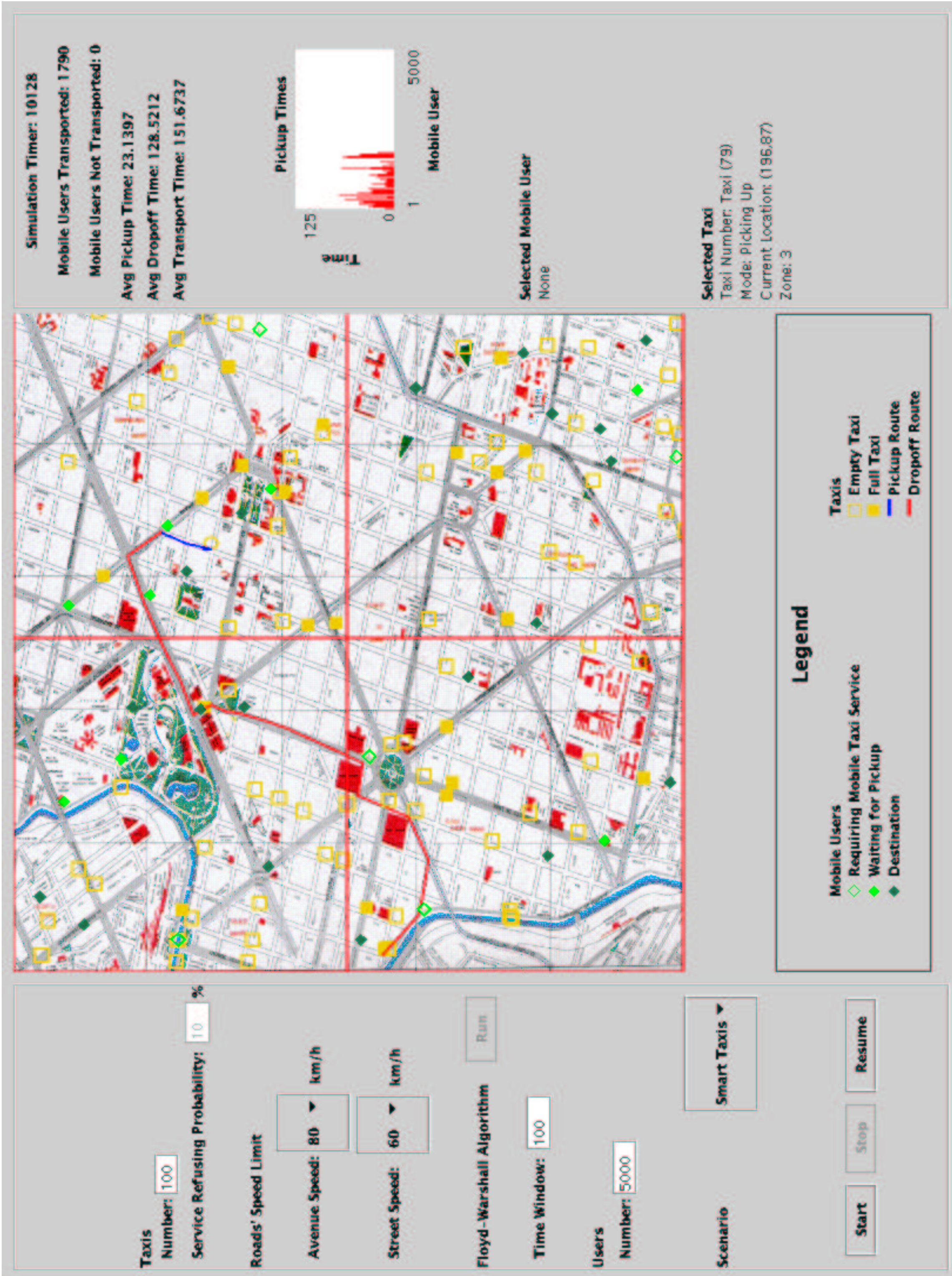


Figure 5.9: LBTS simulator user interface in action over "Smart taxi" scenario

Chapter 6

Experimental Results

6.1 Introduction

This chapter describes the results obtained and the system conformance degree to the original intent and design. The testing setup is also described along with the results obtained for a lot of experiments. In addition, the aim of this section is also to compare two approaches for handling new requests in mobile environments, when the arrival rate of these requests is of increasing intensity and other parameters are changed. Some remarks are made based on the results, along with suggestions for improvement.

6.2 Experimental Setup

The following results were obtained using three workstations. Each station is a Sun SPARC with 1Gbytes of main memory and 30Gbytes hard disk. Two of them run Solaris 8 and the third runs Linux SuSe 7.1.

The day is divided into five time periods: early morning, late morning, lunch time, early afternoon and late afternoon. The lunch time period is smaller than the others (see Table 6.1). The reason that we divide a day into periods is that we want to verify the avalability of taxis fleet for different probability of new request occurrence during a normal day. Table 6.1 shows the service request occurrence probability per periods. These probability values will be used throughout this chapter.

Next we give definitions that can help us describe our analysis clearly and concisely (see Figure 6.1).

Pickup: a link that directs from the taxi's current location to the MU's origin.

Dropoff: a link that directs from the MU's origin to the MU's destination.

Periods	Interval Time	Probability (%)
early-morning	00:00 - 06:00	10
late-morning	06:00 - 12:00	38
lunch-time	12:00 - 14:00	05
early-afternoon	14:00 - 17:00	40
late-afternoon	17:00 - 24:00	07

Table 6.1: Service request probability definition per periods

Transport: a transport must satisfy the following expression: $Transport = Pickup +$ Dropoff. This implies that a transport always starts with the request of the MU.

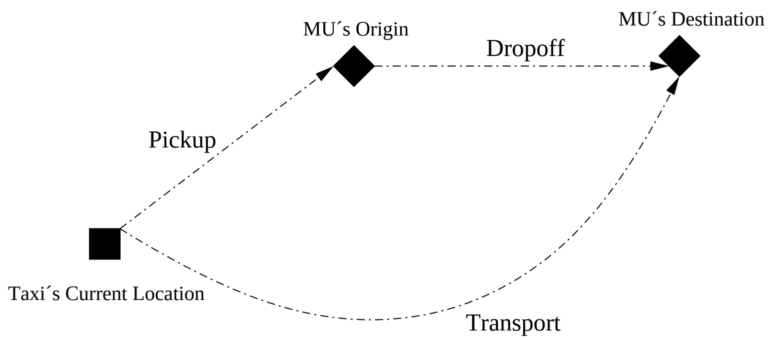


Figure 6.1: Movements definition

Each simulation have been run 24 hours, changing some parameters. All the results were obtained by average of the values (time and distance of pickup, dropoff and transport) over each set of 25 simulation runs. The average of each simulation is 2 hours and 30 minutes (in real-time).

The number of requests varies over time. To provide a meaningful comparison with the real world, we define periods of peak, using again the five periods of Table 6.1. This peaks are reached in both late morning and early afternoon periods as show in Figure 6.2.

In the periods of peak, when the new request arrival rate increases, the assignment algorithm used in the "Smart Taxi" approach does not require a great time to produce a solution. That is, the cost of running assignment algorithm does not increase significantly since the shortest distances and the routes have already been calculated before beginning the simulation. Figure 6.3 illustrates this point produced under "Smart taxi" approach. Within each set of simulation runs we considered 100 taxis, 1000 MUs and 63 request per hours per period, on average. Note that the average pickup time is not as great. Clearly, the assignment algorithm is effective, since the pickup time has not great oscillation. This approach should perform better than the current approach, since the algorithm optimizes the current service. This improvement can be explained based on the fact that, each time a new request comes in, the new approach assigns the nearest taxi to MU considering free and busy taxis.

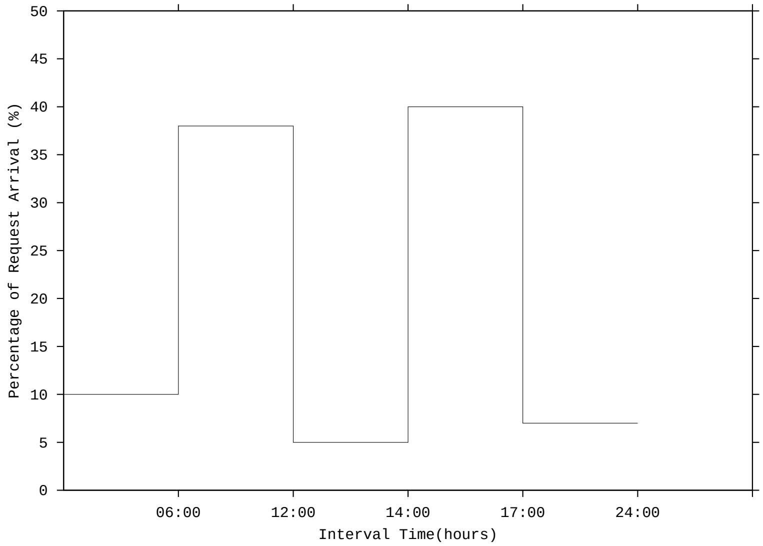


Figure 6.2: Percentage of requests per periods

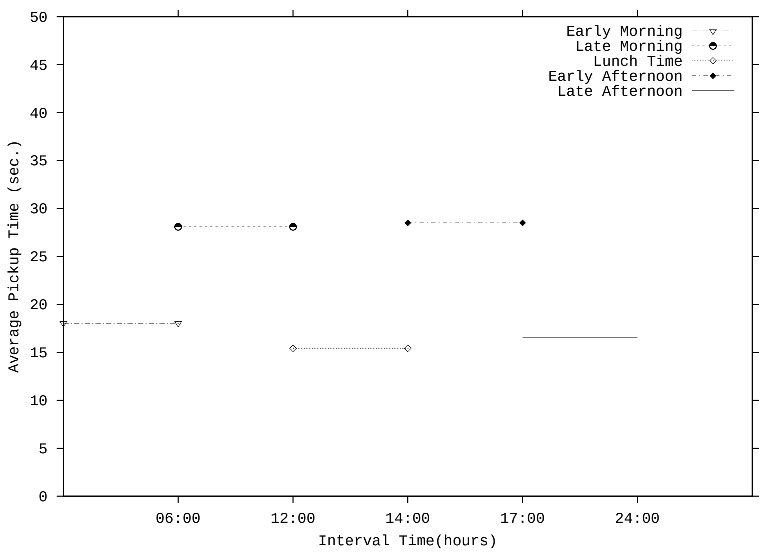


Figure 6.3: Average pickup time per periods in the “Smart Taxi” approach.

In particular, the “peaks” observed in Figure 6.3, which happen when there is a fairly large number of service requests, shows that the assignment algorithm can take advantage of optimization opportunities. Moreover, we can see that number of request from 00:00AM to 06:00AM and 12:00AM to 14:00AM and 17:00PM to 24:00PM is low. One aim of studying day’s periods is to predict the different probabilities of request during a normal day.

It is clear that a comparison between “Basic Taxi” and “Smart Taxi” approaches needs to be valided by approximated location of taxis and MUs and the number of requests in both approaches. A serie of simulation runs have been made for both approaches. Figure 6.4 shows a simulation using 100 taxis with 60/80 Km/h speed in the avenue and street, respectively. The current and new approaches histogram are similar, then we only show current approach in Figure 6.4.

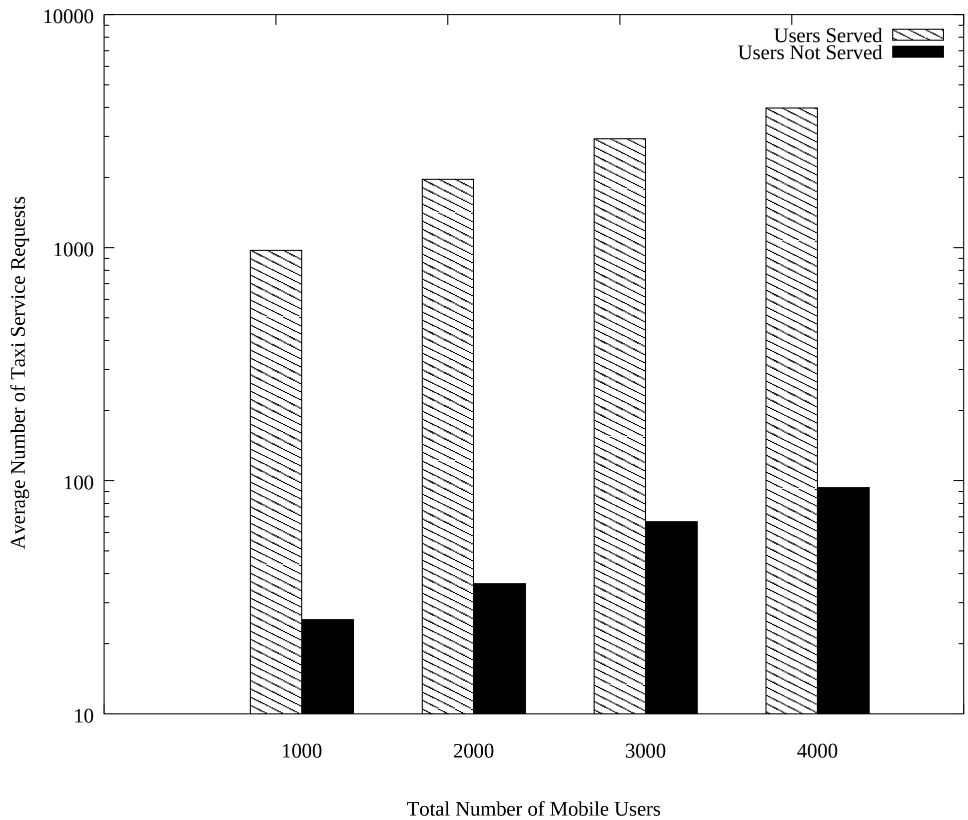


Figure 6.4: Average number of taxi service requests

The histogram shows the average number of service requests per different number of MUs. This datum allows us to compare the two approaches. In addition, the location points of both taxis and MUs repeat when the number of simulation runs increases.

We also observe that the average number of user not served is basically the same in both approaches. However, the percentage of users not served in the new approach, is approximately 2.13% while the current approach is 2.16%. This small difference, can be explained by the fact that the new approach always assigns the nearest taxi to MUs. In this case, the taxis end a service more early and it can serve another request. Moreover, the new approach considers the assignment of a busy taxi, whether its destination point is near of a new request location.

The results in Figure 6.4 show that when the number of MUs increases the average number of users not served also increases. The obvious explanation, is that as the number of request increases, the taxi company either has not more taxi available or in the case of the current approach, the service refusing probability is high. Thus, we can conclude that the demand is greater than the supply capacity.

We can examine now the case where increasing the number of MUs, the average total distance traveled also increases as shown in Figure 6.5. We note that when there are more MUs in the system, more taxi service requests are made. The aim of the taxi fleet is to maximize your gain. For this, the taxi fleet should serves more service requests. Then, since more user are served, the average total distance traveled increases. We know from previous discussion that the new approach minimizes the total distance traveled. On the other hand, the current approach is based on queue-type strategy. Therefore, Figure 6.5 clearly depicts that the distance traveled on the new approach is smaller than on the current approach. The error bar shows one standard deviation in each direction.

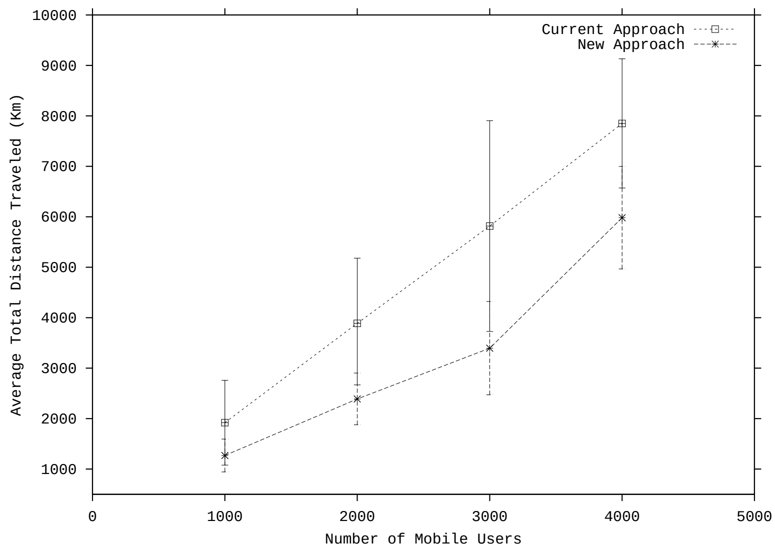


Figure 6.5: Average total distance traveled for both approaches.

We are interested in various aspects in the result analysis. Among them, the total distance traveled and pickup time, which reveal the advantages of using either the current approach or the new approach. Figure 6.6 shows the average total distance traveled per different values of speeds and different number of taxis over the "Basic Taxi" approach (current approach), while keeping the same number of user (2000). We can observe that the average total distance traveled is linearly increasing with speed (especifically the curves that indicate 50, 100, and 200 taxis). In the current approach, the taxis are located in queues, which belongs to a specific zone. In this case, when the number of taxis increases, the number of users not served decreases and the distance traveled value tends to increase.

More surprisingly, we observe also that as the number of taxis is 100, the correspondent curve does not follow the same behaviour of others curves. This can be delineated, by the fact that the number of users not served is low, indicating that more users are served when we compare with the curve correspondent to 150 taxis, in the point where the speed is 40/60 Km/h.

We examine now the case of the new approach. Figure 6.7, also considers the average

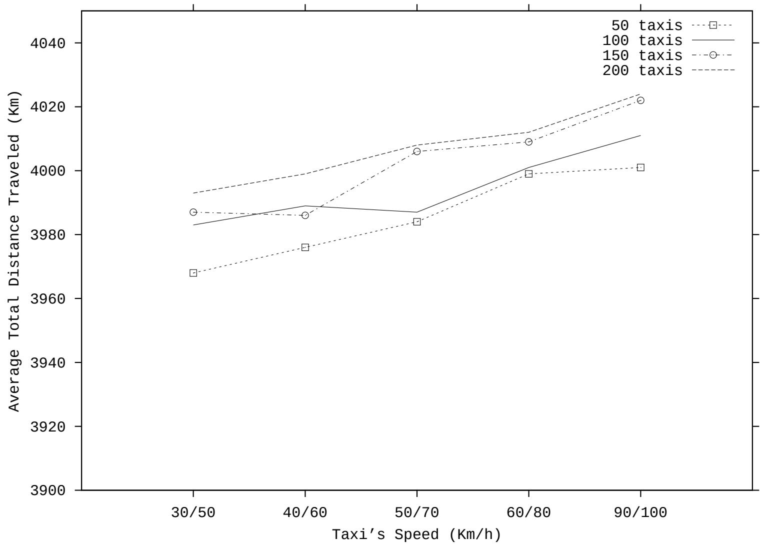


Figure 6.6: Average total distance traveled for current approach.

total distance traveled per different values of speeds and different number of taxis with time window equal to 50 seconds. We see that, the average distance traveled decreases, when the number of taxis increases. That means, though the number of taxis is great, they account for a large users percentage that can be served and at the same time, the probability of the taxi to be near of users location is high. In this case, the pickup distance is very low. Moreover, our approach is always looking for assigning a nearest taxi to MU and the distance traveled is calculated by the sum of pickup distance and dropoff distance. The average total dropoff distance is almost equal throughout simulation runs, around 2393 Km. Thus the average total distance traveled tends to decrease.

We can also note that the average total distance traveled tends to have a linear behaviour, similar to the current approach, when the taxi's speed increases. It is clear that a taxi with high speed ends its job more quickly and then it might serve more user.

Figure 6.7 also shows an unexpected result, which is the change of behaviour of the curves that refer to 100, 150 and 200 taxis. In these case, there is a difference of 10% approximately between the results of distance traveled, when the speed is greater or equal to 50/70 Km/h. This difference tends to 0 in the case of very high speed values or very high number of taxis. Thus the average total distance traveled tends to be similar. This can be also explained by the fact that the MUs appears very near of taxis location, decreasing too much the average pickup distance.

Figure 6.8 shows the average pickup distance per different number of MUs for both current and new approaches. We can see that the pickup distance is linearly increasing with the number of MUs. The new approach, which has the lower average pickup distance values is therefore better when compared with the higher average pickup distance values of current approach. Moreover, we can conclude that the pickup distance is proportional to the number of request of that day.

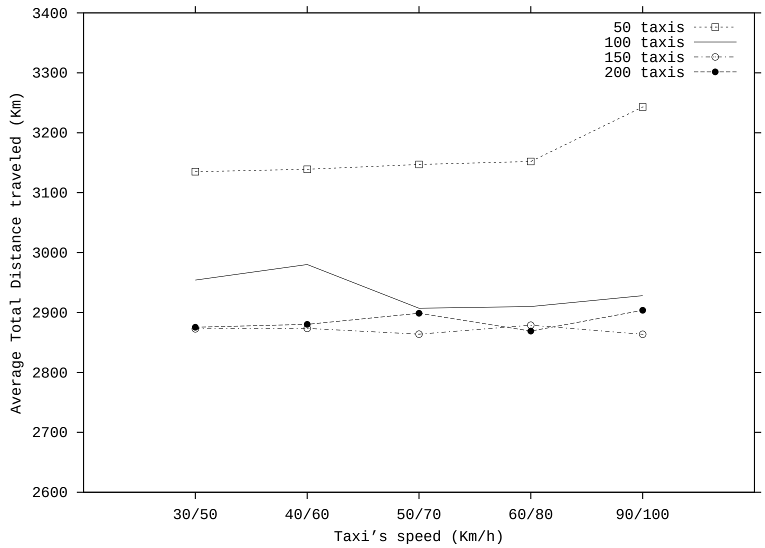


Figure 6.7: Average total distance traveled for new approach

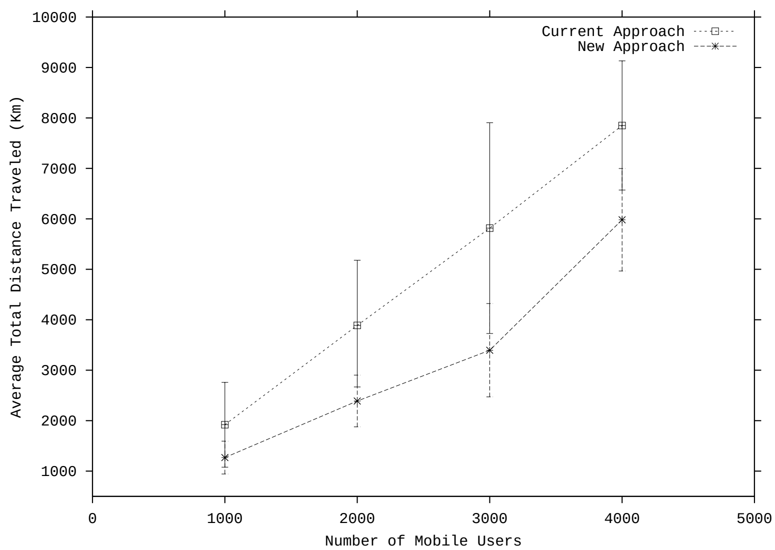


Figure 6.8: Average total pickup distance for both approaches.

Figure 6.9 shows the average travel time versus the number of MUs in the system for both current and new approach. We see that each approach has similar average travel time for different number of MUs. This can be explained by the fact that we have used a limited area and we can conclude that after a lot of simulation runs, the average travel time tends to be similar, due to the limited environment. In addition, we can also see that the average travel time of current approach is large than the average travel time of the new approach.

Figure 6.10 shows the average pickup time per different number of MUs for both current

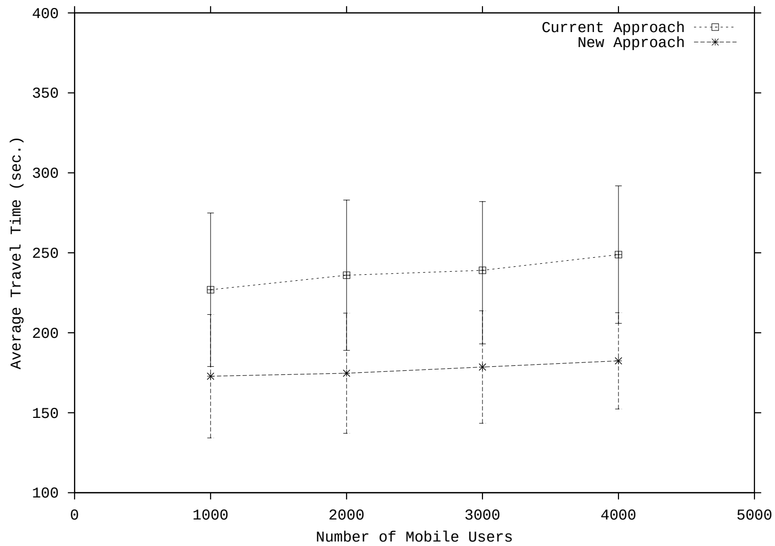


Figure 6.9: Average travel time for both approaches.

and new approaches. We see that the pickup time in the current approach has higher values for different number of MUs.

In all simulations of the current approach, the service refusing probability parameter had the same value (10%). Table 6.2 shows the service refusing percentage per different number of MUs. We observe that service refusing increases when the number of MUs increases. This implies that the system can take away more time to serve the users, so we can also say that long serve time obviousily generate high pickup time.

MU	Service Refusing Percentage (%)
1000	3.7
2000	5.8
3000	6.0
4000	9.6

Table 6.2: Percentage of refused requests, but that could have been served by another taxi

An important point to be discussed is the refused call at current approach penalizes this approach in aid of the new proposal approach. In the new approach we consider that a taxi driver never refuses a service request. A primary reason is all system uses wireless communication technology. What should enable the system manager knows the fleet status which can be tracked by a graphical interface. Therefore, the system manager knows whether a particular taxi represents the state it claims to be at. Besides of this, as the system goals are fleet management high demand and quality of service, the vehicles are took to accept a service request independently, for example, of the price or fare.

In the previous case, the current approach is penalized when the service refusing probability parameter is set. This can be explained by the fact that at the current approach

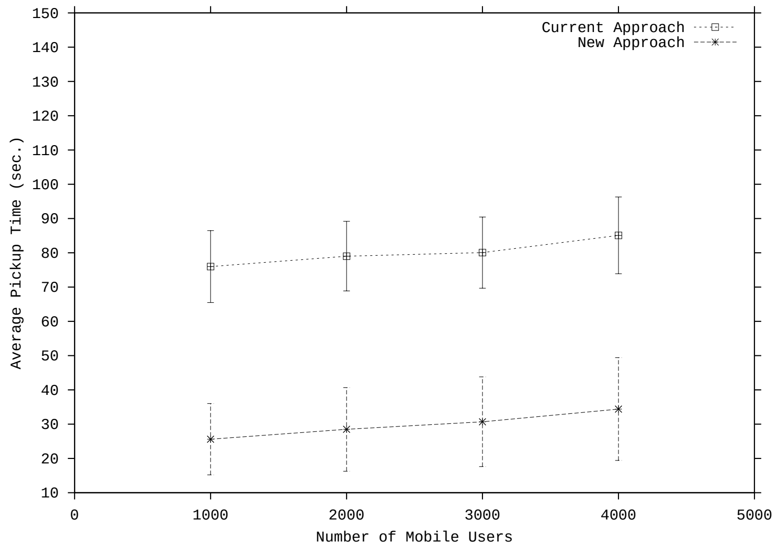


Figure 6.10: Average pickup time for both approaches.

there is no appropriate fleet management. It is being very common at current approach a taxi to refuse a service request because of the low price. In addition, as the telephonist depends on location information provided by the taxi driver, this information could not be true. As a matter of fact, the fleet management can be more complicated.

Figure 6.11 and Figure 6.12 show the average number of users not served per different taxi's speed and different number of taxis for current and new approaches, respectively. Clearly, we can observe that for very high speeds the number of users not served decreases, for both approaches. The same happens when the number of taxis is increased. However, Figure 6.11 shows an unexpected result, which is the spike in the average number of users not served that occurs when the speed is 40/60 Km/h. It is expected that the number of taxis that refusing a service could have been high. The solution is to decrease the service refusing probability.

Figure 6.13 shows the average pickup time per different time windows. Clearly, we see that the average pickup time linearly increasing with time window. For a time window $\approx$ 2000 seconds, the average pickup time of the new approach is similar to the average pickup time of the current approach. We can also observe that for time window $\leq$ 500 the average pickup time values are almost equal.

Figure 6.14, Figure 6.15, Figure 6.16 and Figure 6.17, shows the histograms of average pickup time of some simulation runs, with time windows 100, 500, 1000, and 2000, respectively. We can see that the pickup time on the "Smart taxi" approach only reaches the same average time on the "Basic taxi" approach when the time window is equal 2000 seconds (see Figure 6.17).

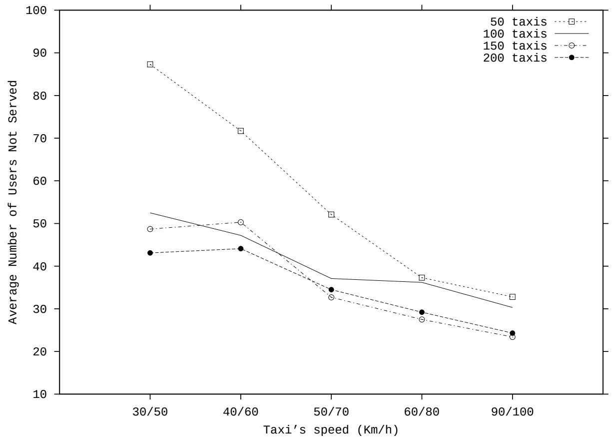


Figure 6.11: Average number of users not served in the current approach

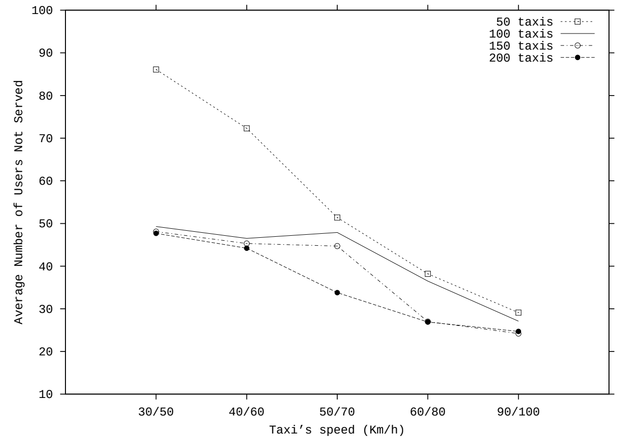


Figure 6.12: Average number of users not served in the new approach

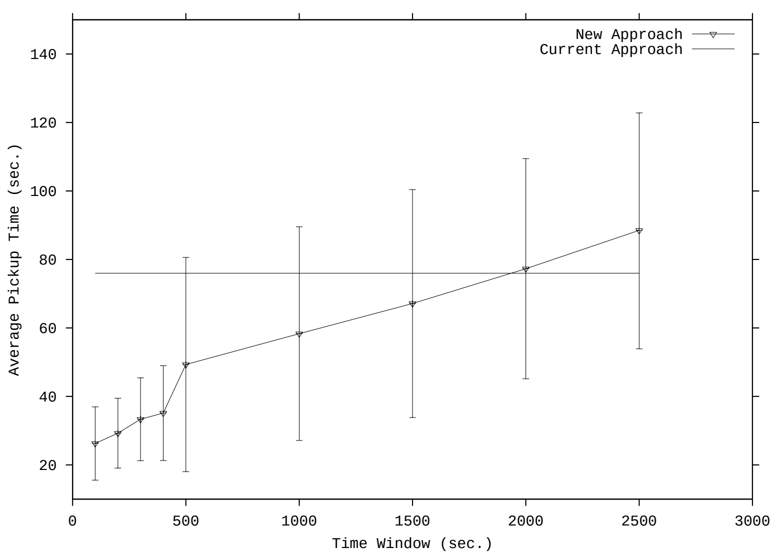


Figure 6.13: Average pickup time per different time windows.

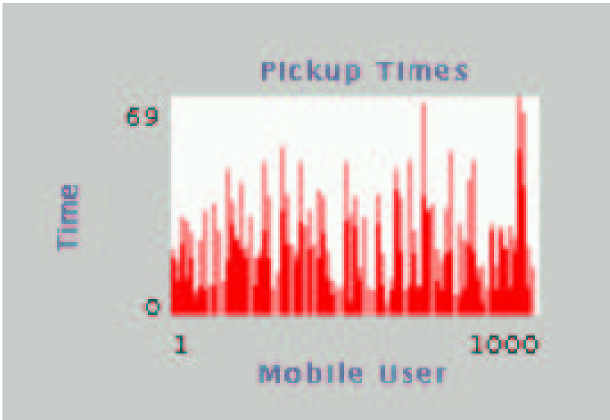


Figure 6.14: Time window = 100

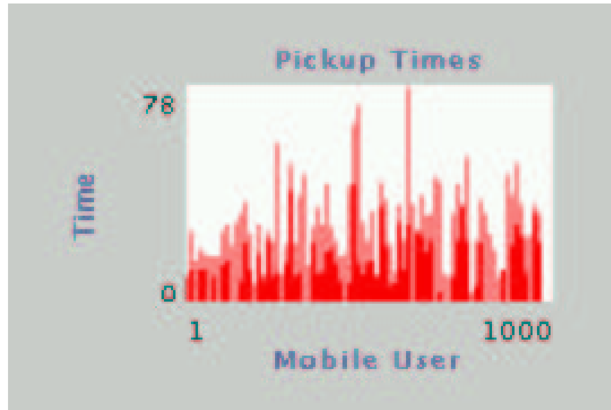


Figure 6.15: Time window = 500

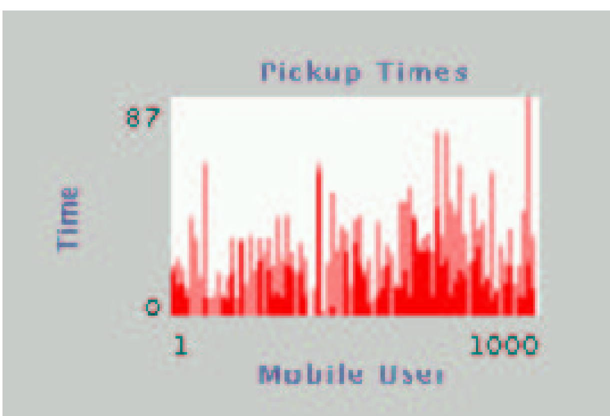


Figure 6.16: Time window = 1000

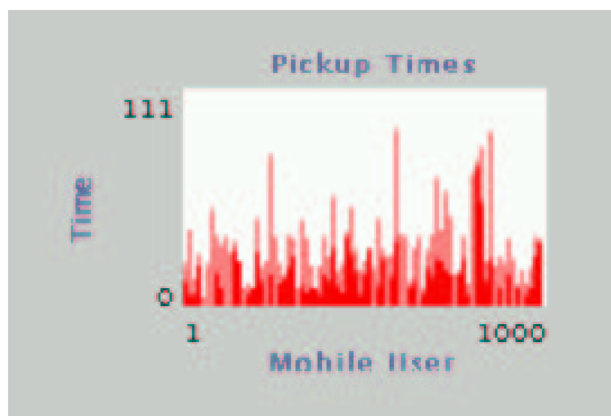


Figure 6.17: Time window = 2000

Chapter 7

Conclusions and Future Work

7.1 Introduction

This chapter presents the conclusions drawn from our research and possible future extensions to our work. Section 7.2 summarizes the dissertation and identifies our contributions. Section 7.3 discusses directions for further research on the interaction among location-based services, wireless communication and network optimization.

7.2 Summary

This research investigates the integration of taxi service with mobile environments. Current taxi service being provided to users is a relatively simple process. But, with wireless communication progress, it is possible to provide new applications, high quality services and better resource management.

This work led to the development of an initial framework for providing location-based taxi service to mobile users. The framework is realized as a prototype system, which involves a Java-based application.

More specifically, this work models and develops a distributed environment where mobile users appear at different locations randomly over time, and these MUs have access to taxi service dynamically as they need to be transported to different destinations. Moreover, there is a fleet of taxis that performs the transport of these MUs. Given this situation, the problem at hand is to decide which taxi is assigned to transport each MU, and doing so in a way that the distance traveled and time of both MU and taxi are minimized.

Two approaches have been modeled, "Basic taxi" and "Smart taxi". The first approach has simulated the current approach, which uses queue-type strategy. The second and last approach simulated a new proposal approach, which is based in wireless communication and network optimization.

The implemented prototype has allowed to simulate both approaches through some parameter setup. And also simulation data files are generated for each execution. In addition, the prototype implementation works well under reasonable loads, that is, we have been run simulations with great number of MUs and taxis (scalability).

Our numerical results show that the current approach is not the more efficient strategy for taxi companies. Because of the taxi company does not have the control of the taxi locations and the assignment process seldom allocates the nearest taxi to MU. The previous fact results in a great pickup distance and also high pickup time.

On the other hand, the results have also shown that the new approach produces better results than current approach. Clearly, the new approach is more effective, since the distance and time values were lower. Moreover, when the request arrival rate increases, we observe that this tendency keeps on.

At this point, the benefits associated with the new approach are the greater taxi fleet management, smaller distance traveled and pickup time. The first benefit is associated with the use of location tecnologies (wireless communication) and the second and third are associated with the network optimization techniques.

For last, the framework showed how computing, wireless communications and location information combined can provided a typical location-based taxi service.

7.3 Future work

7.3.0.1 Demand Forecasting

The work performed to date on local area vehicle routing and dispatching problems leaves a number of unanswered questions and open research avenues. Among them, the issue of demand forecasting has not been addressed yet, in contrast with dispatching systems found in emergency services and truckload trucking. In these applications, each vehicle is dedicated to a single user and the issue of relocating vehicles to anticipate future demands, once the service is completed at the current user location, is of paramount importance.

7.3.0.2 Additional Stochastic Elements

In this work, we did not consider additional stochastic elements, like vehicle breakdown or congestion due an accident. In future work, in order of minimizing road network traffic, we are interested in identify congested roads and transmit reports to wireless receivers in taxis. One alternative would be stored historic information about congestion for use when planning routes.

7.3.0.3 Transportation of Multiple MU

In this work, each taxi can only transport one MU for only one destination. In future work, we are interested in making possible that each taxi could transport users group for different destinations.

We also want to investigate the possibility of update the route in each destination point. In the case of the delay to compute the best route that pass by every destination exceed a specific limit value.

7.3.0.4 Direct Communication Between MU and Taxi

One of the important features of this work is that, users and their mobile devices should be able to request taxi service provided by a taxi fleet company.

Therefore, in this work, we did not consider those users that communicate directly with a taxi. Although the users are using the taxi service, they are not accessing the company system, which uses wireless communication technologies.

This fact takes news issues. For instance, we can predict that the modeling of this fact in the current approach is trivial, but for the new approach has some inconveniences, such as minimizing the user satisfaction.

In this case, the assignment algorithm might include in the solution a free taxi, which may be occupied during the algorithm running. Thus, when the algorithm ends, the best solution may be the worst solution for the MU that was assigned to that particular taxi. In this way, the computing quality may be lost because the solution computed does not apply anymore to the current situation.

In future work, we intend to investigate heuristics that raise the issue of finding good trade-offs between computation time and solution quality. In addition, theoretical work may also be spent on the worst-case analysis of algorithm to evaluate how much is lost by not having the full information in advance. Moreover, we know that is not possible to guarantee the total user satisfaction, but algorithms can be developed to run in background between events, with new starting solutions that truly reflect the current state of the system for further reoptimization.

7.3.0.5 Performance Evaluation with Wireless Data

Other future direction of research is to evaluate the behaviour of our system while the gateway send multimidea data for both MU e taxi driver. In light of this fact, it is important to study the data lost probability and generated traffic in the gateway. For example, a taxi company can want to guarantee the security of their taxi drivers, sending for it a photo of the MU that requested a taxi service. Then we can investigate the load in the gateway when the service requests increase too much.

7.3.0.6 New Strategies for Others kinds of Transportation

In this work, we presented a specific location-based service for taxi fleet. In future work, we intend to study new strategies to treat other kinds of vehicles, as bus. First, we want to evaluate the best strategy, which allows us to assign the MU to the nearest bus-stop. In this case, the buses have a fixed route and determined stop points. Moreover, we intend to assign MU to bus-stop take into account the shortest distance between MU and bus-stop, and the schedule bus.

Thus, we should be looking at the possibility of introducing a location-based service, which uses GPS. A location-based bus service, may be used for tracking the location of buses, can be linked up with a bus-stop information system that will tell passengers (mobile users) when the bus will arrive at the next bus-stop. At the mobile devices, the arrival times of the next bus of each service will be displayed on screen so that passengers will know how long they will have to wait. It can also be used to tell how crowded a bus is so that more vehicles can be dispatched if necessary. In addition, if a bus is found to be late because of heavy congestion arising from, say, a special event, managers can take or diverting buses.

Appendix A

GLPK - GNU Linear Programming Kit

GLPK is a set of routines written in the ANSI C programming language and organized in the form of a library. It is intended for solving linear programming (LP), mixed integer linear programming (MIP), and other related problems. The current version of GLPK [Mak01a] package includes the API routine glp-read-lpm. This routine is an interface to the GLPK/L [Mak01b] language processor and may be used in the same way as the API routine glp-read-mps.

In this work, we use GLPK/L modeling language, which is intended for writing mathematical programming models. The name GLPK/L is derived from GNU Linear Programming Kit Language. Model Description written in the GLPK/L language consists of a sequence of statements constructed by the user from the language elements described in this document.

In a process called translation, a program calls the language processor, analyzes the model description statements and translates them into internal data structures, which may be then used either for generating mathematical programming problem data or directly by a program called the solver for obtaining numerical solution of the problem.

GLPK/L assumes the following formulation of mathematical non-linear programming (NLP) problem[1]:

$$Min\ (or\ Max)\ \ Z\ = x_{obj} \tag{A.1}$$

[1]Processing model description written in GLPK/L assumes generating data for mathematical programming problem. Although currently the GLPK package is designed for linear and mixed integer linear programming problems only, the GLPK/L modeling language is suitable to express linear as well as non-linear problems.

subject to constraints

$$
\begin{aligned}
x_1 &= a_{11}x_{m+1}, a_{12}x_{m+2}, \ldots, a_{1n}x_{m+n} \\
x_2 &= a_{21}x_{m+1}, a_{22}x_{m+2}, \ldots, a_{2n}x_{m+n} \\
&\quad \cdots\cdots \\
x_m &= a_{m1}x_{m+1}, a_{m2}x_{m+2}, \ldots, a_{mn}x_{m+n}
\end{aligned}
$$

and bound of variables

$$
\begin{aligned}
&l_1 \leq x_1 \leq u_1 \\
&l_2 \leq x_2 \leq u_2 \\
&\quad \cdots \\
&l_{m+n} \leq x_{m+n} \leq u_{m+n}
\end{aligned}
$$

Where:

$x_1, x_2, \ldots, x_m$ rows or auxiliary variables;

$x_{m+1}, x_{m+2}, \ldots, x_{m+n}$ columns or structural variables;

Z objective function;

$a_{11}, a_{12}, \ldots, a_{mn}$ constraint coefficients;

$l_1, l_2, \ldots, l_{m+n}$ lower bounds of variables;

$u_1, u_2, \ldots, u_{m+n}$ upper bounds of variables;

Bound of variables can be finite as well as infinite. Besides, lower and upper bounds can be equal to each other. Thus, the following types of variables are possible.

Bounds of Variables	*Type of Variable*
$-\infty < x_k < +\infty$	Free (unbounded) variable
$l_k \leq x_k < +\infty$	Variable with lower bound
$-\infty < x_k \leq u_k$	Variable with upper bound
$l_k \leq x_k \leq u_k$	Double-bounded variable
$Ll_k = x_k = u_k$	Fixed variable

In addition to LP and NLP problems GLPK/L allows mixed integer linear or non-linear programming (MIP) problems, in which same variables are required to be integer. GLPK/L assumes that MIP problem has the same formulation as ordinary LP or NLP problem.

A.1 Model Expressions

Principal constituents of the system of equality constraints are functions $f_1, f_2, \ldots, f_m$ of structural variables. In GLPK/L it is assumed that these functions may be expressed in the form of model expressions.

Model expression is an algebraic formula built of numeric constants, structural variables and arithmetic operators. In GLPK/L model expressions are used as values, which can be assigned to same language objects. In this sense, the language objects are close to objects of computer algebra systems. Although model expressions have special internal representation, the user can think them as character strings that express the corresponding formula.

Being algebraic objects, model expression allows all the operations, which are applicable to numbers.

The GLPK package is able to solve linear problems only, all model expressions that define equality constraints should be linear. Model expression, which contain no structural variables, is called constant model expression.

Linear model expression has the following recursive definition:

- Constant model expression is linear model expression.
- Structural variable is linear model expression.
- Let e_1 e e_2 are linear model expressions, c is constant model expression. Then expression $+e_1, -e_1, e_1 + e_2, e_1 - e_2, c * e_1, e_1 * c$, and e_1/c are linear model expressions.
- No else is linear model expression.

A.2 Model Objects

In GLPK/L model is described in terms of sets, parameters, predicates, variables, and constraints, which are called model objects.

Each model object is provided with a symbolic name, which uniquely identifies this object.

Set is a collection of elementary abstract objects called items.

Parameter is a multidimensional array built over sets. A particular element of parameter is called parameter member. Within array each parameter member is uniquely identified by its subscript list called tuple, which is an ordered sequence of items of the corresponding index sets.

Structure of n-dimensional model parameter is schematically shown in Figure below.

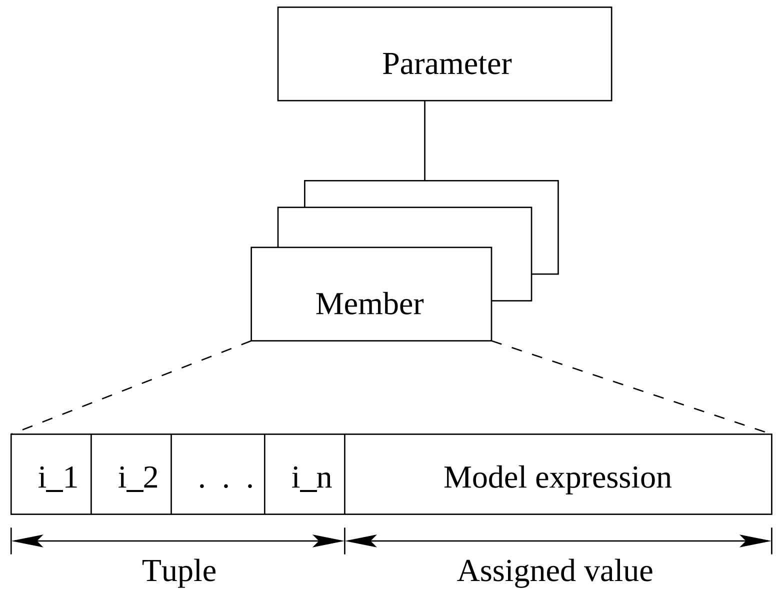


Figure A.1: Structure of n-dimensional model parameter

All members of a parameters should have different tuples, such that members with identical tuples are not allowed. In case of 0-dimensional parameter, the cartesian product formally has one element.

Predicates have the same structure as parameters. However, predicate members have no assigned values. Formally, a predicate P is the mapping $P : S_1XS_2X \ldots XS_n \rightarrow true, false$. Where $S_1XS_2X \ldots XS_n$ is the cartesian product of index sets, over which the predicate P is built, true, false is the set of boolean values.

Predicates are mainly intended for expressing relationships between items, that may be used for constructing various logical conditions.

Variables is a multidimensional array built over sets like parameters and predicates. Members of variable defines the corresponding structural variables of the problem.

Constraints is a multidimensional array built over sets. Members of constraint called elemental constraints play double role. Each elemental constraint of the problem. And each elemental constraint has an associated model expression.

The user introduces constraints into the model by means of the constraint declaration statement.

Implicit Parameters being declared each model variable and each model constraint causes three implicit parameter to be automatically introduced into the model, which are lo-parameter, up-parameter, and fx-parameter. These three parameters have the same dimension and are built over the same sets as the corresponding model object.

Implicit parameters allow the user to change types and bounds of elemental variables as well as to use the current bounds as numeric data for computing other quantities.

A.3 Using GLPK/L Models With the Solver GLPSOL

The current version of the solver GLPSOL (which is a part of the GLPK package) allows processing models written in the GLPK/L modeling language.

In order to tell the solver that the input file is a model description, the option *–lpm* should be given in the command line. For example: *glpsol –lpm file-name.lpm*. Where *".lpm"* is a recommended extension (wich means Linear Programming Model).

GLPK/L modeling language is a part of the GLPK package and can be download from Internet (freeware) to linux plataform.

Bibliography

[3rd00a] 3rd Generation Partnership Project. *Technical Specification Group Services and System Aspects; Virtual Home Environment / Open Service Architecture*, 2000. 3G TS 23.127 version 1.10.0.

[3rd00b] 3rd Generation Partnership Project. *Technical Specification Group Services and Systems Aspects; Network Architecture*, 2000. 3G TS 23.002 v3.3.0.

[AH00] Kristoffer Anderson and Daniel Högberg. System description. `www.nada.kth.se/projects/proj00/Fairway`, 2000.

[AMO93] R. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows*. Prentice Hall, 1993.

[AMOT90] R. K. Ahuja, K. Mehlhorn, J. B. Orlin, and R. E. Tarjan. Faster algorithms for shortest path problem. *Journal of the ACM*, 37(2):213–223, 1990.

[And01] Christoffer Andersson. Mobile positioning where you want to be! `www.wirelessdevnet.com`, 2001. Berkeley lab.

[BC97] J. E. Beasley and N. Christofides. Vehicle routing with a sparse feasibility graph. *European Journal of Operational Research*, 98:499 – 511, 1997.

[BHE] Nirupama Bulusu, John Heidemann, and Deborah Estrin. GPS-less low cost outdoors localization for very small devices. estrin@isi.edu. University of Southern California, Information Sciences Institute.

[Bir00] Marck Bircher. E911 phase 2 location solution landscape, 2000. Wireless Access Technology Research, Motorola Labs.

[BL01] Todd Berkowitz and Xavier Lopes. www.oracle.com, June 2001.

[BMP95] C. Brind, C. Muller, and P. Prosser. Stochastic techniques for resource management. *BT Technology Journal*, 13(1), January 1995.

[BP00] Paramvir Bahl and Venkata N. Padmanabhan. Enhancements to the RADAR user location and tracking system. Technical Report MSR-TR-2000-12, University of California at San Diego, February 2000. Microsoft Research.

[Bri97] Dennis L. Bricker. Assignment problem, 1997. Department of Industrial Engineering, University of Lowa.

[BS96a] A. G. Bruzzone and R. Signorile. Genetic algorithms and simulation as support for planning a port terminal. In *8th European Simulation Symposium*, pages 24 – 26. Society for Computer Simulation International, October 1996.

[BS96b] H. A. Buss and K. A. Stork. Discrete event simulation on the world wide web using java. In *The 1996 Winter Simulation Conference*, pages 781–785, 1996. San Diego, CA.

[BSL] D. J. Bertsimas and D. Simchi-Levi. A new generation of vehicle routing research: Robust algorithms, addressing uncertainty. In *Operations Research*, volume 44, pages 286 – 304.

[Cam99] CAMEL stage 1. www.3gpp.org, October 1999. 3GPP TS22.078.

[CC58] A. Charnes and W. W. Cooper. The theory of search: Optimal distribution of effort. In *Management Science*, pages 44–49. 1958.

[CLR90] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. McGraw-Hill Book Company, 1990. pages 558-563.

[Cur] Cambridge positioning systems CPS. `www.cursor-system.com`.

[Dav01] Brian Davis. www.sensoria.com, 2001.

[DW00] Sourav Dey and Wen-Fu Wu. The resource allocation problem, 2000. 6.034 Artificial Intelligence, Massachusetts Institute of Technology.

[ESR00] What are location services? the GIS perspective. `www.esri.com/international`, December 2000. An ESRI White Paper.

[fed] Fedex. www.feddex.com.

[Fis96] P. A. Fishwick. Web-based simulation: Some personal observations. In *The 1996 Winter Simulation Conference*, pages 772–779, 1996. San Diego, CA.

[FJG96] C. J. Fenton, W. Johnston, and J. D. Gilliland. Mobile data services. *BT Technology Journal*, 14(3), July 1996.

[FR00] John Fawcett and Peter Robinson. Adaptative routing for road traffic. In *IEEE Computer Graphics and Applications*, pages 46–53. May/June 2000.

[Fre89] Roger L. Freeman. *Telecommunication System Engineering*. John Wiley & Sons, 1989. Wiley Series in Telecommunications and Signal Processing.

[FZ94] George H. Forman and John Zahn Zahorjan. The challenges of mobile computing. Technical Report UW CSE # 93-11-03, Computer Science & Engineering, March 1994.

[Gei90] D. Geist. *Semantic Control in Continuous Systems: Applications to Aerospace Problems*. PhD thesis, Washington University, 1990.

[Gm91] Gilbert Got-man. NHS in the taxi industry: Computer dispatch. IVHS Annual Meeting., March 1991.

[Gor78] G. Gordon. *System Simulation*. Englewood Cliffs, Prentice Hall, 1978.

[Gra00] Gravitate plataform for location-precise services. `www.grvt8.com/docs/whitepapers.pdf`, October 2000.

[Gre00] Dennis Gregorovic. Market-based mechanisms for allocating agents to location-based tasks. Master's thesis, Departament of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, May 2000.

[Hay00] Stephen Hayes. Location services just a matter of time! `www.3gpp.org`, February 2000. ETSI Mobile News, GSM World Congress.

[HB01] Jeffrey Hightower and Gaetano Borriello. A survey and taxonomy of location system for ubiquitous computing. Technical Report UW-CSE 01-08-03, University of Washington, Computer Science and Engineering, August 2001.

[IK88] Toshihide Ibaraki and Naoki Katoh. *Resource Allocation Problems, Algorithmic Approaches*. The MIT Press, 1988. Massachusetts Institute of Technology.

[IMo] All about i-mode index. www.nttdocomo.com/i/.

[Int00] Intuwave unveils mobile taxi hailing service. `www.intuwave.com`, 2000.

[Jef91] Parker Jeffrey. Mobility management and market oriented local transportation., March 1991. Final report of the International Taxi and Livery Association to the Urban Mass Transportation Administration.

[JKV98] Cor Jong, Goos Kant, and André Van Vliet. On finding minimal route duration in the vehicle routing problem with multiple time windows, 1998. Project ALCOM IT: Algorithms and Complexity in Information Technology, Erasmus University, Rotterdam.

[Kah96] D. K. Kahaner. Singapore's smart taxis. Technical report, Asian Technology Information Program (ATIP), Februrary 1996.

[Kal97] Richard Kalukas. Cellular telephone using GPS time synchronization. In *The Institute of Navigation GPS-97 Conference*, pages 16–19, 1997. Kansas City.

[Kit96] Hiroaki Kitano. Robocup rescue: A grand chanllege for multi-agent systems. In *ICMAS - International Conference on Multi Agent Systems*, 1996.

[KJ98] Decker K. and Li J. Coordinated hospital patient scheduling. In *ICMAS - International Conference on Multi Agent Systems*, 1998.

[Knu81] D. E. Knuth. *The Art of Computer Programming, Seminumerical Algorithms*, volume 2. Addison-Wesley, 1981.

[Koo53] B. O. Koopman. The optimum distribution of effort. In *Operations Research*, pages 52–63. 1953.

[Lai95] R Laithwaite. Work allocation challenges and solutions in a large-scale work management environment. *BT Technology Journal*, 13(1), January 1995.

[Lee61] C. Y. Lee. An algorithm for path connectivity and its applications. In *IRE (Institute of Radio Engineers) Trans. on Eletronic Computer*, volume 10, pages 346 – 365. September 1961.

[LS96] Jy-Shane Liu and Katia P. Sycara. Multiagent coordination in tightly coupled task scheduling. In *ICMAS - International Conference on Multi Agent Systems*, 1996.

[Mak01a] Andrew Makhorin. *GLPK - GNU Linear Programming Kit Version 2.4*. Department for Applied Informatics, Moscow Aviation Institute, Russia, May 2001.

[Mak01b] Andrew Makhorin. *GLPK - GNU Linear Programming Kit Version 2.4 Modeling Language GLPK/L*. Department for Applied Informatics, Moscow Aviation Institute, Russia, May 2001.

[Mal98] Thomas W. Malone. Enterprise: A marketlike task scheduler for distributed computing environments. *The Ecology of Computation, Elsevier Science Publishers B. V.*, pages 177 – 205, 1998.

[Map99] Mobile application part (MAP). www.3gpp.org, April 1999. 3GPP TS29.002.

[MC93] Brian Marsh and Fred Douglis Ramòn Càceres. Systems issues in mobile computing. Technical Report MITL-TR-50-93, Matsushita Information Technology Laboratory, February 1993.

[MJT+01] Pragnesh Jay Modi, Hyckchul Jung, Milind Tambe, Wei-Min shen, and Shriniwas Kulkarni. Dynamic distributed resource allocation: A distributed constraint satisfaction approach, 2001. Information Sciences Institute, University of Southern California.

[MLRG00] G. R. Mateus, A. F. Loureiro, R. C. Rodrigues, and O. Goussevskaia. Server location in mobile computing. In *SBRC2000* 18º *Simpósio Brasileiro de Redes de Computadores*, pages 275 – 285, Maio 2000.

[MRZZ97] J. A. Miller, S. N. Rajesh, Z. Zhang, and H. Zhao. JSIM: A Java-based simulation and animation environment. In *Annual Simulation Conference*, April 1997. Atlanta, GA.

[Mul94] N. J. Muller. *Wireless Data Networking*. Artech House, 1994.

[Nil98] Nils J. Nilsson. *Artificial Intelligence: a new synthesis*. Morgan Kaufmann, 1998.

[NMZ96] R. Nair, J. A. Miller, and Z. Zhang. A java-based query-driven simulation environment. In *The 1996 Winter Simulation Conference*, pages 786–793, 1996.

[Pat00] Vikash Patel. A study of how location-based services can be provider users. Master's thesis, Departament of Information Systems and Computing, Brunel University, September 2000.

[PKK01] Gunther Pospischil, Harald Kunczier, and Alexander Kuchar. LOl@: a UMTS location-based service, 2001. Mobilkom Austria and Forschungszentrum Telekommunikation Wien.

[PQTC01] Michel Pasquier, Hiok-Chai Quek, Boon Tee Tan, and Chee Keong Chan. Intelligent taxi deployment system, 2001.

[Pub] Módulo de rastreamento de veículo (AVL) do guia inteligente publintel - via satélite, telefonia e rádio. `http://www.publintel.com.br/ModuloRastreamento.html`.

[RBP+91] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen. *Object-oriented modeling and design*. Englewood Cliffs, Prentice Hall, 1991.

[Rul95] Kevin Scott Ruland. *Polyedral Solution to the Pickup and Delivery Problem*. PhD thesis, Washington University, Server Institute of Technology Departament of Systems Science and Mathematics, August 1995.

[Sam98] Rudini Menezes Sampaio. Estudo de implementação de algorithmos de roteamento. Divisão de Ciência da Computação, Centro Técnico Aeroespacial, Instituto Tecnológico de Aeronáutica, 1998.

[San01] Sanders. Ecm chanllege problem. www.sanders.com/ants/ecm.htm, 2001.

[Sei88] A. F. Seila. SIMMTOOL: A software tool kit for discrete event simulation in pascal. In *Simulation*, volume 3, pages 93–99. 1988.

[SG01] H. J. Sebastian and T. Grnet. Introduction to the minitrack: Intelligent systems in traffic and transportation. In *34th Hawaii International Conference on System Science*, 2001.

[SGN92] John R. Stone, Gorman Gilbert, and Anna Nalevanko. Assessment of computer dispatch technology in the paratransit industry. Technical report, Institute of Transportation Research and Education, University of North Carolina, 1992.

[SHH92] W. Spendley, G. R. Hext, and F. R. Himsworth. Technometrics, 1992.

[Sie00] Intelligent networks end-to-end solutions for specific industry sector. `www.siemens.com`, 2000.

[SIG93] SIGMOD Record. *Data Management for Mobile Computing*, volume 1, March 1993.

[Sim] Mobile sim toolkit. www.mobilesimtoolkit.com/technical.asp.

[Sim69] H. A. Simon. *The Science of the artificial.* Cambridge: MIT, 1969.

[SM] Pèter Stefá and László Monostori. On internet routing problem in dynamically changing environment. Computer and Automation Research Institute of Hungarian Academy of Sciences.

[Swe99] Göran Swedberg. Ericsson's mobile location solution. Technical Report 4, Ericsson, 1999.

[Tei95] José Eustáquio Teixeira. Planejamento de sistemas de telefonia móvel celular. Master's thesis, Universidade Federal de Minas Gerais - Departamento de Ciência da Computação, 1995.

[TOA00] Time difference of arrival technology for locating narrowband cellular signal. `www.trueposition.com/tdoa.htm`, 2000.

[Tru00] Carrier and and-user application for wireless location systems. `www.trueposition.com/spie/_app.htm`, 2000.

[Van00] Anne Vanhala. Virtual home environment / open service architecture, 2000. Advanced Wireless Communication Systems, University of Helsinki, Departament of Computer Science.

[Wal98] John Walker. *Advances in Mobile Information Systems.* Artech House, 1998.

[Wal99] John Walker. *Mobile Communications Series.* Artech House Publishers, 1999.

[WAP] Wireless application protocol version 1.1. `www.wapforum.org`.

[Web00] Mobile internet application plataform for delivering turnkey location-based services. www.webtrac.net, 2000.

[Wil70] N. H. M. Wilson. *Dynamic Routing: A study of assignment algorithms*. PhD thesis, Massachusetts Institute of Technology, Departament of Civil Engineering, 1970.

[XL] Jianliang Xu and Dik Lun Lee. Quering location-dependent data in wireless cellular environment. Hong Kong University of Science and Technology.

[ZG] M. Zaffalon and L. M. Gambardella. A network design approach to the allocation of resources in an intermodal terminal. Technical report.

[Zha97a] Zhiwei Zhang. A Java-based simulation and animation environment: Jsim's foundation library. Master's thesis, University of Georgia, 1997. Athenas, Georgia.

[Zha97b] Yilin Zhao. *Vehicle Location and Navigation Systems*. Artech House, 1997. Intelligent Transportation Systems.

[Zhu01] Kenny Qili Zhu. A reactive method for real time dynamic vehicle routing problem. Master's thesis, Dept. Computer Science, National University of Singapore, 2001.

[ZRGM98] Marco Zaffalon, Andrea E. Rizzoli, Luca Maria Gambardella, and Monaldo Mastrolilli. Resource allocation and scheduling of operations in an intermodal terminal. In *10th European Simulation Symposium and Exhibition, Simulation in Industry*, pages 520 – 528, October 1998. Nottingham, United Kingdom.