UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

An initial solution heuristic for the vehicle routing and scheduling problem

Johannes Wilhelm Joubert

A dissertation in partial fulfillment of the requirements for the degree

Master of Engineering (Industrial Engineering)

in the

Faculty of Engineering, Built Environment, and Information Technology

University of Pretoria

November 2003

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Abstract

AN INITIAL SOLUTION HEURISTIC FOR THE VEHICLE ROUTING AND SCHEDULING PROBLEM

JOHANNES WILHELM JOUBERT

Supervisor:	Professor S.J. Claasen
Department:	Industrial and Systems Engineering
University:	University of Pretoria
Degree:	Master of Engineering (Industrial Engineering)

South Africa provides a fascinating interface between the developed and the developing world and poses a multitude of opportunities for enhancing the sustainable development of local cities. The concept of *City Logistics* is concerned with the mobility of cities, and entails the process of optimizing urban logistics activities by considering the social, environmental, economic, financial, and energy impacts of urban freight movement. Vehicle routing and scheduling has the potential to address a number of these key focus areas. Applying optimization to vehicle routing and scheduling results in a reduced number of trips, better fleet utilization, and lower maintenance costs; thereby improving the financial situation of the fleet owner. Improved fleet utilization could have a positive environmental impact, while also improving the mobility of the city as a whole. Energy utilization is improved while customer satisfaction could also increase through on-time deliveries and reliability.

The *Vehicle Routing Problem* (VRP) is a well-researched problem in Operations Research literature. The main objective of this type of problem is to minimize an objective function, typically distribution cost for individual carriers. The area of application is wide, and specific variants of the VRP transform the basic problem to conform to application specific requirements. It is the view of this dissertation that the various VRP variants have been researched in isolation, with little effort to integrate various problem variants

into an instance that is more appropriate to the South African particularity with regards to logistics and vehicle routing.

Finding a feasible, and integrated initial solution to a hard problem is the first step in addressing the scheduling issue. This dissertation attempts to integrate three specific variants: multiple time windows, a heterogeneous fleet, and double scheduling. As the problem is burdened with the added constraints, the computational effort required to find a solution increases. The dissertation therefor also contributes to reducing the computational burden by proposing a concept referred to as *time window compatibility* to intelligently evaluate the insertion of customers on positions within routes.

The initial solution algorithm presented proved feasible for the integration of the variants, while the *time window compatibility* decreased the computational burden by 25%, and as much as 80% for specific customer configurations, when using benchmark data sets from literature. The dissertation also improved the quality of the initial solution, for example total distance travelled, by 13%. Finding an initial solution is the first step in solving vehicle routing problems. The second step is to improve the initial solution iteratively through an *improvement heuristic* in an attempt to find a global optimum. Although the improvement heuristic falls outside the scope of this dissertation, improvement of the initial solution has a significant impact on the quality of improvement heuristics, and is therefor a valuable contribution.

Contents

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

List of Figures

List of Tables

Chapter 1

Research problem

South Africa's level of urbanization closely follows international trends in developed countries, with the highest level of economic activity focused in a few metropolitan areas; attracting both people and investments. The good functioning of these metropolitan areas is of strategic importance to the country, as these areas are the main focus for economic and social development. The level of service of transport provided impacts directly on the efficiency and the quality of the development in the metropolitan areas. South African metropolitan areas are experiencing rapid growth, and are having difficulties in controlling the physical urban expansion. Both public and freight transport costs are negatively impacted by these phenomena. As demand for transport increases faster than the supply of these services, commuting and freight transportation costs increase at a rate higher than inflation. The community at large also experiences the higher expenses required to support demands for more extensive infrastructure and services.

1.1 Motivation for the project

South Africa provides a fascinating interface between the developed and the developing world. In a critical review, Leinbach and Stansfield [28] have emphasized that Industrial Engineers should re-adopt a systematic view. They argue that the perception of Industrial Engineers has been negatively impacted by their ability to model the obvious, and in the over-simplification of their models, to the extent that the reality is not represented comprehensively. Industrial engineers should therefor appreciate the complex and intertwined relationships between social, political, and economic factors influencing urban freight transport systems.

Freight carriers are sharing the road network with various modes of public transport. The use of private vehicles have rapidly increased. The increase can be attributed to both an increase in the number of trips undertaken, and increased journey lengths [5, 48]. Road network performance is negatively impacted by the higher usage of private vehicles and results in higher levels of congestion, and a significant reduction in operating speeds. Public transport performance is impacted negatively when operating speeds decrease, resulting in increased operating costs for the carriers, and thus impacting negatively on its attractiveness. As a result, the economically able part of the population turn to their private vehicles for a reliable source of transport, and unknowingly contributes to the hyper-congestion phenomenon.

The concept of *City logistics* has emerged to address a new area in transport planning. The objective of the concept is to support the sustainable development of cities and to address challenging problems such as high levels of traffic congestion, negative environmental impacts, high energy consumption and a shortage of trained labour. Taniguchi *et al.* [57] states that *City logistics* is the process of totally optimizing urban logistics activities by considering the social, environmental, economic, financial and energy impacts of urban freight movement. In an earlier article Taniguchi *et al* [55] emphasizes that the optimization drive is focussed on private shippers and carriers in a free market economy, although consideration is given to the costs and benefits for both the private and public stakeholders. Usually one or more of the following initiatives are included:

- *Advanced information systems* utilize the ever-increasing computational ability to analyze and rationalize existing logistics operations.
- *Co-operative freight transport systems* allow for a reduced number of vehicles in the system by means of load consolidation.
- *Public logistics terminals* are implemented with success in the northern hemisphere. These terminals are typically located on the outskirts of cities, and are helpful promoters of the co-operative transport systems mentioned previously.
- *Load factor controls* are also new initiatives where certificate systems are introduced for carriers. The concept has been successfully implemented for freight vehicles in parts of Europe. Service vehicles, such as telecommunication service providers, refuse removal, plumbing and electrical contractors, are dealt with ineffectively. Service vehicles transport *service providers*, as opposed to *freight*. The objective of service vehicles is to provide a reliable service, either on schedule, or when requested. Load factor controls, as a performance measure, are therefore ineffective. The load factor initiative aims to reduce the number of vehicles in congested central business districts by promoting load consolidation. This results in higher fleet utilization, and thus reducing the number of large freight vehicles with small loads.

- *Underground freight transport systems* are innovative, yet costly solutions for freight transport problems. Ooishi and Taniguchi [40] evaluate a proposed initiative in Tokyo, and conclude that the overall effects upon the society and the economy is positive.

It is important to quantify the consequences of such *City Logistics* initiatives as this will enhance the evaluation of the significance and benefits thereof, especially when designing new, or improving existing, urban infrastructure and freight transport activities. Models, representing the various stakeholders and their particular objectives, should be used to quantify the changes in logistics costs, traffic congestion, fleet utilization, hazardous gas emissions, accident occurrences, etc. of proposed initiatives.

1.2 Overview of the subject

The existing urban structures in South Africa have been brought about by apartheid policies over many generations. Spence [48] argues that the extent of the separation between people – black and white, rich and poor, advantaged and disadvantaged – and the resources and opportunities which they require, has produced urban conditions that are morally, socially, politically, and economically, unsustainable. The separation resulted in inequitable spatial development and economic structures that favored growth in existing well-resourced areas. More specifically, the urban land use disposition produced low-density residential development and urban sprawl with opportunities concentrated in the vicinities of the more affluent and privileged areas. Conversely, the majority of the population are settled in remote areas with few opportunities or social amenities. Lipman and Monaghan [30] elaborate on the spatial planning, indicating that the legacy from past spatial planning policies has resulted in long travel distances and insufficient residential densities for effective transport services. With the national transport policies under revision, a deliberate focus on transport is essential [34]. Transport in itself is a key factor in creating sustainable economic growth.

1.2.1 Stakeholders

Various key stakeholders participate in the economy and often have competing and egocentric objectives.

Residents

The community, or *residents*, are the people that live, work, shop, and entertain in the metropolitan areas. Their objectives include minimizing traffic congestion, noise, air pollution due to traffic, and traffic accidents. Residents do not welcome large freight carrying vehicles in residential areas [42]. Nevertheless, these carriers are required as residents have an expectation to

Table 1.1: Distances and average time spent in commuting in the main cities

	Distance between black township(s) & CBD (in kms)	**Average time spent in transport (minutes per journey)**
Johannesburg (bus \ train)	20.0	77
Johannesburg (taxi \ cars)	20.0	44
Pretoria	52.0	75
Durban	20.0	n.a.
Bloemfontein	58.4	86
Port Elizabeth	16.1	n.a.
East London	21.4	n.a.
Cape Town	18.8	65
Average	28.3	69

receive their commodities at convenience stores scattered all over residential areas. The South African transport system, from the residents' frame of reference, is a dual system. For the more affluent portion of the population the system imitates the American way of life with houses, malls, services and offices distributed abundantly, necessitating automobile use, as they are geographically dispersed. The malls are small, and do not offer the shopper a comprehensive range of products. This is in contrast with European models where cities have a central business district, offering the visitor a full range of products and services in a small area, accessible on foot, and in close proximity of a well-established public transport system. On the other hand, given the distorted spatial structure, a large portion of the urban population simply aspires to minimize their commuting expense and ensure their safety while commuting. Here the less affluent move through various modes, from on-foot transportation or bikes, to busses, minibus taxis, commuter trains, and to a lesser extent cars and trucks. Saint-Laurent [18] presents the distances and average time spent commuting by commuters in South-Africa in table 1.1.

Carriers

Carriers represent both public and private stakeholders executing the logistic and distribution functions. The *cargo* is not limited to freight, but also encompasses passengers in the form of public transport. Freight carriers are continuously expected to provide higher levels of service at lower rates, and therefor try to minimize their logistic costs, and maximize their profits.

Modes of public transport in South African urban areas are limited to *commuter trains* (9%), *busses* (16%), *minibus taxis* (24%), and *cars* (50%).

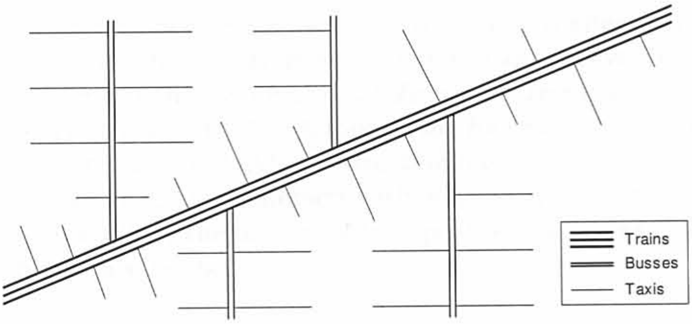

Figure 1.1: The ideal mode configuration

The values in brackets indicate the market share of each mode with respect to urban trips, based on the number of passengers multiplied by the distance travelled [18]. These different modes of public transport are often in competition for passengers. This is in contrast to the ideal of being cooperative and integrated. An ideal configuration would see trains forming the backbone of the network and supplying the major (primary) routes; busses on secondary routes, and taxis on local routes; feeding into primary and secondary routes, as is indicated in figure 1.1.

Shippers

Shippers are the customers of carriers, and often receive (or send) goods from (to) other shippers, or residents. Examples of shippers include manufacturing plants, wholesale and retail outlets, and mail centers. The objective of the shippers is to maximize their level of service, which can be a function of cost, reliability, and/or traceability. Shippers place requirements on carriers for specific collection and delivery times. These requirements are referred to as time windows.

Administrators

Administrators represent local, provincial, and national government whose objective is to resolve conflict between stakeholders involved in urban freight transport, while facilitating sustainable development of urban areas.

Transport authorities are responsible for planning, coordination, implementing, monitoring, funding and applying law-enforcement of land transport in provincial and local government spheres. Traffic densities and road construction costs are much higher in urban areas. Municipal resources for transport is also severely restricted due to issues such as housing, safety, and education that enjoy preference in the urbanized environment. Public transport has a better cost and space effectiveness for mass transportation.

Public-transport should therefor be considered a serious alternative in urban areas. Not only do administrators appreciate the alternative, but they also directed the focus of the *National Land Transport Transition Bill* towards public passenger transport [37] . Although the freight transport industry is largely deregulated in South Africa, attention has been given to freight operations in urban areas, as it interacts with other modes and traffic streams. Nothnagel [34] outlines the history of transport legislation as it progressed since South Africa's independence in 1994.

Regulation of urban freight transport is a powerful tool for improving the efficient use of transport networks and infrastructure. Numerous regulations have been implemented internationally [55], and include:

- Weight limits
- Load factor control
- Designated routes
- Zoning
- Time windows

A recent, and controversial, event has seen the mayor of London introduce a *congestion charge* for all vehicles entering the central district [7]. Baseman [6] illustrates that congestion is neither a new, nor a unique problem to our century, and mentions that Julius Caesar placed limits, and raised taxes, on the number of vehicles entering Rome in A.D. 125.

1.2.2 The *City Logistics* approach

It is appropriate to distinguish between the various *impacts* that the transport network and infrastructure can have on the stakeholders identified in section 1.2.1.

- *Financial* impacts relate to, but is not limited to, commuting costs for residents; the payback period of investments that shippers consider making in establishing facilities in new locations; fuel and fleet costs for shippers; and *internal rate of return* (IRR) for administrators investing in public transport capacity such as train wagons and busses.
- *Social* impacts relate to equity for various user and non-user groups of transport, and could include the impact of accidents, accessibility to transport, or even business competition in the case of taxi operators servicing the same area.

- *Economic* impacts are more comprehensive than financial impacts. Cost benefit analysis does not simply focus on the immediate financial implications, but rather the viability of a transport scheme over the period of the scheme's entire life. *Costs* could include the upfront acquisition of capital equipment and maintenance costs throughout the operational life, while *benefits* could represent the reduced travel times experienced by commuters or reduced fleet operating costs experienced by carriers on items such as consumables and maintenance.

- *Energy* conservation is becoming more important since there is a limited amount of natural resources. Petroleum and automotive diesel oil are the two main sources of energy for the transport industry. *City Logistics* initiatives, such as *route optimization*, could potentially reduce the total amount of fuel consumed by freight and public transport vehicles, if the objective of the exercise is to reduce route lengths.

- *Environmental* effects pose a direct risk on human health. Greenhouse gasses produced by vehicles constitute of the following pollutants:

  - Carbon monoxide (CO)
  - Oxides of nitrogen (NO_x)
  - Suspended particulate material (SPM)
  - Hydrocarbons (HC)

  Noise pollution is also an area of concern, especially in urban areas.

To be able to limit the scope of an investigation or optimization exercise, Taniguchi *et al* [57] introduce a third *spatial* boundary to the transport system, as figure 1.2 indicates.

A *terminal* refers to a single location, node, or venue, in the transport system, for example a distribution center or a bus stop. The infrastructure along which carriers move between two terminals are referred to as a *link*. Examples could include a bus route between two stops, or a segment of rail track between two stations. Multiple links make up an *area*, for example Centurion in Gauteng, South Africa. A *corridor* refers to a number of directly-connected areas. The Mabopane-Centurion corridor would serve as an example. It is significant to distinguish between the various spatial elements, as each element holds unique improvement opportunities.

To address problems, or opportunities for improvement, models could be created that is used to represent the transport system, and predict the effects of proposed solutions and improvements. Taniguchi *et al.* [55] elaborate on three distinct types of models:

- *Supply models* are created when the performance of the road system is represented or predicted through, for example, simulation models.

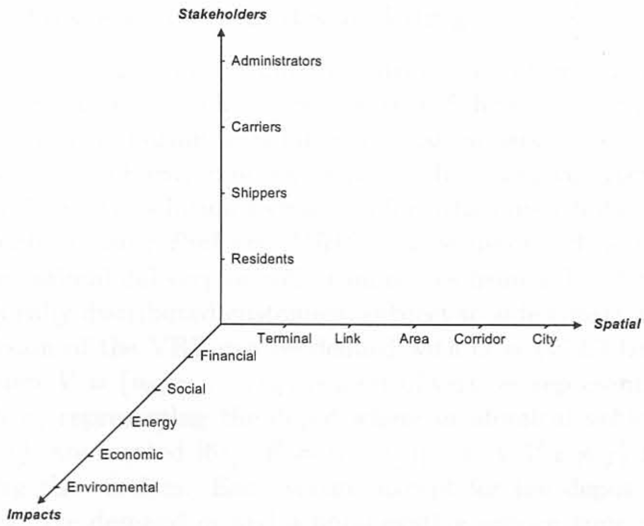

Figure 1.2: System boundaries

These models include the various cost factors associated with routes, vehicles, and terminals. Supply models are created with the suppliers' objective in mind, such as minimizing outbound logistic costs, distribution strategies for a given network, etc.

- *Demand models* focus on commodity flows and vehicle trips. Models include mode choice (e.g. rail, road, etc.) that is based on commodity (freight) generation; distribution of freight; and vehicle routing and scheduling.

- *Impact models* are used to estimate and predict various environmental, economic, and social impacts that a transportation system might have on stakeholders.

In establishing the scope of a *City Logistics* project, it is therefore necessary to identify the stakeholders that will form the object of the study. The next step would be to determine the need of the stakeholders, and thus identify the relevant impacts under consideration. Once these foci have been set, an appropriate model can be developed.

1.2.3 Vehicle routing and scheduling

Vehicle routing and scheduling problems are well-researched in the field of *Operations Research*. The main objective of these types of problems are to minimize the distribution costs for individual carriers. Given the complexity of the type of problem, extensive research has been conducted to develop exact and heuristic solution techniques for urban distribution problems.
The *Vehicle Routing Problem* (VRP) can be described as the problem of assigning optimal delivery or collection routes from a depot to a number of geographically distributed customers, subject to side constraints. The most basic version of the VRP can be defined with $G = (V, E)$ being a directed graph where $V = \{v_0, v_1, \ldots, v_n\}$ is a set of vertices representing customers, and with v_0 representing the depot where m identical vehicles, each with capacity Q, are located [61]. $E = \{(v_i, v_j)|v_i, v_j \in V, i \neq j\}$ is the edge set connecting the vertices. Each vertex, except for the depot $(V \backslash \{v_0\})$, has a non-negative demand q_i and a non-negative service time s_i. A distance matrix $C = \{c_{ij}\}$ is defined on E. In some contexts, c_{ij} can be interpreted as travel cost or travel distance. Hence, the terms distance, travel cost, and travel time are used interchangeably. The VRP consists of designing a set of m vehicle routes having a minimum total length such that

- each route starts and ends at the depot,
- each remaining vertex $(V \backslash \{v_0\})$ is visited exactly once by one vehicle,
- the total demand of a route does not exceed Q, and

- the total duration (including service and travel time) of a route does not exceed a preset limit L

The VRP is a hard combinatorial (*np*-hard) optimization problem for which Laporte [27] has indicated several exact and approximate solution algorithms. An *np*-hard problem implies that the solution space will increase at an exponential or factorial rate (non-polynomial) as the number of customers/vertices increases. Early researchers such as Clarke and Wright [14] realized that exact algorithms can only solve relatively small problems, but a number of heuristic (near-optimal) algorithms have proved very satisfactory.

The basic VRP makes a number of assumptions, including utilizing a homogeneous fleet, a single depot, one route per vehicle, etcetera. These assumptions can be eliminated by introducing additional constraints to the problem. This implies increasing the complexity of the problem, and, by restriction, classifies the extended problem as an *np*-hard problem. The various side constraints, and their application, are discussed in more detail in chapter 2. It should be noted that most of these additional side constraints are implemented in isolation, without integration, due to the increased complexity of solving such problems.

1.3 Research questions

The concept of *City Logistics* is as important in South Africa as it is in the rest of the world. To contribute to the school of subject knowledge, the following general research question could be asked.

> *How can we "Glocalize"*[1] *the concept of* City Logistics *in South African urban environments?*

Urban environments are complex societies with various stakeholders. Although the perceptions and motivation of stakeholders vary, their objective towards freight movement is uniform: civilize the truck. One way of achieving the objective is to reduce the number of freight vehicle trips. Residents will appreciate the alleviated congestion; shippers welcome reduced shipping and receiving transactions; and carriers are happy to reduce their fleet operating costs as a result of the reduced number of trips. Trip optimization is a key focus area of *City Logistics*, and vehicle routing and scheduling procedures provide core techniques in the modelling of *City Logistics*. Most practical distribution problems are not limited to occurrences of isolated VRP variants. No literature, however, could be found that integrate an appropriate number of these variants. To make the proposed demand model for carriers more applicable to the South African environment, the aim of this dissertation is to answer the following research question:

[1]To have a problem, or opportunity, with global relevance, yet with local application.

> *Is it possible to solve a vehicle routing problem with multiple integrated constraints?*

The dissertation aims to investigate the feasibility of integrating a number of these side constraints into a single problem instance, referred to as the *Vehicle Routing Problem with Multiple Constraints* (VRPMC).

1.4 Research design and methodology

Mouton [33] notes that theory-building studies aim to develop new models to explain particular phenomena. The *phenomena* in this particular dissertation would be the routing of vehicles to service a number of geographically dispersed customers. The VRP and its variants are well-researched problems in academic literature but, as stated, lack integrated models. One of the limitations of model-building studies are their ineffectiveness if the models make implausible claims on reality. To ensure that the most common trap – over-abstract formulation – is avoided, figure 1.3 illustrates the proposed *Operations Research* process, as adapted from Rardin [43].

The objective of the dissertation is to address a particular *problem*; both empirical data and actual problem instances exist that could be used to understand the actual problem and its underlying assumptions. Given the problem and the available literature, a representative *model* could be formulated. The model will consist of:

- *decision variables* that will indicate the decisions (solution space) open to decision makers,
- a set of *constraints* that limit the decision choices, and
- an *objective function* that will indicate preferred decisions.

The modelling task includes the process of evaluating the assumptions made in literature, and criticizing the assumed reality the model is based on. Figure 1.4 from Taha [49] illustrates the modelling task appropriately.

It is believed that, as is the case for most VRP variants, the modelling of the problem is relatively simple. In the case of the VRP variants, the *model* is only the first step of the *Operations Research* process. *Solving* the model involves finding an initial solution, and then optimizing the solution until a local, or optimistically, a global, optimum is found. The objective of this dissertation is to document the initial solution algorithm, and to code the algorithm in an appropriate language, such as *MATLAB*, for testing purposes using empirical data. To make conclusions from the results, it is necessary to evaluate the quality of the results generated. Solomon [46] developed sets of test problems representing various environments. These test sets have become a popular benchmark for result comparisons. Although

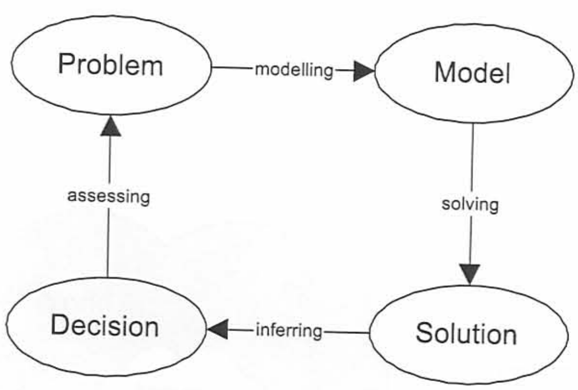

Figure 1.3: *Operations Research* process

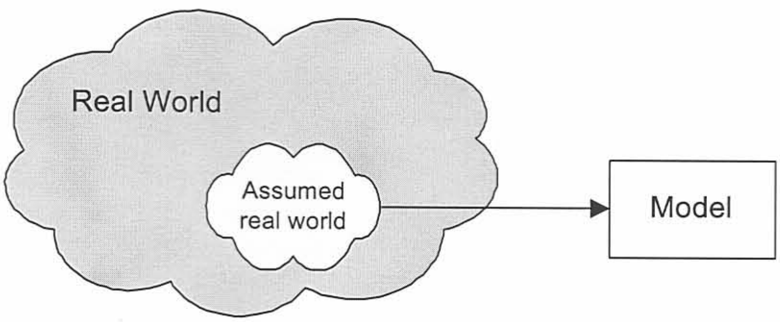

Figure 1.4: Levels of abstraction in model development

the sets may not adhere to the perceived model reality of the dissertation's proposed model, it could be used as point of departure when developing test sets for the model. Only once the model *solutions* are verified and validated, could they be used to support *decisions* regarding the problems experienced in reality.

1.5 The project outline

A comprehensive two-part literature review is undertaken in Chapter 2. The aim of the review is to firstly introduce the concept of *City Logistics* with specific reference to the South African particularity.
The latter part of the literature review elaborates on vehicle routing and scheduling, focussing on the VRP variants, and finding an appropriate point of departure for the proposed model.

Due to the complex nature of the proposed model, and as a result of additional constraints introduced to the model, a new concept of *Time Window Compatibility* is introduced in Chapter 3. The concept serves as a means to reduce the computational complexity of the solution algorithm. In Chapter 4 the proposed model is defined, and the initial solution algorithm introduced.

Results are presented in Chapter 5, followed by interpretation and related discussions. The dissertation is concluded with a research agenda for *City Logistics* studies and applications.

Chapter 2

Literature Review

The aim of the review is to establish the broader environment within which the optimization of logistic operations occur. The process of *City Logistics* is introduced. The multi-actor environment is emphasized, where a number of non-transport influences affect the transport activity within cities. The need exists to describe and predict influencing factors in the urban network. For this purpose a new modelling approach is introduced that takes the various perceptions of stakeholders into account.
Vehicle routing and scheduling procedures are formalized as core techniques to model the logistic system in urban areas. Realistic variants of the vehicle routing problem are introduced, and their impact on the modelling task is highlighted.

The complex nature of vehicle routing problems requires the introduction of heuristic solution algorithms. The second half of the chapter reviews the process of finding a good initial solution – a critical parameter in the solution quality of the optimization process. Finally, three improvement heuristics are introduced that proved successful in vehicle routing problems.

2.1 Fundamental concepts

Transportation, as the business of conveying passengers and/or goods [2], has evolved tremendously in South Africa over the past century [1]. Transportation occurs even in the absence of infrastructure: nature transports seeds and natural materials through forces, such as wind and water; people in remote areas travel by foot in the absence of accessible road surfaces, and get from point A to point B, even across inhospitable areas. It is, however, as a result of the frequent use of transport routes that the users of the routes collectively demand improvements. The improvements apply to both travel conditions, such as shorter journeys and smoother rides, and travel reliability, for example the accessibility of roads due to weather conditions.

It is also not uncommon for users to pay for improvements. The Chair of Transportation Engineering elaborates on the evolution of transportation engineering as a discipline [38]. The demand for improvements gave rise to the development of transport infrastructure such as hardened road surfaces, bridges, ports, etc., and established disciplines such as structural, pavement, railway, and traffic engineering.

The discipline of traffic engineering is based on developing a mathematical understanding of the operation of vehicles on sections of the road network. This ability to model vehicular movement is extended when the need emerge to forecast *what* transport facilities would be required in the future. The discipline of transport planning, that applies to single and multi-modal problems, emerged. It can also be used to understand the inter-relationship of land-use and the demographic aspects that generate the need for transportation.

2.2 Modelling City Logistics

It is appropriate to elaborate on a few basic transport concepts to emphasize the complex relationships between various stakeholders in society.

2.2.1 Transport Concepts

The function of transport is to move passengers and goods from where they are to where they want to be or where their relative value is greater. The demand for transport arises from the fact that not all places are equally endowed with resources, and surpluses are then moved to areas experiencing shortages [39]. Transport is a service and does not occur for its own purpose, and is therefor referred to as a *derived demand.* In the development of industries, transport plays a vital part in linking the sources of raw material, manufacturing or processing centers, and the markets. Raw material are moved to, and between processing centers, while finished goods are moved via wholesalers and retailers to the point of consumption or utilization. Transport is essential to enable people to travel between their homes and places of employment. This is even more true in South Africa where the average commuting time for residents is in excess of one hour (Table 1.1).

Transport and development are closely linked, and effective transport is a prerequisite for the development of a country. Investment alone, however, does not guarantee prosperity. Whilst *transport* is focussed on the physical movement of objects, be it freight or passengers, *logistics*, in the context of this dissertation, is concerned with the activity of transport within a larger environment. The environment of the logistics system is illustrated in figure 1.2. The *City Logistics* process has numerous interfaces with various built environment, development, engineering, and geographical disciplines. It is concerned with the *mobility*, *sustainability*, and *liveability* of cities [56]. A few non-transport aspects of logistics is explained:

Economic aspects of transport

Transport cost consists mainly of a fixed portion, C_f at a point A, and a variable portion based on the distance travelled from point A to a point B, as indicated in figure 2.1. The cost function need not be linear as indicated,

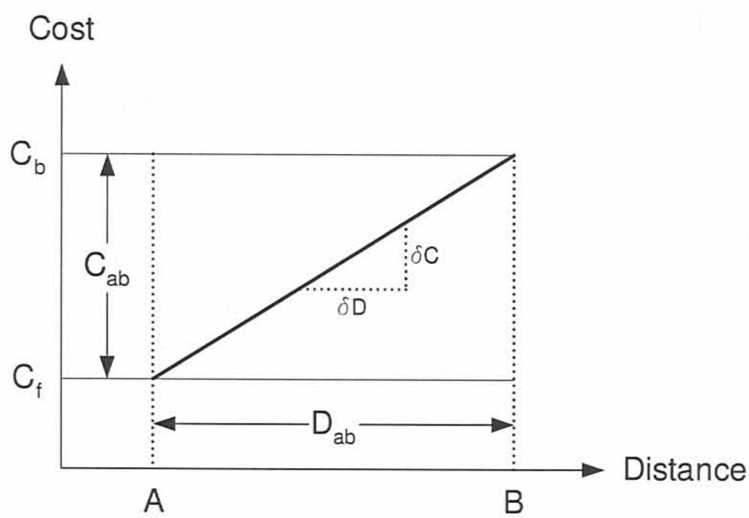

Figure 2.1: Transport cost components

and the slope of the cost function could be calculated as $\frac{\delta C}{\delta D}$. The transport cost between points A and B, C_b, can be calculated as $C_b = C_f + D_{ab} \times \frac{\delta C}{\delta D}$. The transport cost function is influenced by improvements in transport facilities and infrastructure, e.g. improved transhipment methods, more efficient vehicles, and optimized vehicle routes.

Competing carriers and shippers react differently to these advances, and could gain a *competitive advantage* by effectively implementing the advances. Consider the example illustrated in figure 2.2 where two carriers, A and B, compete on the basis of cost. At present, carriers A and B supply their services at a cost of C_{a1} and C_{b1} respectively. A technological advance becomes available to both carriers, and carrier A increases the fixed portion of its transport cost by investing in the advance – to a greater extent than carrier B. This results in carrier A being able to compete in the market at a cost of C_{a2}, lower than carrier B's cost of C_{b2}.

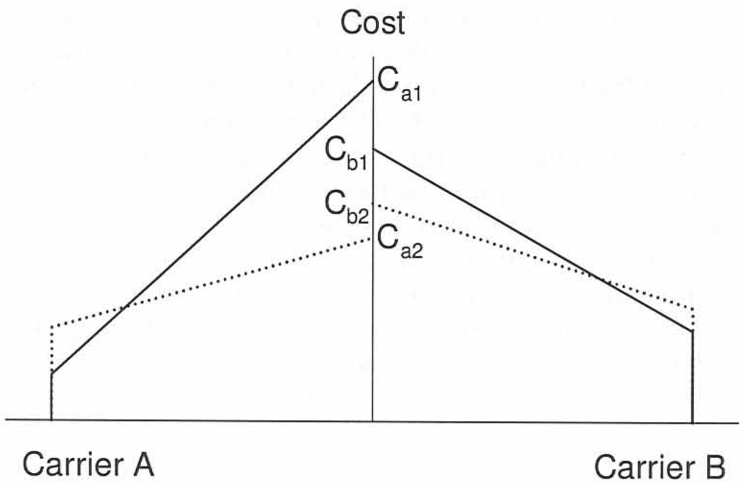

Figure 2.2: Effect of transport advances on comparative advantage

Economies of scale influences the cost per unit transported, and is illustrated in figure 2.3. This can be achieved should if a carrier is able to

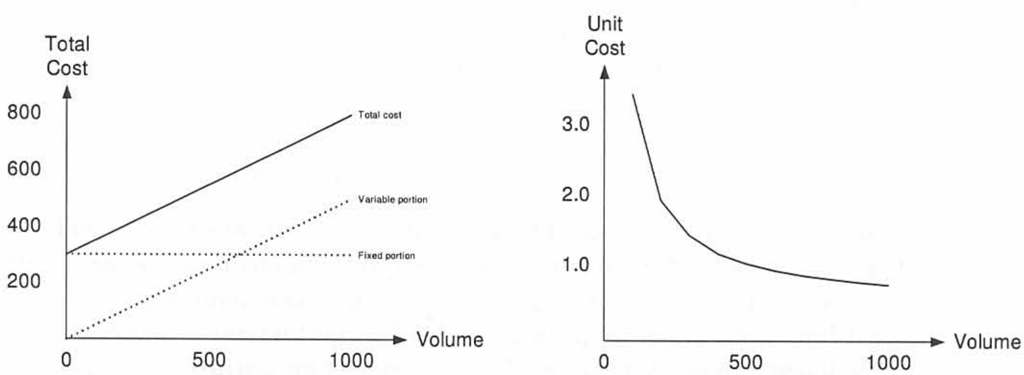

Figure 2.3: Fixed and variable costs

consolidate loads on a vehicle, or acquire new business for that vehicle.

Although transport services is generally initiated by the private sector, the concept of *public-private-partnerships* have gained prominence, and could benefit both sectors. It is important to preserve transport through joint initiatives. The term *preserve* is used to emphasize the perishable nature of the service. Once a vehicle, such as a delivery van, or a bus, have departed with a load factor less than 1, the opportunity to sell the *empty*

seats for that trip is immediately, and permanently, destroyed.
The South African government has introduced commercialization and privatization policies to share the control over transport with the private sector. Examples include the creation of the *Airports Company of South Africa* and the concession of rail services to private companies such as *Metro-Rail* [36, 37]. Government subsidizes various aspects of transport activities in both the public and private sector. Bus and rail subsidies amount to R3,737 million in 2002/03, which include maintenance, as well as the creation of additional infrastructure [15]. Government is implementing a *Taxi Re-capitalization Programme* in the private sector whereby economic potential is unlocked in a taxi industry that provides 65% of the 2.5 billion annual passenger trips in urban areas [8, 12, 15].

Political aspects of transport

The relationship between government and transport is bidirectional. As a *user* of transport, the government communicates its decisions to all areas. This happens indirectly when provincial and local government officials travel between parliament and their representative offices. Government also uses transport to move its defence force to defend the county's borders. Transport is used more strategically to achieve developmental goals in the country, and to achieve political goals in the form of incentives and investments to attract industrial parties to settle in specific areas. As a transport *influencer*, government provides funds for infrastructure, and supports innovations through pilot and demonstration projects.

Social aspects of transport

The main social impact of improved transport is the reduced friction that results from distance: people are given the opportunity to participate, socially and economically, in a larger geographical area. This results in various cultures interacting and communicating different ideas and frames of reference. Mutual understanding is fostered that should result in the elimination of suspicion between races and/or cultures. The improvement of transport, specifically air travel, has widened the scope of opportunities between nations to the extent that the world is often referred to as a *global village*.

Developments in the automotive industry, and the accessibility to cars in South Africa, have contributed to increased use of this mode of transport. In the beginning of 2002, there were some 6.9 million registered vehicles in South Africa, more than 3.98 million of which were cars [15]. Although the private car provides spatial freedom to travel easily, and give status in terms of prosperity and adulthood, it does produce some significant social problems:

- Residential development occur at a lower density, and further away from the working place.
- Specialized urban activities, such as shopping and entertainment, are concentrated in a way that is supporting the use of private cars, as opposed to infrastructure supporting pedestrian access.
- A reduction in the number of public transport trips. This results in inefficient public transport use, decreased service levels, and eventually more people reverting to private car use.

The higher number of vehicles also results in a demand for more roads: infrastructure that, in itself, create pedestrian barriers, force inhabitants to relocate, change land use patterns, and reduce residential (social) quality. The effects of freeways are even more profound. De Boer [16] introduces transport sociology and states that traffic, and the desires of planners to provide additional infrastructure, should be tamed. The emphasis should be shifted from the *expansion* of infrastructure, to the *management* of the available transport infrastructure. When a transport system is in place, its utilization and physical condition should be improved. Additional infrastructure creates potential competition, resulting in suboptimal utilization of all infrastructure.

Environmental aspects of transport

Transportation is a major source of air pollution, emitting tons of carbon monoxide (CO), hydrocarbons (HO), and nitrogen oxides (NO_x) into the air [35]. The environmental effects of a transport system is not limited to a negative change in the chemical composition of the atmosphere, but also impacts the social environment. Noise pollution created by traffic results in sleep and speech interference, annoyance, and impairment of hearing after exposure over long periods of time [39].

2.2.2 A new approach to City Logistics Modelling

It is clear from the previous section that stakeholders do not participate as individual entities, but rather as complex networks. De Bruijn and tenHeuvelhof [17] identifies four important characteristics of stakeholder networks:

- *Interdependence.* Actors cannot achieve their goals without cooperation, as they are dependent on the resources of others actors, such as funding, information, and statutory powers.

- *Pluriformity.* Corporate actors do not behave as individuals, but as coalitions, since their constituents may have diverging and competing interests.

- *Self-containment.* Corporate actors are inclined to close themselves off from their environment, developing their own frame of reference and norms, making it harder to induce the cooperation.

- *Instability.* Positions and relations in policy networks are continually undergoing changes.

Villa [63] confirms autonomous decision making in networks. These characteristics impede the ability of stakeholders to make decisions that is optimal for the sustainability of the system as a whole. It does open a window of opportunity to address the dynamics of the relationships according to each stakeholder's perception. The Thomas-theorem states that *if man define situations to be real, they are real in their consequences* [60]. It implies that the operations research practitioner should not pursue a comprehensive model of a specific situation, but rather a set of models reflecting the diversity in actor perceptions. The soundness of the set of Thomas-models is not determined by the degree of correspondence to reality, but by the acuteness with which it mirrors the assumptions that actors make about their reality.
It may appear as if the Thomas-models are in conflict with figure 1.4 and the discussion in paragraph 1.4 that argues that the model assumptions should be challenged to ensure that a real representation of the problem is modelled. It is the opinion of the author that the Thomas-model approach emphasizes the importance of the operations research practitioners' skillful definition of the target audience (object) before engaging in the modelling task.

Van Duin *et al* [62] address the perspectives of individual actors and their strategic behavior through *perception based modelling.* There are various analysis and modelling techniques to capture decision-making processes. Taniguchi *et al* [57] discuss a computerized support tool, called *Dynamic Actor Network Analysis* (DANA), and proceed to introduce a new approach to city logistics modelling.

Dynamic actor network analysis

The first step towards modelling the stakeholder network is to capture the perceptions of multiple actors. The DANA tool is set up as an open database into which several perceptions of actors can be submitted. Perceptions are modelled in terms of assumptions. *Factual assumptions* represent how an actor perceives the current state of his environment. *Causal assumptions* represent changes that will occur in the perception of the actor, and uses *if-then* statements. *Teleological assumptions* represent the actor's view on his desirability, or purposefulness in the network. An important feature of the tool is a query generator. Queries include questions like: *which actors have conflicting goals on a specific factor*, or *which actors have different definitions for a factor?* These conflicts are brought to the modeler's attention during the design of a logistics model.

Performance measures

Performance measures, or performance indicators, are the normative values of the perception-based factors that represent how an actor perceives the current state of his environment. It is used to compare actual performance against a pre-defined norm. An actor creates the norm by attaching a quantified value, based on his perceptions, to a factor influencing his environment.

Logistics modelling

The last part of the approach is directed towards the calculation of the value part of the performance indicators. Dedicated models are developed to measure the impact of logistics concepts. The models are based on the important factors identified during the dynamic actor network analysis, as interpreted from the operations research practitioner's perspective. The challenge is to develop a model at such a level that it is both comprehensive and easy to understand for the actors involved, yet sufficiently detailed to be validated in practice.

2.3 Vehicle routing

There are significant features of truck operations in urban areas affecting mobility in cities:

- Pickup/delivery trucks visit a number of customers on a single trip. Optimized route schedules decreases operational expenses
- Several pickup/delivery trucks are usually operated as a group by a shipper/carrier. It is possible to reduce the number of vehicles used, as improved load factors increase fleet efficiency.
- Each customer specifies a time window to be visited by the pickup/delivery vehicle, complicating route schedules for carriers

Vehicle routing and scheduling procedures address the mobility in cities [57]. *Vehicle travel time/distance*, *total fleet cost*, and *customer satisfaction* are factors that are valued by both carriers and shippers. Vehicle routing and scheduling problems involve an optimization process of assigning customers to trucks and determining the visiting order of customers on truck routes.

According to Van Breedam [61] and Laporte [27], the general *Vehicle Routing Problem* (VRP) can be defined as the problem of finding a set of routes for a fleet of vehicles, which have to serve a number of customers (also referred to as stops or nodes) by offloading their goods. The vehicles depart from, and return to, a single depot. Vehicles must complete their individual routes within a maximum total route time. Although the basic VRP has been described mathematically in section 1.2.3, it will now formally be defined.

Let:

N	be the total number of customers
q_i	be the known demand for node i, where $i = \{1, 2, \ldots, N\}$
s_i	be the service time at node i, where $i = \{1, 2, \ldots, N\}$
d_{ij}	be the distance between nodes i and j, where $i, j = \{1, 2, \ldots, N\}$
c_{ij}	be the cost incurred on the arc between nodes i and j, where $i, j = \{1, 2, \ldots, N\}$
t_{ij}	be the travel time between nodes i and j, where $i, j = \{1, 2, \ldots, N\}$
K	be the total number of vehicles available
p	be the capacity of each vehicle in the homogeneous fleet

The principle decision variable, x_{ijk}, is defined as

$$x_{ijk} = \begin{cases} 1 & \text{if vehicle } k \text{ travels from node } i \text{ to node } j \text{, where} \\ & i, j = \{1, 2, \ldots, N | i \neq j\}, k = \{1, 2, \ldots, K\} \\ 0 & \text{otherwise} \end{cases}$$

$$\min z = \sum_{i=0}^{N} \sum_{j=0, j \neq i}^{N} \sum_{k=1}^{K} c_{ij} x_{ijk} \tag{2.1}$$

subject to

$$\sum_{j=1}^{N} x_{0jk} = \sum_{j=1}^{N} x_{j0k} = 1 \qquad \forall k = \{1, 2, \ldots, K\} \tag{2.2}$$

$$\sum_{j=1}^{N} \sum_{k=1}^{K} x_{0jk} \leq K \tag{2.3}$$

$$\sum_{i=1; i \neq j}^{N} \sum_{k=1}^{K} x_{ijk} = 1 \qquad \forall j \in \{1, 2, \ldots, N\} \tag{2.4}$$

$$\sum_{j=1; j \neq i}^{N} \sum_{k=1}^{K} x_{ijk} = 1 \qquad \forall i \in \{1, 2, \ldots, N\} \tag{2.5}$$

$$\sum_{i=1}^{N} q_i \sum_{j=0; j \neq i}^{N} x_{ijk} \leq p \qquad \forall k \in \{1, 2, \ldots, K\} \tag{2.6}$$

$$x_{ijk} \in \{0, 1\} \tag{2.7}$$

The objective of the problem is to minimize the total travel cost incurred during the process of servicing the N customers. The element c_{ij} in (2.1) can be replaced with either t_{ij} to minimize travel time, or with d_{ij} to minimize travel distance. Constraint (2.2) ensures that all routes start and end at the depot, node 0, while (2.3) ensures that the maximum number of vehicles/routes are not exceeded. Constraints (2.4) and (2.5) limit the number of visits to each node to 1, while (2.6) ensures that the cumulative demand on any route does not exceed the vehicle capacity.

2.3.1 The vehicle routing problem and its variants

The basic VRP assumes that nodes can be visited anytime during the route, that all vehicles are homogeneous in terms of cost and capacity, and that each vehicle can only service one route during a scheduling period. There exist numerous extensions to the VRP. These arise when additional side constraints are added to adapt the basic VRP to many real-life business scenarios. The three variants considered in this dissertation are described in the following paragraphs.

Time windows

A *time window* can be described as a window of opportunity for deliveries. It is an extension of the VRP that has been researched extensively. Examples include the work of Ibaraki *et al.* [25], Taillard [50], Taillard *et al.* [51], and Tan *et al.* [54]. A time window is the period of time during which deliveries can be made to a specific customer i, and has three main characteristics:

- the earliest allowed arrival time, e_i (also referred to as the *opening time*),
- the latest allowed arrival time, l_i (also referred to as the *closing time*),
- whether the time window is considered *soft* or *hard.*

Consider the example, illustrated in figure 2.4, where customer i requests delivery between 07:30 and 17:00.

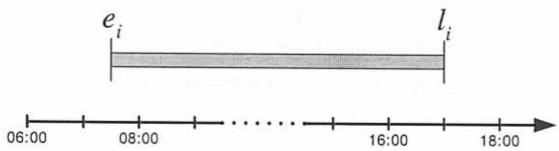

Figure 2.4: Double sided hard time window

To distinguish between the actual and the specified times of arrival, the variable a_i denotes the actual time of arrival at node i. Should the actual arrival time at node i, a_i, be earlier than the earliest allowed arrival at the node, e_i, then the vehicle will incur a waiting time, w_i, which can be calculated as $w_i = \max\{0, e_i - a_i\}$. The introduction of time windows to the basic VRP sees the introduction of three new constraints.

$$a_0 = w_0 = s_0 = 0 \tag{2.8}$$

$$\sum_{k=1}^{K} \sum_{i=0;i \neq j}^{N} x_{ijk}(a_i + w_i + s_i + t_{ij}) \leq a_j \qquad \forall j = \{1, 2, \dots, N\} \tag{2.9}$$

$$e_i \leq (a_i + w_i) \leq l_i \qquad \forall i = \{1, 2, \dots, N\} \tag{2.10}$$

Constraint (2.8) assumes that vehicles are ready and loaded by the time the depot opens, which is indicated as time 0 (zero). Constraint (2.9) calculates the actual arrival time, while (2.10) ensures that each customer i is serviced within its time window.

When both an earliest and latest allowed arrival is stipulated, the time window is referred to as *double sided.* If no arrivals are allowed outside of the given parameters, the time window is said to be *hard,* as is the case in figure 2.4. When delivery is allowed outside the specified time window, the time window is said to be *soft,* and the lateness is penalized at a cost of α_i. Customer i may specify a maximum lateness, L_i^{max}. The example illustrated in figure 2.5 sees customer i specifying a time window between 07:30 and 15:30. The customer will, however, allow late deliveries until 17:00. A hard time window is therefor a special type of soft time window where $L_i^{max} = 0$.

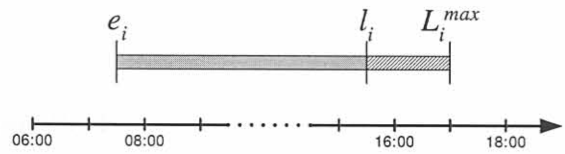

Figure 2.5: Soft time window

Should a vehicle arrive after the latest allowed arrival time, l_i, but prior to the maximum lateness, L_i^{max}, the lateness at node i, L_i, can be calculated as $L_i = \max\{0, a_i - l_i\} | a_i \leq L_i^{max}$. The lateness is penalized by introducing a penalty term to the VRP objective function (2.1).

$$\min \sum_{i=0}^{N} \sum_{j=0, j\neq i}^{N} \sum_{k=1}^{K} c_{ij} x_{ijk} + \sum_{i=1}^{N} \alpha_i \times \max\{0, L_i\} \tag{2.11}$$

The time window for the depot, node 0, can be specified. The case illustrated in figure 2.6 sees the depot specifying operating hours (time window) from 06:00 to 18:00, while the first customer on the route, customer 1, specifies a time window between 07:00 and 09:00, and the last customer, customer n, requests delivery between 15:00 and 17:00.

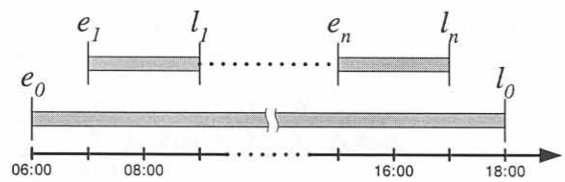

Figure 2.6: Time window for the depot, node 0

Should a customer specify multiple time windows, an indexing symbol, a, is introduced as superscript to the earliest and latest allowed arrival times, respectively, where $a \in \{1, 2, \ldots, A\}$ in which A indicates the maximum number of time windows allowed for each customer. Consider the example where customer n requests delivery either between 06:30 and 09:00, or between 16:00 and 17:30. The case is illustrated in figure 2.7. This example

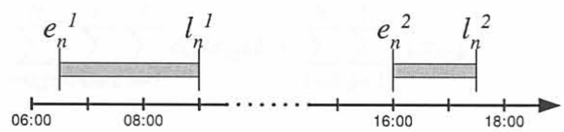

Figure 2.7: Multiple time windows

is typical of residents requesting home shopping deliveries outside business hours.

Heterogeneous fleet

Gendreau *et al.* [20] propose a solution methodology for cases where the fleet is heterogeneous, that is, where the fleet is composed of vehicles with different capacities and costs. Their objective is to determine what the optimal fleet composition should be, and is referred to as either a *Heterogeneous Fleet Vehicle Routing Problem* (HVRP), or a *Fleet Size and Mix Vehicle Routing Problem* (FSMVRP). Taillard [50] formulates the *Vehicle Routing Problem with a Heterogeneous fleet of vehicles* (VRPHE) where the number of vehicles of type t in the fleet is limited; the objective being to optimize the utilization of the given fleet. Salhi and Rand [45] incorporates vehicle routing into the vehicle composition problem, and refer to it as the *Vehicle Fleet Mix problem* (VFM).

The implication of a heterogeneous fleet on the standard VRP is that T type of vehicles are introduced, with $t \in \{1, 2, \ldots, T\}$. The vehicle capacity parameter p is changed. The new parameter, p_t, represents the capacity of vehicles of type t, resulting in each vehicle k having a unique capacity, p_k. The use of one vehicle of type t implies a fixed cost f_t. A unique fixed cost, f_k, is introduced to each vehicle k, based on its vehicle type. The objective function changes to

$$\min \sum_{i=0}^{N} \sum_{j=0, j \neq i}^{N} \sum_{k=1}^{K} c_{ij} x_{ijk} + \sum_{k=1}^{K} \sum_{j=1}^{N} f_k x_{0jk} \tag{2.12}$$

while (2.6) changes to indicate the new capacity parameter

$$\sum_{i=1}^{N} q_i \sum_{j=0; j \neq i}^{N} x_{ijk} \leq p_k \quad \forall k = \{1, 2, \ldots, K\} \tag{2.13}$$

Taillard [50] introduces a variable c_{ijt} to represent the cost of travelling between nodes i and j, using a vehicle of type t. It is possible to introduce the variable portion of the vehicle cost into the objective function (2.12). The introduction will lead to

$$\min \sum_{i=0}^{N} \sum_{j=0, j \neq i}^{N} \sum_{k=1}^{K} \sum_{t=1}^{T} c_{ijt} x_{ijk} \xi + \sum_{k=1}^{K} \sum_{j=1}^{N} f_k x_{0jk} \tag{2.14}$$

where

$$\xi = \begin{cases} 1 & \text{if vehicle } k \text{ is of type } t, \text{ where } k = \{1, 2, \ldots, K\}, \\ & t = \{1, 2, \ldots, T\} \\ 0 & \text{otherwise} \end{cases}$$

Double scheduling

It is often not viable to assume that each vehicle will only complete a single route. *Double scheduling* is concerned with the case where a vehicle could complete deliveries on a scheduled route, return to the depot where its capacity is renewed, after which a second, or consecutive trip is executed with the renewed capacity. Taillard *et al.* [53] refer to this type of problem as the *Vehicle Routing Problem with Multiple use of vehicles* (VRPM). Butt and Ryan [11] consider the *Multiple Tour Maximum Collection Problem* (MTMCP) and assumes that the routes are constrained in such a way that all of the customers cannot be visted. Their approach aims to maximize the number of customers serviced. Brandão and Mercer [9] introduce the *Multi-Trip Vehicle Routing Problem* (MTVRP), and address the combination of multiple trips with time windows.

This dissertation considers a vehicle that starts and ends its tour at the depot. A *tour* consists of one or more *routes*, each starting and ending at the depot. The same vehicle can only be used for two or more routes if the routes do not overlap. As opposed to (2.8), multiple routes require a service time to be specified for the depot. Consider the example illustrated in figure 2.8. The depot has a time window from 06:00 to 18:00. A vehicle fills its

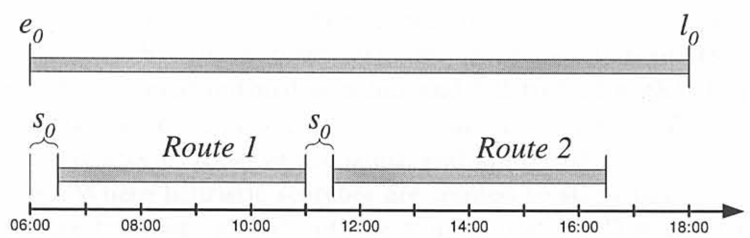

Figure 2.8: Double scheduling

capacity at the depot for a time period of $s_0 = 0.5$ hours. It leaves the depot at 06:30, services the first route, and returns to the depot at 11:00, where its capacity is renewed. A second route, of five hours, is serviced before the vehicle returns to the depot.

The mathematical implication of double scheduling on the basic VRP could not be established from literature. Taillard *et al.* [53] confirm that this type of problem has received very little attention in literature. This dissertation proposes a way to deal with multiple routes. The proposed solution involves a time verification process. If a vehicle arrives back at the depot at time a_m, and the service time is specified as s_0, then the vehicle is considered for an additional route on its current tour if, after the capacity has been renewed, the depot's time window is still open. The case is represented in (2.15).

$$a_m + s_0 \leq l_0 \tag{2.15}$$

2.3.2 Computational complexity of the VRP

Problems that are considered hard to solve are those problems for which there are not polynomial solution algorithms, and are referred to as *nondeterministic* (NP) *class problems*. The problems that require an inordinate amount of computer processing time in the *NP* class are identified as *NP-hard* problems. Lenstra and Rinnooy, and Laporte, have classified vehicle routing and scheduling problems as *NP-hard* problems [27, 29].

When confronted with difficult problems to be solved (*NP-hard*), operations research (OR) practitioners have used approximation techniques to arrive at a good solution. The approximate rule-of-thumb techniques where optimality cannot be assured are classified as *heuristic techniques* in OR practice [65].

2.3.3 Solving the vehicle routing problem

Heuristics typically uses a greedy approach to obtain a good initial solution in efficient time and then incrementally improve the solution by neighborhood exchanges or local searches. As a result, heuristics tend to get trapped in a local optimal solution and fail to find a global optimum. Heuristics have evolved into *global optimization heuristics*. These are general master strategies to solve problems, and are based on intelligent search techniques. Where heuristic searches are limited to steps that will improve the objective function, global optimization heuristics allow steps that will temporarily decrease the objective function value, in an attempt to escape the local optimum and look for the global optimum, or at least a better local optimum. These global optimization heuristics are often called *metaheuristics* because the procedure used to generate new solution out of the current one, is embedded in a heuristic which determines the search strategy. The main drawback of metaheuristics is that they do not have definitive stopping criteria; the longer the computation time, the higher the probability of finding the global optimum [61].

Initial solution algorithms

Solomon divides VRP tour-building algorithms into either sequential or parallel methods [46]. Sequential procedures construct one route at a time until all customers are scheduled. Parallel procedures are characterized by the simultaneous construction of routes, while the number of parallel routes can either be limited to a predetermined number, or formed freely. Solomon concludes that, from the five initial solution heuristics evaluated, the *sequential insertion heuristic* (SIH) proved to be very successful, both in terms of the quality of the solution, as well as the computational time required to find the solution.

Initialization criteria refers to the process of finding the first customer to insert into a route. The most commonly used initialization criteria is the *farthest unrouted* customer, and the customer with the *earliest deadline*, or the earliest *latest allowed arrival.* The first customer inserted on a route is referred to as the *seed customer.* Once the seed customer has been identified and inserted, the SIH algorithm considers, for the unrouted nodes, the insertion place that minimizes a weighted average of the additional distance and time needed to include a customer in the current partially constructed route. This second step is referred to as the *insertion criteria.* The third step, the *selection criteria,* tries to maximize the benefit derived from inserting a customer in the current partial route rather than on a new direct route. Note that the terms *nodes* and *customers* are used interchangeably. The insertion and selection criteria can be simplified using the example illustrated in figure 2.9. The partially constructed route in the example consists

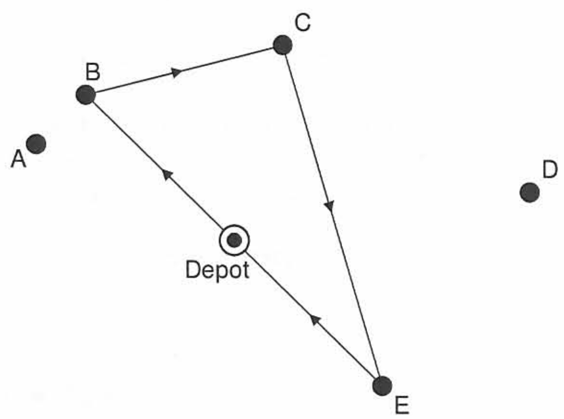

Figure 2.9: Sequential insertion of customers

of the depot and three routed nodes, namely B, C, and E. The route can be expressed as $Depot - B - C - E - Depot$. Nodes A and D are currently unrouted.

The *insertion criteria,* $c_1(i, u, j)$, calculates the best position and associated cost, between two adjacent nodes i and j on the partial route, to insert a customer u, and is calculated for each of the unrouted nodes. Consider node A in the example, there are currently four edges where the node can be inserted, namely $Depot - B$, $B - C$, $C - E$, or $E - Depot$, as illustrated in

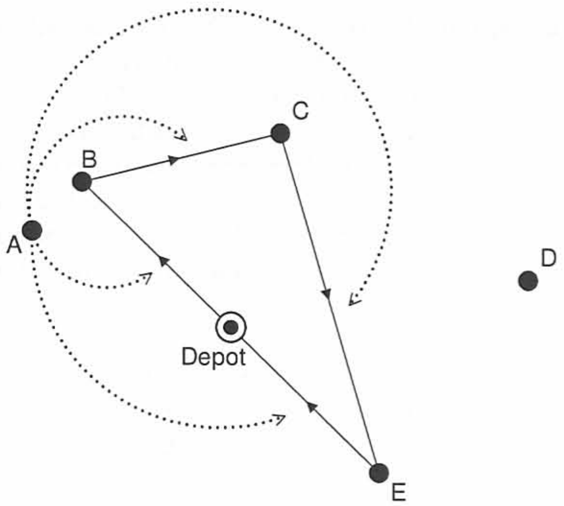

Figure 2.10: Selection criteria

figure 2.10. Dullaert *et al.* [19] extends Solomon's heuristic and determines $c_1(i, A, j)$ for the unrouted node A as

$$c_1(i, A, j) = \min_p[c_1(i_{p-1}, A, i_p)], p = \{1, 2, \ldots, m\} \tag{2.16}$$

in which m represents the routed nodes in the partially constructed route. If the expressions are generalized for all unrouted nodes u, the insertion criteria is calculated as

$$c_1(i, u, j) = \alpha_1 c_{11}(i, u, j) + \alpha_2 c_{12}(i, u, j) + \alpha_3 c_{13}(i, u, j) \tag{2.17}$$

with

$$c_{11}(i, u, j) = d_{iu} + d_{uj} - \mu d_{ij}, \mu \geq 0 \tag{2.18}$$
$$c_{12}(i, u, j) = a_j^{new} - a_j \tag{2.19}$$
$$c_{13}(i, u, j) = ACS, AOOS, or AROS \tag{2.20}$$

With the extension to Solomon's heuristic, the weighting factors α_i need not add up to 1. $c_{11}(i, u, j)$ denotes the additional distance, and $c_{12}(i, u, j)$ the additional time needed to serve customer u after customer i, but before customer j. The new actual arrival time at node j is denoted by b_j^{new} in (2.19). The vehicle savings criteria, denoted by $c_{13}(i, u, j)$, considers any one of three parallel approaches to vehicle cost, where Dullaert *et al.* [19] adapts the savings concepts first introduced by Golden *et al.* [22]. To elaborate on the concepts, let

$F(z)$	be the fixed cost of the smallest vehicle that can service a cumulative route demand of z
$F'(z)$	be the fixed cost of the largest vehicle whose capacity is less than or equal to z
$P(z)$	be the capacity of the smallest vehicle that can service a demand of z
Q	be the load of the vehicle currently servicing the route
$\overline{Q}$	be the maximum capacity of the vehicle currently servicing the route
Q^{new}	be the new load of the vehicle after the customer has been inserted into the route
$\overline{Q}^{new}$	be the (new) capacity of the vehicle after the customer has been inserted into the route

The *Adapted Combined Savings* (ACS) is defined as the difference between the fixed costs of the vehicles capable of transporting the load of the route after, and before, inserting customer u, and is calculated as

$$ACS = F(Q^{new}) - F(Q) \tag{2.21}$$

The *Adapted Optimistic Opportunity Savings* (AOOS) extends the ACS by subtracting the fixed cost of the vehicle that can service the unused capacity, and is calculated as

$$AOOS = [F(Q^{new}) - F(Q)] - F(\overline{Q}^{new} - Q^{new}) \tag{2.22}$$

The *Adapted Realistic Opportunity Savings* (AROS) takes the fixed cost of the largest vehicle smaller than or equal to the unused capacity, $F'(\overline{Q}^{new} - Q^{new})$, into account as an opportunity saving. It only does so if a larger vehicle is required to service the current route after a new customer has been inserted. AROS is calculated as

$$AROS = [F(Q^{new}) - F(Q)] - \delta(\omega)F'(\overline{Q}^{new} - Q^{new}) \tag{2.23}$$

where

$$\delta(\omega) = \begin{cases} 1 & \text{if } Q + q_u > \overline{Q} \\ 0 & \text{otherwise} \end{cases}$$

Any *one* of these savings criteria can be used as all three outperformed previous best published results for the initial solution [19]. Once the best position for each unrouted node has been determined, as illustrated in figure 2.11, the customer that is best according to the *selection criteria*, is selected. The procedure can be expressed mathematically as

$$c_2(i, u^\star, j) = \min_u [c_2(i, u, j)], u \text{ unrouted and feasible} \tag{2.24}$$

$$c_2(i, u, j) = \lambda(d_{ou} + t_{ou}) + s_u + F(q_u) - c_1(i, u, j), \lambda \geq 0 \tag{2.25}$$

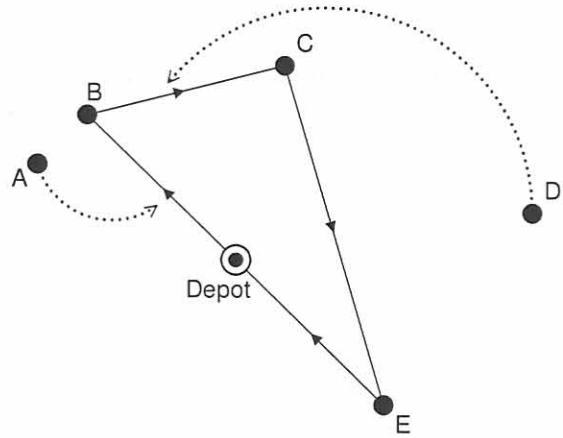

Figure 2.11: Best insertion position determined for each unrouted node

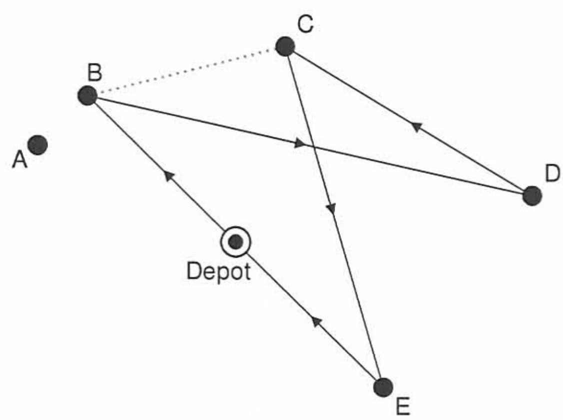

Figure 2.12: New route after inserting best customer

The best customer, $u^\star$, is then inserted into the partially created route between its specific nodes i and j. From figure 2.11, consider node D to be the best node. After inserting D into the current route, node A remains the only unrouted node, and the new route is illustrated in figure 2.12, and can be expressed as $Depot - B - D - C - E - Depot$. The insertion process is repeated until no remaining unrouted nodes have a feasible insertion place. A new route is then initialized and identified as the *current* route.

Van Breedam [61] introduces an initial solution parameter in his evaluation of improvement algorithms, and finds that, in most cases, a good initial solution gives significantly better final results.

Solution improvement algorithms

Initial solutions serve as input to improvement heuristics, and although these improvement heuristics are not the main focus of this dissertation, the three main metaheuristics is described briefly.

The evaluation of any heuristic is subject to the comparison of a number of criteria that relate to different aspects of the algorithm's performance. Such criteria include computational time, quality of solution, ease of implementation, and flexibility. Bräysy and Gendreau [10] state that flexibility is important criteria for algorithms that are designated to to be used in real-world problems, as an algorithm should be able to handle changes to the constraints and objective function. There is generally a trade-off between computational time and solution quality. This characteristic is a key feature of metaheuristics.

Tan *et al.* [54] compare the three popular meta heuristic methods with one another and concludes that there is no single heuristic that is generic enough to solve problems for all situations. Instead, they are inevitably problem specific. Each of the three popular metaheuristics has its advantages and disadvantages.

The *Simulated Annealing* (SA) methodology is similar to the annealing process of solids. When a metal is heated to high temperatures it is structurally weak and unstable. If the metal is allowed to cool slowly, it orders itself into a stable, structurally strong configuration. This process is called annealing. Kirkpatric *et al.* [26] first proposed the use of SA for optimization, while Osman [41] applied it to vehicle routing and scheduling problems. The states of the solids correspond to the feasible solution, and the temperature at each state corresponds to the improvement in objective function, with the minimum temperature being the optimal solution. SA involves a process in which the temperature is gradually reduced during the simulation. Often, the system is first heated and then cooled. The system is given the opportunity to surmount energy barriers in a search for conformations with temperatures lower than the local-minimum temperature found by energy minimization. At each step of the simulation algorithm, a new state of the system is constructed from the current state by giving a random displacement to a randomly selected particle. If the energy associated with this new state was lower than the energy of the current state, the displacement was accepted, that is, the new state becomes the current state. This basic step, called a metropolis step, can be repeated indefinitely. The whole procedure is called a metropolis loop. Some of the choices that need to be made with a SA strategy are

- the expression used to calculate the initial temperature,

- the cooling function used for the reduction of temperature,
- the conditions for thermal equilibrium – where the temperature can be lowered, and
- the stopping criteria – usually a number of consecutive degrading values.

SA produces good solutions much faster than other metaheuristics, in terms of computational time, although the solution quality of the final solution is inferior.

Reeves [44] describes the *Genetic Algorithm* (GA) as the intelligent exploitation of a random search that was first presented by Holland [23]. The name originates from the analogy between the representation of a complex structure by means of a vector of components, and the idea of the genetic structure of a chromosome. GA is described as follows:

> *A population of solutions are maintained and a reproductive process allows parent solutions to be selected from the population. Offspring solutions are produced which exhibit some of the characteristics of each parent. The fitness of each solution can be related to the objective function value, in this case the total distance travelled [4].*

Analogous to biological processes, offspring with good fitness levels are more likely to survive and reproduce. Selection of the fittest ensures that fitness levels throughout the population will improve with evolution. The result is a population of chromosomes, in this case vehicle routes, with high performance characteristics. In most applications the component vector, or chromosome, is represented as a string of bits (0 and 1). Tan *et al.* [54] argue the case of replacing the binary digits with integer digits. A chromosome can now easier represent the order in which customers are visited. Genetic operators are used to improve the quality of the population. The most common genetic operators used to manipulate theses chromosomes are crossover and mutation. *Crossover* is the exchange of sections of the parents' chromosomes, while *mutation* is the random modification of the chromosome.

Researchers have explored the adaptations of GA to form *hybrid-GA* [54, 59]. The highest potential for hybrid-GA seems to be when the basic principles of GA are combined with adaptive memory features introduced by Taillard *et al.* [52]. This flexibility of the hybrid-GA makes it a very popular heuristic and holds promising prospects for the application of GA. Thangiah [58] recognizes that GA does not perform well for problems in which customers are geographically clustered together.

The *Tabu Search* metaheuristic (TS), a memory based search strategy, deals with the problem of being trapped at a local optimum by temporarily forbidding moves that would return to a solution recently visited (cycling). The result is that the tabu search heuristic prevents short-term cycling, although solutions can be repeated over a longer period of time. This heuristic was proposed in its present form by Glover [21] and has been applied to many optimization problems besides vehicle routing. A *tabu list* is used to record these forbidden moves, which means that each iteration choose a non-tabu move. After each step, a collection of moves that includes any immediate return to the previous point is added to the tabu list [43]. These recently added moves are not allowed for a fixed number of iterations, but are eventually removed from the tabu list.

As with other metaheuristics, any particular iteration can either improve or degrade the objective function value. It is therefore important to continuously update the best feasible solution found so far, referred to as the *incumbent solution*. When the tabu search is complete, usually after a finite number of iterations have been completed, the incumbent solution is accepted as the final solution: an approximation of the optimum solution. It is important to design the criteria of the TS carefully. The decision of which moves to add or remove from the tabu list is imperative as it has a direct effect on both the accuracy and computational time of the search. If too few moves are disallowed, it may lead to cycling; too many disallowed moves restrict the search from finding superior solutions quickly.

The tabu list contains record of three elements: the list position, the original route and the string of stops moved. The list is implemented as a queue that operated on a first-in, first-out principle. The memory of the tabu list can be *recency* or *frequency* based. The recency based list, called short-term memory, contains the last x number of moves the algorithm has encountered and sets them as tabu, assuming that the tabu list size is x. The frequency based list, called long-term memory, complements the short-term memory by providing the additional information of how many times each tabu move have been attempted. Tan *et al.* [54] propose the use of a multi-functional list structure that serves both purposes of a recency and frequency-based list, and state that frequency-based memory provides better incentive as to the choice of the next move. Van Breedam [61] concludes that the use of long-term memory gives significantly worse solutions for the majority of problems. The impact of these contradictory opinions is that, should TS be used, the easier implementable option of short-term memory is preferred. A way of still retaining long-term memory is to increase the length of the tabu list.

Van Breedam [61] notes that unlike the SA heuristic, the performance of the TS is highly dependent on the quality of the initial solution. The use of TS requires careful consideration of the mentioned characteristics of TS, as it should be chosen to best represent the problem environment of the project.

TS solutions are generally closest to optimal, but the computational time is about two to three times that of GA, and almost twenty times that of SA. The current computational power of modern computers makes the slower performance of TS less of a problem. Tan *et al.* [54] report that TS was able to solve 56 problem instances, each containing 100 customers, in an average of 1,500 seconds. Carriers furthermore require an operational scheduling system on a daily basis, with real-time scheduling as a future functionality.

2.4 Conclusion

The term *City Logistics* refer to the process of optimizing urban freight movement in a multi-actor environment. Changes occur in the environment, and stakeholders have conflicting perceptions with regards to how the changes will impact them, and their co-stakeholders. It is not contested that fleet optimization, in the form of vehicle routing and scheduling, will have advantages for all stakeholders. Literature indicates that, to address the specific needs and perceptions of logistic stakeholders, more realistic models are required to manage and predict factors influencing urban networks.
Realistic models require the introduction of additional constraints to the solution space and results in problems being harder to solve. The quality of the final solution is impacted by the quality of the initial solution generated through heuristic methods.

Chapter 3 introduces a new approach, *time window compatibility*, to improve the quality of the initial solution. The aim is to have a mechanism that will assist in identifying seed customers in an innovative way during the route building process, and also ease the computational burden.

Chapter 3

Time window compatibility

The purpose of this chapter is to introduce a new concept, *Time Window Compatibility* (TWC). A matrix, referred to as the *Time Window Compatibility Matrix* (TWCM), is used as the mechanism to calculate the compatibility between all nodes in the network. A brief motivation for the concept is given. Various scenarios of the concept exist, and are illustrated individually. An attempt is made to define the concept in a generalized form.

An example is given to illustrate the use of the concept. Other than reducing the computational burden of existing heuristics, the example also illustrates the opportunity to improve the quality of the initial solution. A modified copy of the concepts discussed in this chapter have been communicated the the *European Journal of Operational Research.*

3.1 Motivation for a new approach

A shortcoming of the Sequential Insertion Heuristic (SIH) of Solomon [46] is that it considers all unrouted nodes when calculating the insertion and selection criteria for each iteration. The fact that *all* unrouted nodes are considered makes it computationally expensive. The VRP variant considered in this dissertation has multiple additional constraints. The occurrence of obvious infeasible nodes in a partially constructed route therefore becomes significant. The introduction of the time window compatibility concept assists in identifying, and eliminating, the obvious infeasible nodes. This results in a more effective and robust route construction heuristic.

3.2 Time window compatibility defined

The purpose of TWC is to determine the time overlap of all edges, or node combinations, (i, j), where $i, j \in \{0, 1, 2, \dots, N\}$, and N the total number of nodes in the network. During the route construction phase, time window compatibility can be checked, and obvious infeasible nodes can be eliminated from the set of considered nodes. The TWCM is a non-symmetrical matrix as the sequence of two consecutive nodes, i and j, is critical.

Let:

N	be the total number of nodes
e_i	be the earliest allowed arrival time at customer i, where $i = \{0, 1, \ldots, N\}$
l_i	be the latest allowed arrival time at customer i, where $i = \{0, 1, \ldots, N\}$
s_i	be the service time at node i, where $i = \{0, 1, \ldots, N\}$
t_{ij}	be the travel time from node i to node j, where $i, j = \{0, 1, \ldots, N\}$
$a_j^{e_i}$	be the actual arrival time at node j, given that node j is visited directly after node i, and that the actual arrival time at node i was e_i, where $i, j = \{0, 1, \ldots, N\}$
$a_j^{l_i}$	be the actual arrival time at node j, given that node j is visited directly after node i, and that the actual arrival time at node i was l_i, where $i, j = \{0, 1, \ldots, N\}$
TWC_{ij}	be the time window compatibility when node i is directly followed by node j

TWC_{ij} indicates the entry in row i, column j of the TWCM. Consider the following five scenarios illustrating the calculation of the time window compatibility. Each scenario assume customer j to be serviced directly after customer i, a service time of one hour, and a travel time of two hours from node i to node j.

Scenario 1: if $a_j^{e_i} > e_j$ and $a_j^{l_i} < l_j$, illustrated in figure 3.1. Customer i

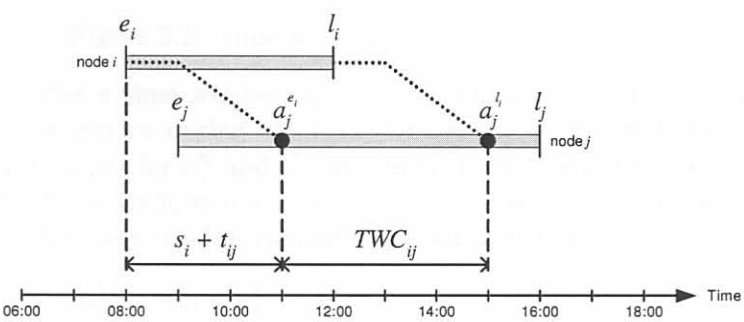

Figure 3.1: Time window compatibility scenario 1

specifies a time window $[e_i, l_i]$ = [08:00,12:00], while customer j requires service during the time window $[e_j, l_j]$ = [09:00,16:00]. If service at customer i starts at the earliest allowed time, e_i, then the actual arrival time at customer j would be calculated as

$$a_j^{e_i} = e_i + s_i + t_{ij} \tag{3.1}$$

In this scenario $a_j^{e_i}$ = 11:00. Similarly, $a_j^{l_i}$ would be the actual arrival time at customer j, given that the actual arrival time at customer i was l_i, and is calculated as

$$a_j^{l_i} = l_i + s_i + t_{ij} \tag{3.2}$$

The difference between $a_j^{e_i}$ and $a_j^{l_i}$ indicates the time window overlap between the two nodes. The time window compatibility is calculated as

$$TWC_{ij} = a_j^{l_i} - a_j^{e_i} \tag{3.3}$$

For this example, the time window compatibility is four hours (04:00).

Scenario 2: if $a_j^{e_i} > e_j$ and $a_j^{l_i} > l_j$, illustrated in figure 3.2. Customer i

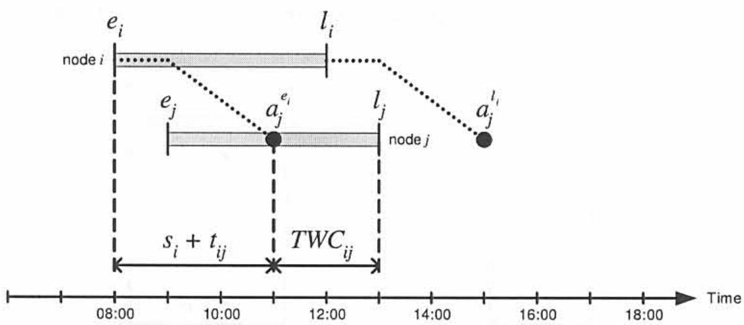

Figure 3.2: Time window compatibility scenario 2

specifies a time window $[e_i, l_i]$ = [08:00,12:00], while customer j requires service during the time window $[e_j, l_j]$ = [09:00,13:00]. The calculations for $a_j^{e_i}$ and $a_j^{l_i}$ are similar to (3.1) and (3.2), respectively. The time windows of customer i and customer j only partly overlap, and the time window compatibility is calculated as

$$TWC_{ij} = l_j - a_j^{e_i} \tag{3.4}$$

For this example, the time window compatibility is two hours (02:00).

Scenario 3: if $a_j^{e_i} < e_j$ and $a_j^{l_i} < l_j$, illustrated in figure 3.3. Customer i specifies a time window $[e_i, l_i]$ = [08:00,12:00], while customer j requires service during the time window $[e_j, l_j]$ = [12:00,16:00]. The

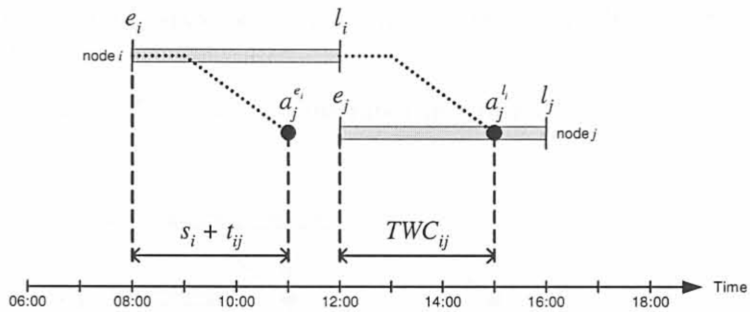

Figure 3.3: Time window compatibility scenario 3

calculations for $a_j^{e_i}$ and $a_j^{l_i}$ are similar to (3.1) and (3.2), respectively. The time windows of customer i and customer j only partly overlap, and the time window compatibility is calculated as

$$TWC_{ij} = a_j^{l_i} - e_j \tag{3.5}$$

For this example, the time window compatibility is three hours (03:00).

Scenario 4: if $a_j^{e_i}, a_j^{l_i} < e_j$, illustrated in figure 3.4. Customer i specifies

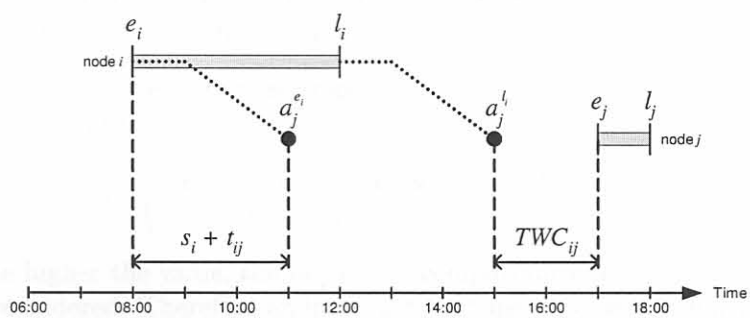

Figure 3.4: Time window compatibility scenario 4

a time window $[e_i, l_i]$ = [08:00,12:00], while customer j requires service during the time window $[e_j, l_j]$ = [17:00,18:00]. The calculations for $a_j^{e_i}$ and $a_j^{l_i}$ are similar to (3.1) and (3.2), respectively. The time windows of customer i and customer j do not overlap. Even if customer i is serviced as late as possible, l_i, a waiting time is incurred at customer j. The time window compatibility is calculated as

$$TWC_{ij} = a_j^{l_i} - e_j \tag{3.6}$$

For this example, the time window compatibility is negative two hours (-02:00). The significance of the negative time is that it is possible, in this case, to service customer j after customer i, although the waiting time is penalized.

Scenario 5: if $a_j^{e_i}, a_j^{l_i} > l_j$, illustrated in figure 3.5. Customer i specifies

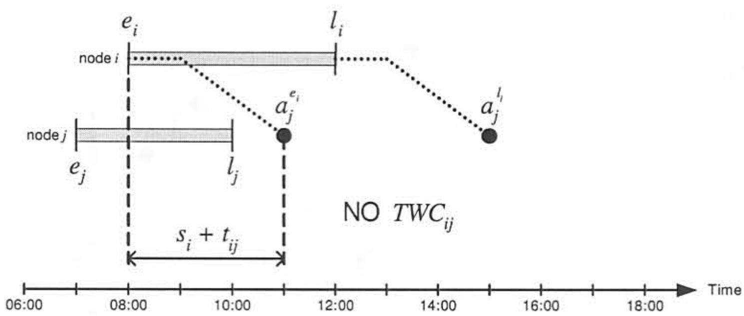

Figure 3.5: Time window compatibility scenario 5

a time window $[e_i, l_i] =$ [08:00,12:00], while customer j requires service during the time window $[e_j, l_j] =$ [07:00,11:00]. The calculations for $a_j^{e_i}$ and $a_j^{l_i}$ are similar to (3.1) and (3.2), respectively. Although the time windows of customer i and customer j partly overlap, it is impossible to service customer j, even if customer i is serviced as early as possible, e_i. Therefor, no time window compatibility exist.

A generalized equation is proposed that will address all five scenarios illustrated, and is given as

$$TWC_{ij} = \begin{cases} \min\{a_j^{l_i}, l_j\} - \max\{a_j^{e_i}, e_j\}, \text{ if } l_j - a_j^{e_i} > 0 \\ -\infty \text{ otherwise} \end{cases} \tag{3.7}$$

The higher the value, the better the compatibility of the two time windows considered. Therefore an incompatible time window is defined to have a compatibility of negative infinity.

Example. Consider the following example with five nodes geographical distributed around a depot in figure 3.6. In the example, node c has indicated two possible time windows. As discussed in section 2.3.1, the customer is artificially split and treated as two separate nodes, c^1 and c^2, respectively. The time windows for each customer, including the depot, as well as the service time at each node, are given in table 3.1. The distance matrix, $\overline{D}$, is calculated using the rectangular distance

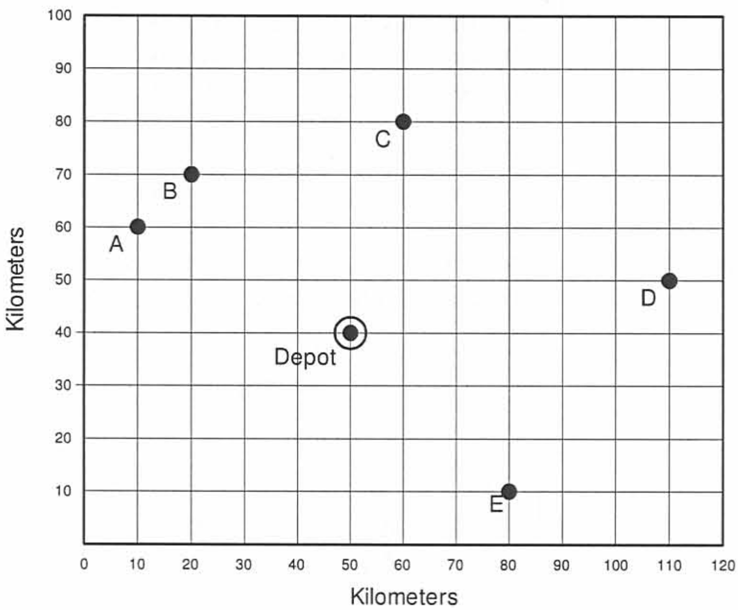

Figure 3.6: Geographical distribution of nodes around a depot

Table 3.1: Time windows and service times

Node (i)	**Time window** ($e_i; l_i$)	**Service time (in hours)** s_i
Depot	07:00 – 18:00	0.00
a	08:00 – 12:00	0.50
b	11:00 – 13:00	0.25
c^1	08:00 – 09:00	0.25
c^2	15:00 – 17:00	0.25
d	08:00 – 12:00	0.50
e	10:00 – 15:00	0.25

between nodes, as taken from figure 3.6.

$$\overline{\mathrm{D}} = \begin{bmatrix} 0 & 60 & 60 & 50 & 50 & 70 & 60 \\ 60 & 0 & 20 & 70 & 70 & 110 & 120 \\ 60 & 20 & 0 & 50 & 50 & 110 & 120 \\ 50 & 70 & 50 & 0 & 0 & 80 & 90 \\ 50 & 70 & 50 & 0 & 0 & 80 & 90 \\ 70 & 110 & 110 & 80 & 80 & 0 & 70 \\ 60 & 120 & 120 & 90 & 90 & 70 & 0 \end{bmatrix}$$

If the average speed is known, the time matrix, $\overline{T}$, can be calculated. In this example, $\overline{T}$ is given. Values are given in hours.

$$\overline{\mathrm{T}} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0.5 & 1 & 1 & 2 & 2 \\ 1 & 0.5 & 0 & 1 & 1 & 2 & 2 \\ 1 & 1 & 1 & 0 & 0 & 1.5 & 1.5 \\ 1 & 1 & 1 & 0 & 0 & 1.5 & 1.5 \\ 1 & 2 & 2 & 1.5 & 1.5 & 0 & 1 \\ 1 & 2 & 2 & 1.5 & 1.5 & 1 & 0 \end{bmatrix}$$

With the information at hand, the time window compatibility matrix can be calculated. For the given example,

$$\overline{\mathrm{TWCM}} = \begin{bmatrix} 11 & 4 & 2 & 1 & 2 & 4 & 5 \\ 4 & 3.5 & 2 & -\infty & -1.5 & 1.5 & 4 \\ 2 & 0.25 & 1.75 & -\infty & -0.75 & -\infty & 1.75 \\ 1 & 1 & -0.75 & 0.75 & -5.75 & 1 & 0.75 \\ 1.75 & -\infty & -\infty & -\infty & 1.75 & -\infty & -\infty \\ 4 & 1.5 & 2 & -\infty & -1 & 3.5 & 3.5 \\ 5 & -\infty & 0.75 & -\infty & 1.75 & 0.75 & 4.75 \end{bmatrix}$$

3.3 Using the time window compatibility matrix

The time window compatibility matrix, TWCM, is calculated before the route building heuristic is evoked.

3.3.1 Reducing computational complexity

Solomon [46] calculates the insertion and selection criteria by means of (2.17) and (2.25), respectively. In each iteration, these criteria are calculated for each edge on the partially constructed route, irrespective of the compatibility of the time window of the node considered for insertion with the time windows of the two nodes forming the edge.

This chapter presents an improved case. Consider the example where node u is considered for insertion between nodes i and j. As the TWCM is already calculated, it is possible to check the compatibility of node u with the routed nodes i and j. If either TWC_{iu} or TWC_{uj} is negative infinity $(-\infty)$, indicating an incompatible time window, the insertion heuristic moves on and considers the next edge, without wasting computational effort on calculating the insertion and selection criteria.

In the earlier example, eleven instances of infeasible time windows occur. If these instances are identified and eliminated, a computational saving in excess of 22% is achieved.

3.3.2 Identifying the seed customer

When looking at the TWCM for the example, it is clear that the incompatibility is distinct for specific nodes. It is therefor possible to identify *incompatible* nodes. As opposed to the two most common initialization criteria, namely *customer with earliest deadline*, and *furthest customer*, as suggested by Dullaert [19], this dissertation proposes the use of the TWCM to identify seed nodes based on their time window compatibility.

Table 3.2 indicates the number of instances where a node has an infeasible time window with another node, either as origin, or as destination. Both

Table 3.2: Number of infeasible time window instances

Node	**Number of infeasible time windows, as origin**	**Number of infeasible time windows, as destination**	**Total**
Depot	0	0	0
a	1	2	3
b	2	1	3
c^1	0	5	5
c^2	5	0	5
d	1	2	3
e	2	1	3

nodes c^1 and c^2 have five infeasible instances. The two artificial nodes are representing the same customer c. It can be concluded that customer c is the most incompatible node, and is therefore identified as the seed customer. Ties are broken arbitrarily. Should two nodes have the same number of infeasible time window instances, any of the two customers could be selected as seed customer.

It may be possible to not have any infeasible time window instances. In such a scenario, a *total compatibility* value can be determined for each node a, and is calculated as

$$\sum_{i=1, i \neq a}^{M} TWC_{ia} + \sum_{j=1, j \neq a}^{M} TWC_{aj} + TWC_{aa} \quad (3.8)$$

or

$$\sum_{i=1}^{M} TWC_{ia} + \sum_{j=1}^{M} TWC_{aj} - TWC_{aa} \quad (3.9)$$

where M refers to all the unrouted nodes, including all instances of those nodes that are split artificially. The customer with the lowest total compatibility is selected as seed customer.

3.4 Conclusion

A new concept, *Time Window Compatibility* (TWC) is introduced in this chapter. A matrix, referred to as the *Time Window Compatibility Matrix* (TWCM), is used as the mechanism to calculate the compatibility between all nodes in the network. A numerical example illustrates the use of the concept to reduce the computational burden of existing heuristics, and proposes an improved criteria for seed customer selection.

The time window compatibility is used in the next chapter when the formal model and initial solution algorithm is defined.

Chapter 4

Model definition

The purpose of the chapter is to depart on the first leg of the journey towards obtaining an initial solution to the extended vehicle routing problem, currently referred to as the *Vehicle Routing Problem with Multiple Constraints* (VRPMC). The *model development process* used to develop the model is taken from Taniguchi *et al.* [57] and is presented in figure 4.1.

Problem definition – The conceptual problem is defined in chapter 1.

Objective – As the model is concerned with determining an initial solution to a routing and scheduling problem, the result produced by the algorithm becomes the *objective* of the model. The choice of solution candidates are influenced by the mathematical objective function of the problem model.

Criteria – To elaborate on the criteria, a comprehensive mathematical model of the VRPMC is presented in section 4.1. The criteria define the solution space through multiple mathematical constraints.

System analysis – The analysis process involves identifying the essential components and interaction within the solution algorithm. Section 4.2.1 describe the interaction of the algorithm's logical processes at a high level.

System synthesis – Although Taniguchi *et al.* [57] specify that this involves expressing the model in mathematical terms, it was already formulated in chapter 2, and presented in its entirety in section 4.1. Synthesis, in this dissertation, is the process of constructing and documenting a robust algorithm that will serve as direct input to the coding stage .

Software development – A computer based procedure will be developed in *MATLAB*. This will allow the mathematical and logical procedures, developed during the synthesis stage, to be used to produce actual

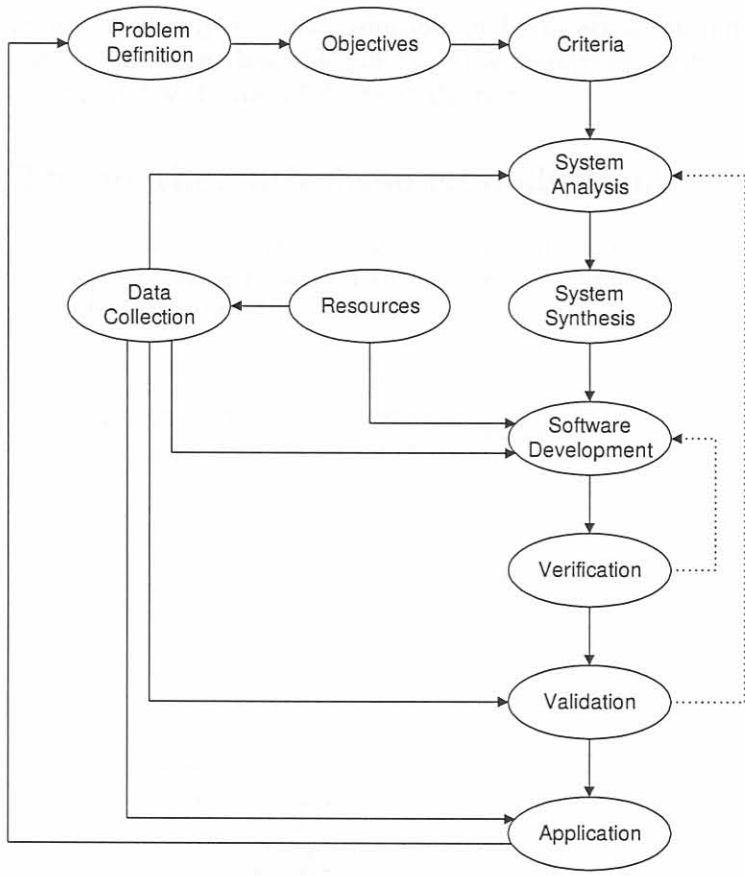

Figure 4.1: A model development process

quantitative results. The software development is further discussed in chapter 5.

Verification – Procedures are tested and checked for correct logical structure. This iterative process makes use of manually simulated and calculated instances, and compares the algorithm's output with its anticipated behavior.

Validation – At this stage the algorithm's output is compared with published results. The objective of validation is to determine if the initial solution created by the algorithm is comparable with those generated by accepted algorithms, as the result should only be marginally better, or worse, than previously published results.

Application – The algorithm will be tested in parallel with current scheduling applications. Given the nature of the algorithm, and the fact that the output is only an initial solution, and will act as an input to an optimization algorithm, the quality of the algorithm's output can not be compared with that of the final algorithm.

4.1 The mathematical model definition

All variables and concepts are defined in chapter 2, and the mathematical model is therefor presented without any declaration of variable or explanation of constraint.

$$\begin{aligned}\min z = &\sum_{i=0}^{N}\sum_{j=0,j\neq i}^{N}\sum_{k=1}^{K} c_{ij}x_{ijk} + \sum_{k=1}^{K}\sum_{j=1}^{N} f_k x_{0jk}\\ &+\sum_{i=1}^{N}\alpha_i \times \max\{0, L_i\} \end{aligned}\tag{4.1}$$

subject to

$$\sum_{j=1}^{N} x_{0jk} = \sum_{j=1}^{N} x_{j0k} = 1 \qquad \forall k = \{1,2,\dots,K\} \tag{4.2}$$

$$\sum_{j=1}^{N}\sum_{k=1}^{K} x_{0jk} \leq K \tag{4.3}$$

$$\sum_{i=1;i\neq j}^{N}\sum_{k=1}^{K} x_{ijk} = 1 \qquad \forall j \in \{1,2,\dots,N\} \tag{4.4}$$

$$\sum_{j=1;j\neq i}^{N}\sum_{k=1}^{K} x_{ijk} = 1 \qquad \forall i \in \{1,2,\dots,N\} \tag{4.5}$$

$$\sum_{i=1}^{N} q_i \sum_{j=0;j\neq i}^{N} x_{ijk} \leq p_k \qquad \forall k = \{1,2,\dots,K\} \tag{4.6}$$

$$a_0 = w_0 = s_0 = 0 \tag{4.7}$$

$$\sum_{k=1}^{K}\sum_{i=0;i\neq j}^{N} x_{ijk}(a_i + w_i + s_i + t_{ij}) \leq a_j \qquad \forall j \in \{1,2,\dots,N\} \tag{4.8}$$

$$e_i \leq (a_i + w_i) \leq l_i \qquad \forall i \in \{1,2,\dots,N\} \tag{4.9}$$

$$x_{ijk} \in \{0,1\} \tag{4.10}$$

4.2 System analysis

It is the objective of this dissertation to promote the use of a systematic approach to model development, as opposed to the rapid-prototyping approach often experienced in practise. To ensure that the algorithm acts in a coherent and logical manner, the algorithm is modelled at various levels prior to being coded.

4.2.1 Overview

A graphical overview of the algorithm is presented in figure 4.2. The

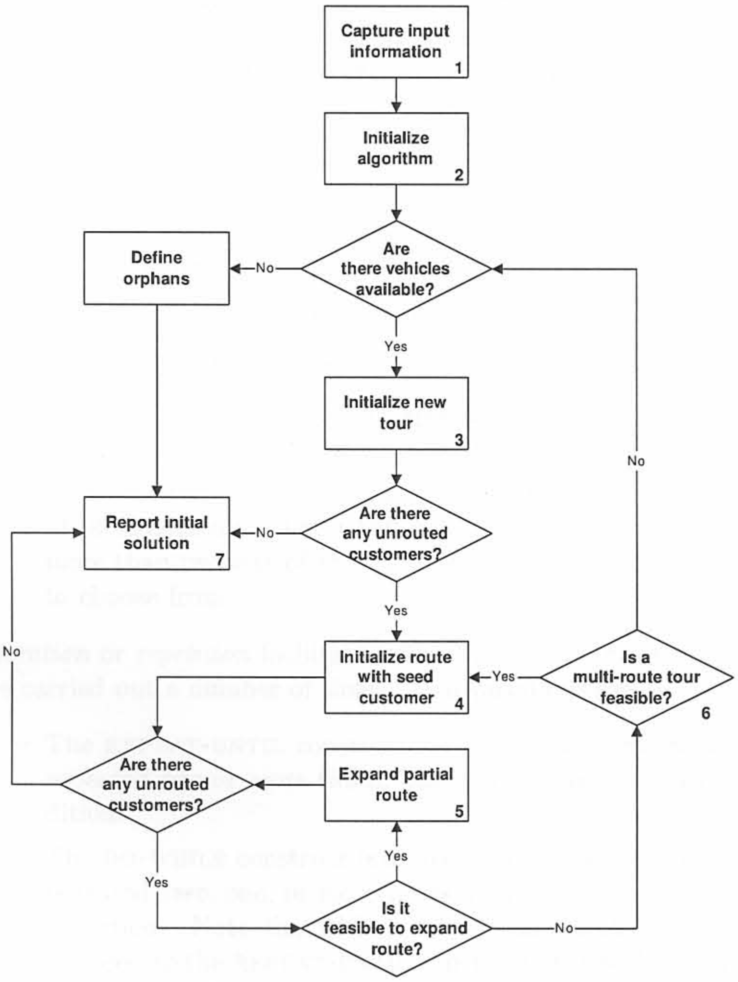

Figure 4.2: Overview of initial solution algorithm

number in the lower right-hand corner of a procedure, or decision in the flowchart, refers to the sequence of discussions in the following subsection with regards to specific algorithm detail that are highlighted.

4.2.2 Algorithm detail

Sections of the overview model is represented using *Structured English* – a language and syntax, based on the relative strengths of structured programming, and natural English [64]. *Structured English* is not pseudocode, as it does not concern itself with the declaration and initialization of variables, linking, and other technical issues. The *Structured English* sections aims to communicate unambiguous logic about the algorithm which is easy to understand, yet not open to misinterpretation [3]. Readability takes preference over programming preferences. It is a strict and logical form of English, and the following constructs reflect structured programming:

- *Sequencing* shows the order of processing a group of instructions – simple, declarative sentences, following one another – without repetition and branching. Compound sentences are avoided, as they create ambiguity. Strong action verbs, such as GET, FIND, CALCULATE, UPDATE, SORT, etc. are used.
- *Selection* or *decision structure* facilitates the choice of actions under well-specified conditions. Variations of sequencing include:
  - the IF-THEN-ELSE construct specifies the actions that must be taken if a specific condition, or set of conditions, are all true.
  - The CASE construct is an elegant substitute for multiple IF-THEN-ELSE statements. The CASE construct is used where there are more than two sets of actions, based on well-specified conditions, to choose from.
- *Iteration* or *repetition* facilitates the same action, or set of actions, to be carried out a number of times. Two variations are
  - The REPEAT-UNTIL construct indicates that certain actions are repeated one or more times, based on the value of a stated condition.
  - The DO-WHILE construct indicates that certain actions are to be repeated *zero*, one, or more times, based on the value of a stated condition. Note that this construct need not be executed, as opposed to the REPEAT-UNTIL construct that will execute the set of actions at least once.

Blocking and indentation are used to indicate the beginning and end of constructs, as opposed to terms such as ENDIF, ENDCASE, ENDDO and ENDREPEAT, as these give the algorithm too much of a programming look and feel. Uppercase terms in the algorithm with italicized bold typeface indicate a variable set that is used in the coding of the algorithm. These sets are treated in square brackets in the document text, for example [VEHICLE] indicates the set of vehicles. The row numbers on the left indicate the line number in the complete algorithm. The algorithm presented in the dissertation is aggregated to eliminate unnecessary technical information helpful during the programming of the algorithm, hence the irregular numbering.

Figure 4.3 describes the capturing of input information. A list of all the technical field names appear in Appendix A.

```
Capture input information
    Capture vehicle information in VEHICLES
    Set average speed as 55 km/h
    Sort available vehicles
        Clear and set VEHAVAIL as an available vehicle matrix
        for all available vehicles in VEHICLES
            Add vehicle to VEHAVAIL
            Sort VEHAVAIL in ascending order on <volumetric capacity>

    Capture general CUSTOMER information
        Capture customer information in CUSTOMER
        for each entry, i, in CUSTOMER
            if CUSTOMER has multiple time windows
                Split customer into customer(i).tw artificial customers
                Add artificial customer to ARTIF
                Capture the time window information for each ARTIFicial customer
            else
                Add the CUSTOMER as a single ARTIFicial customer
                Capture the time window information for the single ARTIFicial customer
        Calculate the DISTance matrix between all the ARTIFicial nodes
```

Figure 4.3: Capture input information

The depot is captured as the first customer. If a customer specifies more than one time window, the customer is artificially split into n customers, each with a single time window, where n indicate the number of time windows specified. Once the customers are split artificially, reference will only be made to nodes – with each node indicating an artificial customer in the [ARTIF] set. Figure 4.4 describes the initialization process.

```
Initialise algorithm
    Set the ROUTED matrix as empty
    for all the ARTIFicial nodes, except the depot (node 1)
        Add the ARTIFicial node to the UNROUTED matrix
```

Figure 4.4: Initialize algorithm

If there are vehicles available, a new *tour* is created. A tour can be made up of one or more *routes*. The initialization of a tour involves assigning the smallest available vehicle to the tour, and matching the tour capacity to that of the vehicle. This is indicated in figure 4.5.

```
Initialise TOUR
    Set the TOUR index (t) to 1
    Establish the starting time for the TOUR
        Starting time for the current TOUR is e0 + s0
        (It is assumed that vehicles are not loaded at the beginning of the depot's time window)
    Assign vehicle to TOUR
        Set the first vehicle in VEHAVAIL as the current vehicle for the TOUR
        Update vehicle availability
            Locate the current vehicle in VEHICLE
            Set vehicle(k).availability = 0
            Recalculate VEHAVAIL
```

Figure 4.5: Initialize new tour

Once a tour has been created, one or more routes are established to make up the route. The iterative route creation process starts with the initialization of a new route. This entails assigning the route to the current tour, adding the depot as first and last node on the route, and identifying and inserting the *seed customer*: the first customer, other that the depot, to be added onto the route. The theory behind determining the seed customer has been elaborated upon in section 3.3. The algorithmic procedure for route initialization is indicated in figure 4.6. Nodes in the [UNROUTED] set are evaluated for insertion on the partially constructed route. The iterative route-building procedure is indicated in figure 4.7.

The concept of scheduling a vehicle to complete multiple routes (referred to as *double scheduling*), is difficult to implement in solution algorithms. The procedure followed in this dissertation to determine multi-route feasibility in a tour, is indicated in figure 4.8. When a vehicle returns to the depot at the end of a route, the multi-route feasibility check procedure determines if the depot's time window is still open after the vehicle's capacity has been replenished/renewed. It might be realistic to add some time to the *potential route* to allow the vehicle to at least service one node. The additional time added, conveniently referred to as *minimum route time parameter*, is different for each environment, and has been set to one hour in this dissertation. The effect will be that an empty route may be assigned to a number of tours when the initial solution is presented. To overcome the effect of empty routes, the final reporting procedure have been adapted to check for empty routes prior to reporting the initial solution. The procedure in indicated in figure 4.9.

```
Initialise ROUTE with seed customer
    Set ROUTE index (r) to 1
    Assign current ROUTE to current TOUR
    Establish the starting time for the TOUR
    Set ROUTE load to zero

    Assign the depot as starting and ending node for the current ROUTE
    Select a seed customer from the UNROUTED nodes
        Calculate the time window compatibility matrix (TWCM) for all UNROUTED nodes
            for each node combination (a,b) where node b is serviced after node a
                Calculate the earliest possible arrival at b as arrival_earliest
                Calculate the latest possible arrival at b as arrival_latest
                if the earliest possible arrival at b is before the latest allowed arrival at b
                    Calculate time window compatibility (TWC)
                        TWC_ab = min {arrival_latest, l_b} - max {arrival_earliest, e_b}
                else
                    TWC is negative infinity

        Calculate the number of infeasible time windows for each UNROUTED node
            for each UNROUTED node (i)
                Determine how many times in row i of TWCM is TWC negative infinity
                Determine how many times in column i of TWCM is TWC negative infinity
                Calculate the total number of infeasibilities by adding row and column count

        if there are infeasible time windows for any UNROUTED node
            The seed customer is the node with the most number of infeasible time windows
        else
            Calculate the COMPATIBILITY vector
                for each UNROUTED node (a) in the TWCM
                    row = TWCM(a,:)
                    column = TWCM(:,a)
                    compatibility(a) = sum(row) + sum(column) - TWCM(a,a)
            The seed customer is the node with the lowest COMPATIBILITY

    Insert seed customer
        Insert seed customer on current ROUTE
        Update UNROUTED customers
            Remove seed customer from UNROUTED
            Remove any other artificial nodes related to seed customer from UNROUTED
            Update ROUTE load
```

Figure 4.6: Initialize new route

```
Expand partial ROUTE
 while UNROUTED is not empty and there are customers that fit into the current ROUTE
    Clear the node selection matrix C2
    for each UNROUTED node (u)
       Clear the node insertion matrix C1
       Select the best position to insert node u on the current ROUTE
          for each edge (i,j) on the current ROUTE
          Determine feasibility to add node u
             Infeasible if either TWC_iu or TWC_uj is unfeasible
             Infeasible if TOUR capacity is exceeded by u
          if it is feasibile to evaluate node u between i and j
             Update the C1 vector for the insertion positions
                Calculate c_1(i,u,j)
                Add the c_1(i,u,j) value to C1(m).value
          else
             Check next edge on current ROUTE
          Select the best edge (i*,j*) based on the lowest C1 matrix value

       Update the C2 matrix for the insertion position
          Calculate c_2(i*,u,j*)
          Add the c2(i*,u,j*) value to the C2 matrix

    Sort C2 in ascending order
    Find first time-feasible node (u*), starting at the beginning of C2
       While no u* has been found, and end of C2 has not been reached
          Check for time feasibility
          if feasible
             Identify applicable node as u*
          else
             Check next element of C2

    if a unique u* node has been identified
       Insert node u*
          Update UNROUTED customers
             Remove u* from UNROUTED
             Remove any other artificial nodes related to u* from UNROUTED
          Update ROUTE
             Update ROUTE load
             if new vehicle has been indicated
                if Q^new > Q
                   Find the smallest available vehicle to service Q^new
                      Update VEHAVAIL
                         Change the availability status of the current vehicle to available
                         Change the availability status of the new vehicle to unavailable
                         Assign new vehicle to current TOUR
                         Recalculate VEHAVAIL

          Recalculate ROUTE schedule for nodes
             Actual start-time at origin (a_o) is the start-time indicated for the current route
             for each node (i) on the current ROUTE, except the depot at both ends
                a_i = max{e_i , a_{i-1} + s_{i-1} + t_{i-1,i}}
                w_i = max{0 , e_{i+1} - (a_i + s_i + t_{i,i+1})}
             Calculate actual arrival at the depot (n^th node) at the end of the current ROUTE
                a_n = a_{n-1} + s_{n-1} + t_{n-1,n}}
    else
       Initialize new ROUTE
```

Figure 4.7: Expand partially constructed route

```
Expand TOUR
    Determine multi-route feasibility
        Check the actual arrival time at the depot of the previous ROUTE of the current TOUR (a_n)
        if a_n + s_o + 1 hour < l_o^max
            then feasible
        else
            infeasible
    if feasible
        Initialize new ROUTE
    else
        if the last ROUTE of the TOUR has no nodes other than the depot
            Eliminate ROUTE from TOUR
        Initialize new TOUR
```

Figure 4.8: Checking for multi-route feasibility

```
Define ORPHANs
    if UNROUTED is not empty
        Assign all elements in UNROUTED to ORPHANs
        Clear UNROUTED

Report initial solution
    Calculate the OBJective function value for the initial solution
    Report initial solution
        for each TOUR
        Report all TOUR and ROUTE information
```

Figure 4.9: Report initial solution

4.3 Conclusion

The chapter introduced the model development process. The objectives and criteria are stipulated in the mathematical definition of the problem. This chapter elaborates on the *system analysis* and *synthesis*. The proposed initial solution algorithm is presented at a high level, with selective detail given in *Structured English*. The complete algorithm is presented in Appendix B. Chapter 5 discusses the implementation, and the results, of the proposed algorithm as coded in *MATLAB*.

Chapter 5

Results

This chapter reports on the results of the initial solution algorithm, as programmed in *MATLAB*. The chapter begins with a discussion on how comparative data sets were established for computational purposes, along with a motivation for modifying existing data sets. The results are presented and discussed, focusing on the contribution of the *time window compatibility* on both the computational burden, and the quality of the initial solution.

5.1 The basic Solomon sets

Solomon [46] discusses the generation of data sets for the *Vehicle routing and scheduling problems with time window constraints* (VRPSTW), and indicates that the design of these data sets highlight several factors that affects the behavior of his routing and scheduling heuristics. The corresponding six data sets, referred to as $R1$, $R2$, $C1$, $C2$, $RC1$, and $RC2$, are often used and referred to in literature.

5.1.1 Geographical distribution

The data used for the customer coordinates and demands are based on the work of Christofides *et al.* [13], and are classified into one of three categories:

- Randomly distributed customers (denoted by an R prefix)
- Clustered customers (denoted by a C prefix)
- Semi-clustered customers (denoted by an RC prefix)

By semi-clustered is implied a random mix of both randomly distributed and clustered customers.

5.1.2 Scheduling horizon

The length of the route-time is regarded as a capacity constraint and, along with the vehicle capacities, limit the number of customers serviced by a specific vehicle. The short scheduling horizon problems are denoted by a "1" as a suffix. The problems denoted by a "2" suffix, on the other hand, have a large scheduling horizon, and along with the vehicle capacities, allow a larger number of customers to be serviced by a single vehicle.

5.2 Test data

Solomon's data sets have become a benchmark for vehicle routing problem variants, although the sets are often used with some modification due to the specific variant's particularity. The published data sets [47] assume a homogeneous fleet, indicate a given vehicle capacity, and assume infinite availability of vehicles. Furthermore the sets only indicate a single time window for each geographically distributed customer.

The aim of the algorithm developed in this dissertation is to test the feasibility of integrating multiple time windows, a heterogeneous fleet, and double scheduling into an initial solution heuristic. As none of these three specific variants are addressed by the Solomon data sets, it was necessary to generate a unique data set to accommodate, and integrate, the problematic variants, yet still have resemblance to the familiar benchmarks in literature. The problem of developing an appropriate data set meant addressing both multiple time windows, and a heterogeneous fleet. Double scheduling is a function of the algorithm, and does not require any manipulation of the data sets.

5.2.1 Incorporating multiple time windows

It originally seemed appropriate to use two of the Solomon sets for a specific class of problem, each with 100 customers and a single, unique time window, to create a new set of 100 customers with multiple time windows. It turned out to be futile, as the time windows for different data sets of the same classification, as presented by Solomon [47], had very similar, if not exactly the same, time windows. The extended data sets presented by Homberger [24] are developed in the same manner as Solomon's sets, but have sets with 200, 400, 600, 800, and 1000 customers.

For each of the six problem classes, an extended set with 200 customers from Homberger's sets is used to create a new set with 100 customers, but with two time windows. Table 5.1 illustrates an excerpt from the data used to create a new set for the $C1$ class. In the Homberger data sets, the *cus-*

tomer number, *x-coordinate*, *y-coordinate*, *demand*, *service time*, and the *start-* and *end times* of a single time window, are given. Table 5.1 indicates

Table 5.1: Constructing data sets with multiple time windows

Customer				Time window 1		Time window 2	
(i)	x	y	...	e_i^1	l_i^1	e_i^2	l_i^2
$\vdots$							
4	4	28	...	616	661	128	195
5	25	26		128	179	142	197
6	86	37		478	531	754	814
7	1	109		616	680	583	647
8	6	135		351	386	1011	1077
9	32	79	...	655	721	950	1003
$\vdots$							

the case where the single time windows of customers 101 through 200 have been used as *second* time windows for customers 1 through 100. For example, the time window for customer 4 is given as $(616, 661)$, measured in minutes from $0:00am$. Customer 104's time window is given as $(128, 195)$ in the original data set. Customer 104's time window now becomes the second time window for customer 4, while all other data about customers 101 through 200 is disregarded. Observe, however, that the two time windows specified for customer 5 overlap, and do not yield two unique time windows as is the case for customer 4.

Procedure 1 indicates the procedure used to manipulate the time windows from multiple sets into a single data set. Where time windows overlap,

Procedure 1 Creating a data set with multiple time windows

if either $e_i^1 > l_i^2$ or $e_i^2 > l_i^1$ **then**
  Number of time windows is 2
  $E_i^1 = \min\{e_i^1, e_i^2\}$
  $L_i^1 = \min\{l_i^1, l_i^2\}$
  $E_i^2 = \max\{e_i^1, e_i^2\}$
  $L_i^2 = \max\{l_i^1, l_i^2\}$
else
  Number of time windows is 1
  $E_i^1 = \min\{e_i^1, e_i^2\}$
  $L_i^1 = \max\{l_i^1, l_i^2\}$
end if

the new, single time window, is defined to start at the opening of the earlier time window, and end at the closing of the later time window. After the manipulation of the data, the start of the first time window for customer i is denoted by E_i^1, and the end of the time window by L_i^1. Where only one time window exist, no second time window is specified. Alternatively, the start of the second time window for customer i is denoted by E_i^2, and the end of the time window by L_i^2.

5.2.2 Incorporating a heterogeneous fleet

Liu and Shen [31, 32] propose a specific fleet structure with the introduction of their insertion-based savings heuristic for a heterogeneous fleet. The proposed cost structure sees the cost of a vehicle more than doubles when its capacity doubles. Although Dullaert *et al.* [19] challenges the cost structure presented by Liu and Shen, they did not propose a new cost structure. It was therefor considered appropriate to use the given fleet composition as indicated in Table 5.2. It should be noted that Liu and Shen assumed an infinite

Table 5.2: Heterogeneous fleet data

Problem class $R1$

vehicle	capacity	cost
A	30	50
B	50	80
C	80	140
D	120	250
E	200	500

Problem class $R2$

vehicle	capacity	cost
A	300	450
B	400	700
C	600	1200
D	1000	2500

Problem class $C1$

vehicle	capacity	cost
A	100	300
B	200	800
C	300	1350

Problem class $C2$

vehicle	capacity	cost
A	400	1000
B	500	1400
C	600	2000
D	700	2700

Problem class $RC1$

vehicle	capacity	cost
A	40	60
B	80	150
C	150	300
D	200	450

Problem class $RC2$

vehicle	capacity	cost
A	100	150
B	200	350
C	300	550
D	400	800
E	500	1100
F	1000	2500

number of each of the vehicles types. To accommodate the infinite number of vehicles into the data sets generated for the algorithm, the number of vehicles for each type is said to be the number of vehicles needed to service the demand of all customers, if *only* that type of vehicle was available.

5.3 Results

Six data sets, one for each class of problem, were generated. The algorithm was coded in *MATLAB 6.5 release 13*. The algorithm was executed on three similar *Pentium IV 1.6GHz* computers, each with *256Mb RAM*, with each class being executed at least once on each of the computers. For each class of problem, five runs of the algorithm were executed. The results presented in this chapter is in each case the average of the five runs. Appendix C contains the solution details with respect to the specific customers assigned to the various routes, the orphaned customers (if any), as well as the total scheduling distance.

A summary of the initial solutions generated by the proposed algorithm is presented in Table 5.3. The influence of the problem characteristics (de-

Table 5.3: Summary of computational results

Problem class	**Average CPU time** (seconds)	**Number of Tours**	**Number of Routes**	**Total scheduling distance** (kilometers)
$R1$	3180	18	71	11260
$R2$	31730	5	7	8722
$C1$	2170	11	22	7330
$C2$	65650	4	6	7240
$RC1$	1120	26	48	8706
$RC2$	6960	12	19	7748

picted by the problem class) can be appreciated when comparing the average number of customers per route for the type 1 and type 2 classes. The type 1 classes, with a narrow scheduling horizon, has an average of 2.13 customers per route, while the type 2 classes with longer scheduling horizons have, on average, 9.38 customers per route.

It should be noted, however, that the average CPU time is extremely high. The reason for the computationally expensive CPU results is twofold:

- The algorithm was coded in *MATLAB* with the specific intent of the candidate to learn the software package. The code structure may therefore be inefficient, with numerous opportunity for code optimization.

- The file format in which *MATLAB* was executed is a non-compiled $^\star.M$ file, which is, from a computational point of view, significantly slower than a compiled **.dll* file. The decision to not compile was influenced by compiling software availability at the time of testing the algorithm.

Although there is ample opportunity to improve the algorithms technical performance, the results, in terms of number of tours, number of routes per tour, as well as the actual scheduling distance, proves to have significant contribution to the field of vehicle routing problem algorithms. Figures 5.1 through 5.6 indicate the cumulative CPU time for each class of problem.

5.4 Evaluating the contribution of TWC

It is important to evaluate the contribution that the proposed *time window compatibility* has on the results of the algorithm. For this purpose, a comparative *control algorithm* is created. The control algorithm differs only in two respects from the proposed algorithm:

- It does not evaluate nodes for *time window compatibility* when calculating the insertion criteria, and therefore considers every node for insertion on every edge of a partially constructed route.
- As no *time window compatibility* is calculated for any node, the initialization criteria is changed to identify the seed customer as the unrouted customer with the earliest deadline.

The control algorithm is executed for two extreme problem classes, namely clustered customers with a short scheduling horizon ($C1$), and uniformly distributed customers with a long scheduling horizon ($R2$). Five runs for each class were executed in a similar fashion to the proposed algorithm, with regards to the computers used, and the distribution of runs on the computers.

Figure 5.7 illustrates the significant improvement that the *time window compatibility* has on the computational burden for the $C1$ class of problems. The average CPU time is down from 10950 seconds to 2170, an improvement in excess of 80%. Not only did the *time window compatibility* reduce the computational burden, but it also improved the quality of the initial solution by almost 13%. By the *quality of the solution* is implied the total scheduling distance. The comparative results summary is given in Table 5.4.

Figure 5.8 indicates a computational saving of more than 24% for the $R2$ class problems, while the quality of the initial solution itself is improved by almost 13%. The $R2$ results correspond with the expectation that the

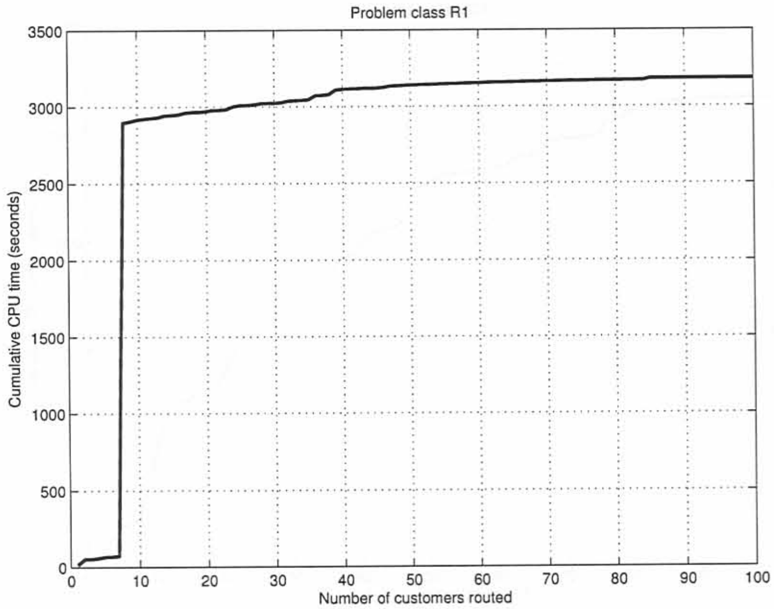

Figure 5.1: Cumulative progress for the $R1$ class problem

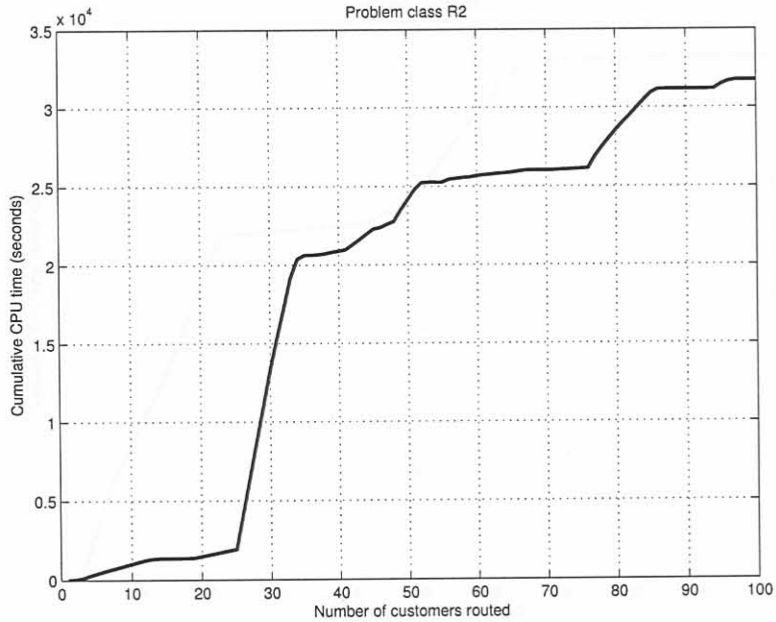

Figure 5.2: Cumulative progress for the $R2$ class problem

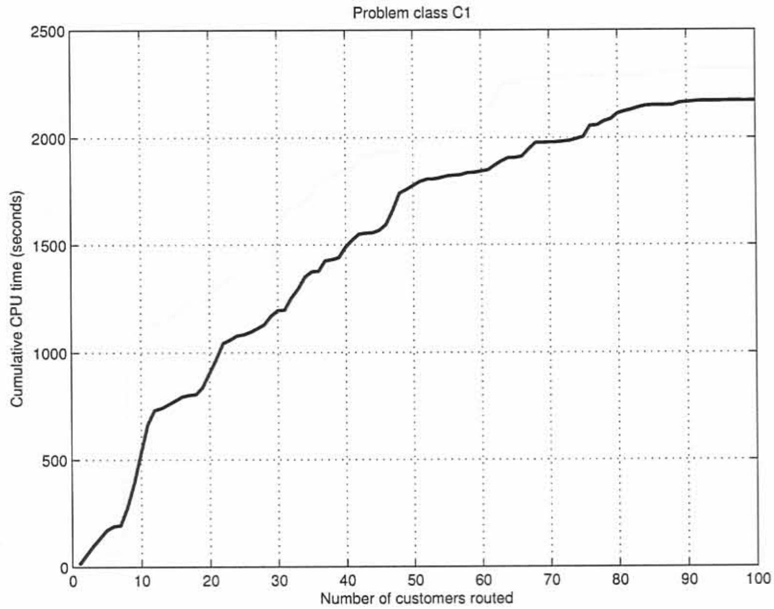

Figure 5.3: Cumulative progress for the $C1$ class problem

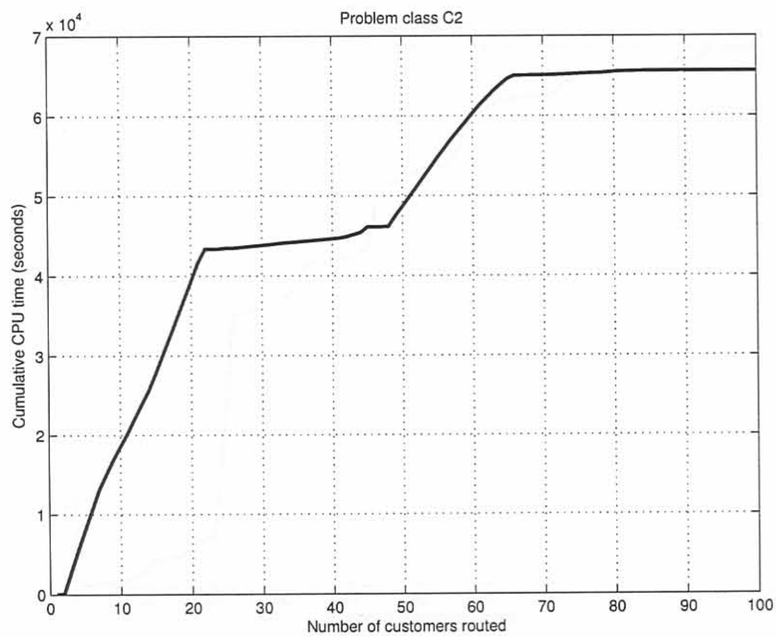

Figure 5.4: Cumulative progress for the $C2$ class problem

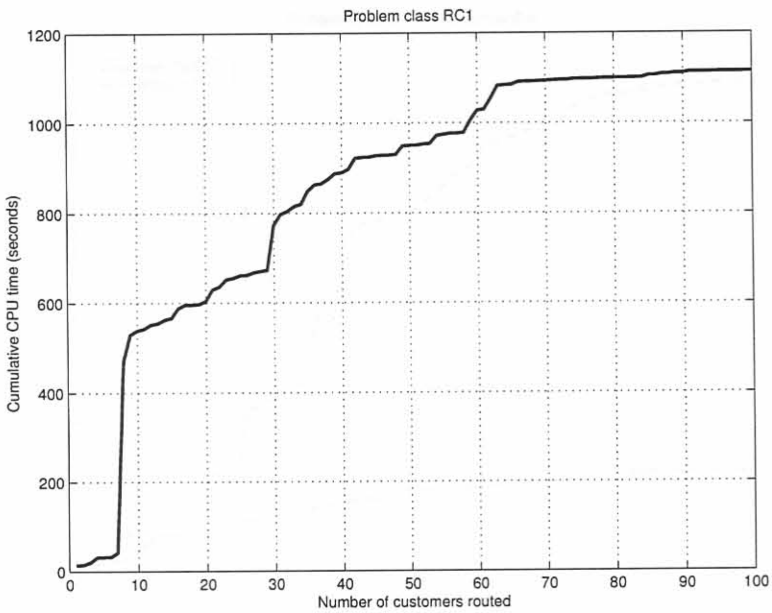

Figure 5.5: Cumulative progress for the $RC1$ class problem

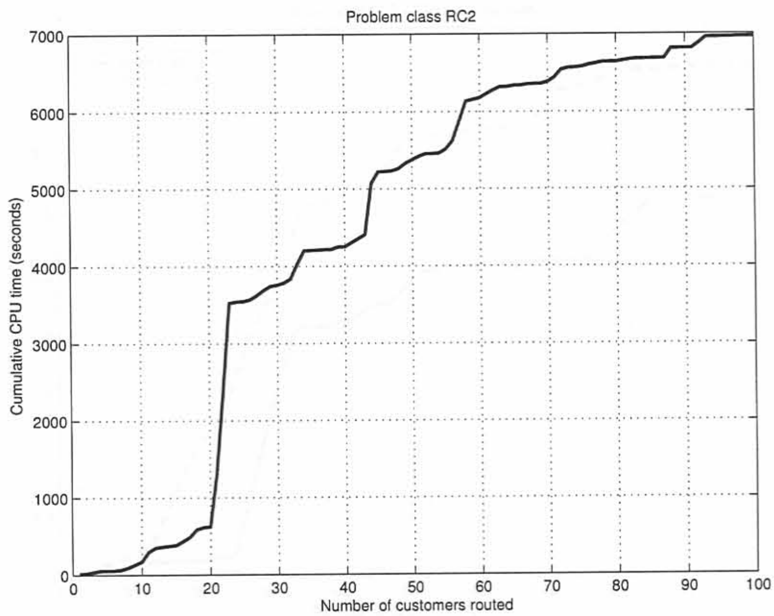

Figure 5.6: Cumulative progress for the $RC2$ class problem

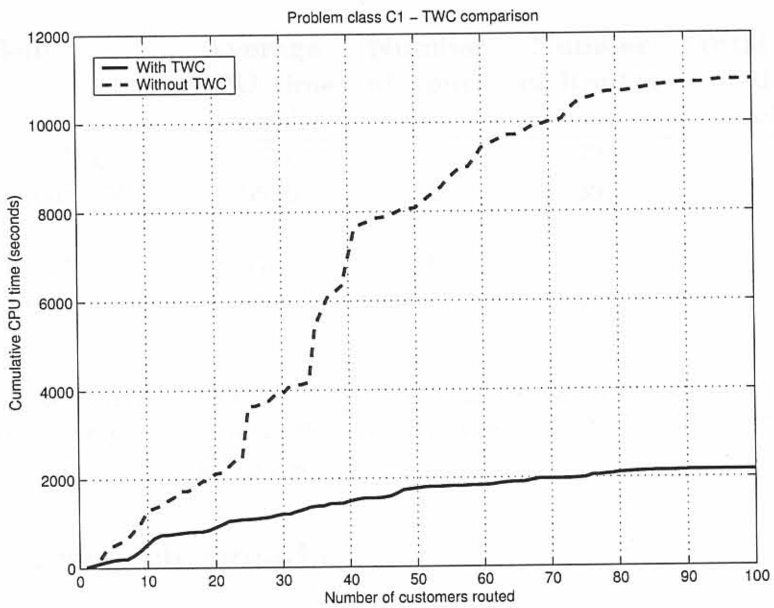

Figure 5.7: The effect of time window compatibility on the $C1$ class

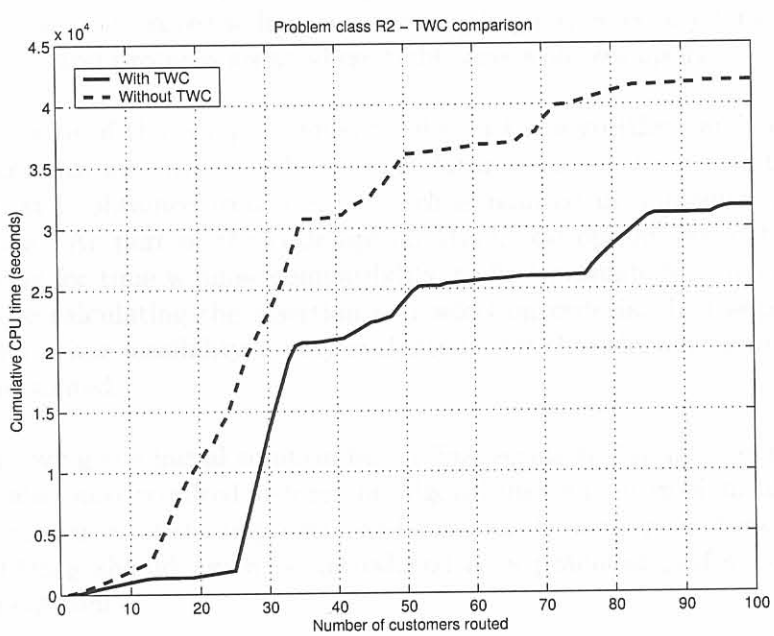

Figure 5.8: The effect of time window compatibility on the $R2$ class

Table 5.4: Summary of comparative results

Problem class	**Average CPU time** (seconds)	**Number of Tours**	**Number of Routes**	**Total scheduling distance** (kilometers)
$C1$ with TWC	2170	11	22	7330
$C1$ without TWC	10950	12	39	8876
$R2$ with TWC	31730	4	7	8722
$R2$ without TWC	41920	2	13	9990

time window compatibility will have less of an impact if the scheduling horizon is relatively large, as is the case for the type 2 problems. Still, the computational saving is significant.

5.5 Research agenda

Chapter 1 states the research question. The objective is to attempt to create an initial solution that caters for multiple time windows, a heterogeneous fleet, as well as double scheduling. Although the results generated during this dissertation is computationally expensive, the aim has been achieved, and with unexpectedly high returns. The concept of *time window compatibility* has proved to have a staggering impact, especially for clustered customers, and problem areas where tight time windows apply.

The issue of the computational burden of the algorithm, and technical code optimization, will be addressed in future research to ensure that the value that is obtained from this research is realized in implementable applications. As part of the code optimization, one option would be, after evaluating for time window compatibility, to first evaluate for time feasibility before calculating the insertion and selection criteria. In the proposed algorithm, time feasibility is only evaluated after the selection criteria have been calculated.

Improving the initial solution into a final solution through meta heuristics is also also required before the algorithms can be implemented. In the development of the improvement heuristic, the concept of *time window compatibility* should again be introduced as a potential performance improvement tool.

A number of minor parameters in the calculation of the insertion and selection criteria have been taken directly from literature, based on their relative performance. These parameters could be challenged to ensure that they contribute to the particularity of the proposed algorithm.

Initial solutions are only the first step in an optimization process such as the vehicle routing problems. The next step would be to create a number of unique (different) initial solutions. The value of having varying initial solutions is that the *Tabu Search* improvement meta heuristic, which is often used for vehicle routing problems, requires a set of unique initial solutions before being invoked. It is envisaged that a single customer will be removed from the original customer data list before invoking the initial solution algorithm. A simple insertion heuristic will then be used to insert the omitted customer into the generated solution after it is created. It is anticipated that if the process is repeated for different customers, different solutions will be generated.

5.6 Conclusion

City Logistics is concerned with the mobility of cities, and often aims to establish best practices and initiatives to improve the state of transport planning. The algorithm proposed in this dissertation contributes to the process of optimizing urban logistics as it proves that initiatives such as time windows and load factors can be planned for by shippers and carriers. The results prove that multiple variants of the vehicle routing problem can be integrated into the initial solution algorithm. The increased complexity is addressed by the time saving concept of *time window compatibility*, which proved to have a significant impact on both the computational burden, and the quality of the initial solution.

Appendix A

Technical fields

Description	Field name
Vehicle's description	$vehicle(k).description$
Vehicle's volumetric capacity	$vehicle(k).cv$
Vehicle's weight capacity	$vehicle(k).cw$
Vehicle's fixed cost	$vehicle(k).f$
Vehicle's availability	$vehicle(k).avail$
Customer's name	$customer(i).name$
Customer's geographical position	$customer(i).x$
	$customer(i).y$
Customer's volumetric demand	$customer(i).dv$
Customer's weight demand	$customer(i).dw$
Service time at customer	$customer(i).st$
Maximum lateness allowed at customer	$customer(i).Lmax$
Lateness penalty factor for customer	$customer(i).alpha$
Number of time windows for customer	$customer(i).tw$
Artificial customer index	$customer(i).index$
Artificial customer reference	$customer(i).ref$
Earliest allowed time of arrival at artificial customer	$customer(i).earliest$
Latest allowed time of arrival at artificial customer	$customer(i).latest$
Actual time of arrival at artificial customer	$customer(i).actual$

Appendix B

Complete algorithm

```
Capture input information
    Capture vehicle information in VEHICLES
        for each vehicle k
            Capture vehicle registration number as vehicle(k).description
            Capture vehicle's volumetric capacity as vehicle(k).cv
            Capture vehicle's weight capacity as vehicle(k).cw
            Capture vehicle's fixed cost as vehicle(k).f
            Capture vehicle availability as vehicle(k).avail

    Set average speed as 55 km/h
    Sort available vehicles
        Clear and set VEHAVAIL as an available vehicle matrix
        for all available vehicles in VEHICLES
            Add vehicle to VEHAVAIL
            Sort VEHAVAIL in ascending order on <volumetric capacity>

    Capture general CUSTOMER information
        Capture customer information in CUSTOMER
        for each customer i
            Capture customer's name as customer(i).name
            Capture customer's position as customer(i).x and customer(i).y
            Capture customer's volumetric demand as customer(i).dv
            Capture customer's weight demand as customer(i).dw
            Capture service time at customer as customer(i).st
            Capture maximum lateness at customer as customer(i).Lmax
            Capture lateness penalty factor for customer as customer(i).alpha
            Capture number of time windows for customer as customer(i).tw
```

for each entry, *i*, in ***CUSTOMER***
if ***CUSTOMER*** has multiple time windows
Split customer into *customer(i).tw* artificial customers
Add artificial customer to ***ARTIF***
Capture the time window information for each ***ARTIF***icial customer
else
Add the ***CUSTOMER*** as a single ***ARTIF***icial customer
Capture the time window information for the single ***ARTIF***icial customer
Calculate the ***DIST***ance matrix between all the ***ARTIF***icial nodes

Initialise algorithm
Set the ***ROUTED*** matrix as empty
for all the ***ARTIF***icial nodes, except the depot (node 1)
Add the ***ARTIF***icial node to the ***UNROUTED*** matrix

Initialise TOUR
Set the ***TOUR*** index (t) to 1
Establish the starting time for the ***TOUR***
Starting time for the current TOUR is $e_0 + s_0$
(It is assumed that vehicles are not loaded at the beginning of the depot's time window)
Assign vehicle to ***TOUR***
Set the first vehicle in ***VEHAVAIL*** as the current vehicle for the ***TOUR***
Update vehicle availability
Locate the current vehicle in ***VEHICLE***
Set *vehicle(k).availability = 0*
Recalculate ***VEHAVAIL***

Build TOUR
While ***UNROUTED*** is not empty
Initialise ROUTE with seed customer
Set ***ROUTE*** index (r) to 1
Assign current ***ROUTE*** to current ***TOUR***
Establish the starting time for the ***TOUR***
Set ***ROUTE*** load to zero

Assign the depot as starting and ending node for the current ***ROUTE***
Select a seed customer from the ***UNROUTED*** nodes
Calculate the time window compatibility matrix (***TWCM***) for all ***UNROUTED*** nodes
for each node combination (a,b) where node b is serviced after node a
Calculate the earliest possible arrival at b as *arrival_earliest*
e_a - the earliest allowed arrival at node u
s_a - the service time at node a
t_{ab} - the travel time between node a and node b
$arrival\_earliest = e_a + s_a + t_{ab}$
Calculate the latest possible arrival at b as *arrival_latest*
l_a - the latest allowed arrival at node a

```
                s_a - the service time at node a
                t_ab - the travel time between node a and node b
                arrival_latest = l_a + s_a + t_ab
            if the earliest possible arrival at b is before the latest allowed arrival at b
                Calculate time window compatibility (TWC)
                    TWC_ab = min {arrival_latest , l_b} - max {arrival_earliest , e_b}
            else
                TWC is negative infinity

    Calculate the number of infeasible time windows for each UNROUTED node
        for each UNROUTED node (i)
            Determine how many times in row i of TWCM is TWC negative infinity
            Determine how many times in column i of TWCM is TWC negative infinity
            Calculate the total number of infeasibilities by adding row and column count

    if there are infeasible time windows for any UNROUTED node
        The seed customer is the node with the most number of infeasible time windows
    else
        Calculate the COMPATIBILITY vector
            for each UNROUTED node (a) in the TWCM
                row = TWCM(a,:)
                column = TWCM(:,a)
                compatibility(a) = sum(row) + sum(column) - TWCM(a,a)
        The seed customer is the node with the lowest COMPATIBILITY

  Insert seed customer
    Insert seed customer on current ROUTE
    Update UNROUTED customers
        Remove seed customer from UNROUTED
        Remove any other artificial nodes related to seed customer from UNROUTED
        Update ROUTE load

Expand partial ROUTE
  while UNROUTED is not empty and there are customers that fit into the current ROUTE
    Clear the node selection matrix C2
    for each UNROUTED node (u)
        Clear the node insertion matrix C1
        Select the best position to insert node u on the current ROUTE
            for each edge (i,j) on the current ROUTE
            Determine feasibility to add node u
                Infeasible if either TWC_iu or TWC_uj is unfeasible
                Infeasible if TOUR capacity is exceeded by u
            if it is feasibile to evaluate node u between i and j
                Update the C1 vector for the insertion positions
                    Calculate c_1(i,u,j)
                        Calculate c_11(i,u,j)
```

d_{ij} - the distance between nodes i and j
$c_{11}(i,u,j) = d_{iu} + d_{uj} - \mu d_{ij}$, $\mu >= 0$ (in Dullaert, $\mu = 1$)
Calculate $c_{12}(i,u,j)$
e_i - earliest allowed arrival at node i
w_i - the waiting time before node i
a_i - the actual start time before node node i
s_i - the service time at node i
t_{ij} - the travel time between nodes i and j
b_i - the original start of service at node i
b_i^{new} - the start of service at node i after node u has been inserted

$b_u = max\{e_u , a_i + s_i + t_{iu}\}$
$b_j = a_j$
$b_i^{new} = max\{e_i , b_u + s_u + t_{uj}\}$
Calculate $c_{12}(i,u,j) = b_j^{new} - b_j$
Calculate $c_{13}(i,u,j)$
Select to use either *ACS*, *AOOS*, or *AROS*
ACS
Q - the load of the current vehicle before node u
$\underline{Q}$ - the max capacity of the current vehicle before node u
Q^{new} - the new load of the vehicle after node u
$\underline{Q}^{new}$ - the new max capacity of the vehicle after node u
$F(C)$ - the fixed cost of the smallest available vehicle that can service a demand of size C for a subtour
$ACS = F(Q^{new}) - F(Q)$
if new vehicle is indicated
flag new vehicle number

AOOS
$AOOS = ACS - F(\underline{Q}^{new} - Q^{new})$
if new vehicle is indicated
flag new vehicle number

AROS
$F'(C)$ - the fixed cost of the largest available vehicle whose capacity is less than or equal to C
$P(z)$ - the capacity of the smallest available vehicle that can service a demand of z
L $\omega = P(z_i + z_j) - P(max\{z_i, z_j\})$
$\delta(\omega) = 1$ *if* $\omega > 0$, *otherwise* 0
$\delta(\omega) = 1$ *if* $Q^{new} > \underline{Q}$, *otherwise* 0
$AROS = ACS - \delta(\omega)F'(\underline{Q}^{new} - Q^{new})$
$c_{13}(i,u,j)$ = *any one of ACS, AOOS, or AROS*
if new vehicle is indicated
flag new vehicle number

a_i - weight factors. The weight need not add up to 1
$c_1(i,u,j) = a_1 c_{11}(i,u,j) + a_2 c_{12}(i,u,j) + a_3 c_{13}(i,u,j)$

Add the $c_1(i,u,j)$ value to *C1(m).value*
else
Check next edge on current ***ROUTE***
Select the best edge (i^*,j^*) based on the lowest ***C1*** matrix value

Update the ***C2*** matrix for the insertion position
Calculate $c_2(i^*,u,j^*)$

d_{ou} - the distance between the depot (node o) and node u
t_{ou} - the travel time between the depot (node o) and node u
s_u - the service time at node u
λ - in Solomon combinations of 1 and 2 are used
$F(q_u)$ - the fixed cost of the smallest available vehicle that can service the demand of node u
$c_2(i^*,u,j^*) = \lambda(d_{ou} + t_{ou}) + s_u + F(q_u) - c_1^{best}(i,u,j)$

Add the $c2(i^*,u,j^*)$ value to the ***C2*** matrix

Sort ***C2*** in ascending order
Find first time-feasible node (u^*), starting at the beginning of ***C2***
While no u^* has been found, and end of ***C2*** has not been reached
Check for time feasibility

Check node u's time feasibility
$a_u = \max\{e_u, a_i + s_i + t_{iu}\}$
if $a_u \leq l_u + L_u^{max}$
Check node j's time feasibility
$a_j^{new} = \max\{a_j, a_u + s_u + t_{uj}\}$
if $a_j^{new} \leq l_j + L_j^{max}$
Check rest of ***ROUTE***'s r nodes for time feasibility
While feasible
$a_r^{new} = \max\{a_r, a_{r-1} + s_{r-1} + t_{r-1,r}\}$
if $a_r^{new} \leq l_r + L_r^{max}$
then feasible
else
infeasible
else
infeasible
else
infeasible

if feasible
Identify applicable node as u^*

```
                else
                    Check next element of C2

        if a unique u* node has been identified
            Insert node u*
                Update UNROUTED customers
                    Remove u* from UNROUTED
                    Remove any other artificial nodes related to u* from UNROUTED
                Update ROUTE
                    Update ROUTE load
                    if new vehicle has been indicated
                        if Q^new > Q
                            Find the smallest available vehicle to service Q^new
                                Update VEHAVAIL
                                    Change the availability status of the current vehicle to available
                                    Change the availability status of the new vehicle to unavailable
                                    Assign new vehicle to current TOUR
                                    Recalculate VEHAVAIL

                Recalculate ROUTE schedule for nodes
                    Actual start-time at origin (a_o) is the start-time indicated for the current route
                    for each node (i) on the current ROUTE, except the depot at both ends
                        a_i = max {e_i , a_{i-1} + s_{i-1} + t_{i-1,i}}
                        w_i = max {0 , e_{i+1} - (a_i + s_i + t_{i,i+1})}
                    Calculate actual arrival at the depot (n^th node) at the end of the current ROUTE
                        a_n = a_{n-1} + s_{n-1} + t_{n-1,n}}
        else
            Initialize new ROUTE

Expand TOUR
    Determine multi-route feasibility
        Check the actual arrival time at the depot of the previous ROUTE of the current TOUR (a_n)
        if a_n + s_o + 1 hour < l_o^max
            then feasible
        else
            infeasible
    if feasible
        Initialize new ROUTE
    else
        if the last ROUTE of the TOUR has no nodes other than the depot
            Eliminate ROUTE from TOUR
        Initialize new TOUR
```

```
Define ORPHANs
    if UNROUTED is not empty
        Assign all elements in UNROUTED to ORPHANs
        Clear UNROUTED

Report initial solution
    Calculate the OBJective function value for the initial solution
    Report initial solution
        for each TOUR
        Report all TOUR and ROUTE information
```

Appendix C

Output

The output presented in this appendix are representations of the actual text files (*.*txt*) generated by *MATLAB 6.5*. Consider the following excerpt from the output for the $R1$ problem class.

```
 :        :
Tour:    2
Vehicle: v2
Route:   1
Customer  Actual time
   c11        0.00
   c881      13.00
   c511      86.00
   c272     347.00
   c11      358.31
Route:   2
Customer  Actual time
   c11      358.31
   c412     412.00
   :         :
```

The last digit (either 1 or 2) represents the customer's specific time window during which it will be serviced. For example, the second tour is assigned vehicle 2. The first and the last nodes on every route is $c11$. The $c1$ denotes the *depot*, while the last digit, 1, denotes the first time window (and for the depot, the only time window).

The third customer (the fourth node) on the first route of the second tour is $c272$. The $c27$ denotes customer 27, while the last digit, 2, indicates that customer 27 is serviced during its second specified time window, and to be specific, at 347 minutes after the vehicle left the depot.

At the end of each output file, any orphans are indicated. Orphans are defined to be customers that could not be inserted into any routes. The total scheduling distances is also indicated.

C.1 Problem class $R1$

The following output was generated my *MATLAB* for the problem class $R1$: randomly distributed customers with a short scheduling horizon.

```
Tour:     1
Vehicle: v1
Route:    1
Customer  Actual time
   c11        0.00
   c561      17.00
   c681      62.00
   c671     161.00
   c11      172.89
Route:    2
Customer  Actual time
   c11      172.89
   c401     179.00
   c231     267.00
   c482     411.00
   c11      422.44
Route:    3
Customer  Actual time
   c11      422.44
   c912     492.00
   c992     512.00
   c11      523.65
Route:    4
Customer  Actual time
   c11      523.65
   c112     559.00
   c11      570.33

Tour:     2
Vehicle: v2
Route:    1
Customer  Actual time
   c11        0.00
   c881      13.00
   c511      86.00
```

```
   c272     347.00
   c11      358.31
Route:    2
Customer  Actual time
   c11      358.31
   c412     412.00
   c432     422.67
   c712     448.00
   c11      459.20
Route:    3
Customer  Actual time
   c11      459.20
   c842     459.95
   c782     503.00
   c11      514.84
Route:    4
Customer  Actual time
   c11      514.84
   c282     516.67
   c982     531.00
   c11      541.40
Route:    5
Customer  Actual time
   c11      541.40
   c92      560.00
   c11      571.91

Tour:     3
Vehicle: v3
Route:    1
Customer  Actual time
   c11        0.00
   c451      60.00
   c121     345.00
   c11      356.91
Route:    2
Customer  Actual time
   c11      356.91
   c802     420.00
   c921     430.67
   c11      441.49
Route:    3
Customer  Actual time
   c11      441.49
```

```
   c32      443.45
   c752     456.00
   c11      467.00
Route:    4
Customer  Actual time
   c11      467.00
   c622     535.00
   c11      546.80

Tour:     4
Vehicle: v4
Route:    1
Customer  Actual time
   c11        0.00
   c861      68.00
   c651     216.00
   c261     240.00
   c11      251.49
Route:    2
Customer  Actual time
   c11      251.49
   c571     291.00
   c812     309.00
   c11      320.27
Route:    3
Customer  Actual time
   c11      320.27
   c182     346.00
   c11      357.87
Route:    4
Customer  Actual time
   c11      357.87
   c792     413.00
   c11      424.87
Route:    5
Customer  Actual time
   c11      424.87
   c502     428.00
   c202     490.00
   c11      501.49
Route:    6
Customer  Actual time
   c11      501.49
   c602     502.02
```

```
   c11     512.55

Tour:     5
Vehicle: v5
Route:    1
Customer  Actual time
   c11       0.00
   c331     10.00
   c211     64.00
   c851     74.38
   c11      85.58
Route:    2
Customer  Actual time
   c11      85.58
   c351    216.00
   c722    232.00
   c11     244.20
Route:    3
Customer  Actual time
   c11     244.20
   c612    269.00
   c732    384.00
   c11     394.96
Route:    4
Customer  Actual time
   c11     394.96
   c972    401.00
   c11     411.95
Route:    5
Customer  Actual time
   c11     411.95
   c662    413.07
   c832    423.62
   c11     435.00
Route:    6
Customer  Actual time
   c11     435.00
   c462    492.00
   c11     502.44
Route:    7
Customer  Actual time
   c11     502.44
   c252    511.00
   c11     521.29
```

```
Tour:    6
Vehicle: v6
Route:   1
Customer  Actual time
   c11        0.00
   c641      42.00
   c191      63.00
   c11       73.91
Route:   2
Customer  Actual time
   c11       73.91
   c311      75.15
   c541     105.00
   c11      116.49
Route:   3
Customer  Actual time
   c11      116.49
   c162     117.64
   c341     188.00
   c11      199.62
Route:   4
Customer  Actual time
   c11      199.62
   c901     206.00
   c11      216.91
Route:   5
Customer  Actual time
   c11      216.91
   c592     334.00
   c11      345.71
Route:   6
Customer  Actual time
   c11      345.71
   c932     378.00
   c11      389.49
Route:   7
Customer  Actual time
   c11      389.49
   c492     426.00
   c11      437.47
Route:   8
Customer  Actual time
   c11      437.47
```

```
   c82      483.00
   c11      493.84

Tour:     7
Vehicle: v7
Route:    1
Customer  Actual time
   c11        0.00
   c951     170.00
   c11      181.22
Route:    2
Customer  Actual time
   c11      181.22
   c372     278.00
   c11      289.42
Route:    3
Customer  Actual time
   c11      289.42
   c892     354.00
   c11      364.85
Route:    4
Customer  Actual time
   c11      364.85
   c301     365.78
   c442     430.00
   c11      441.44

Tour:     8
Vehicle: v8
Route:    1
Customer  Actual time
   c11        0.00
   c361     114.00
   c11      124.76
Route:    2
Customer  Actual time
   c11      124.76
   c321     190.00
   c11      201.04
Route:    3
Customer  Actual time
   c11      201.04
   c152     287.00
   c11      297.91
```

```
Route:    4
Customer  Actual time
   c11      297.91
   c702     313.00
   c11      323.67
Route:    5
Customer  Actual time
   c11      323.67
   c422     397.00
   c11      407.82

Tour:     9
Vehicle: v9
Route:    1
Customer  Actual time
   c11        0.00
   c171     150.00
   c11      162.27
Route:    2
Customer  Actual time
   c11      162.27
   c52      244.00
   c11      254.82
Route:    3
Customer  Actual time
   c11      254.82
   c962     271.00
   c72      285.00
   c11      297.29
Route:    4
Customer  Actual time
   c11      297.29
   c392     325.00
   c11      336.13
Route:    5
Customer  Actual time
   c11      336.13
   c242     347.00
   c11      358.56

Tour:     10
Vehicle: v10
Route:    1
Customer  Actual time
```

```
      c11      0.00
      c1002   225.00
      c11     235.84
Route:    2
Customer  Actual time
      c11     235.84
      c772    237.65
      c11     249.47
Route:    3
Customer  Actual time
      c11     249.47
      c632    268.00
      c11     279.31

Tour:     11
Vehicle: v11
Route:    1
Customer  Actual time
      c11       0.00
      c942    182.00
      c11     193.36
Route:    2
Customer  Actual time
      c11     193.36
      c522    219.00
      c11     230.29

Tour:     12
Vehicle: v12
Route:    1
Customer  Actual time
      c11       0.00
      c472    229.00
      c11     239.69

Tour:     13
Vehicle: v13
Route:    1
Customer  Actual time
      c11       0.00
      c142    189.00
      c11     200.31
Route:    2
Customer  Actual time
```

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

```
    c11      200.31
    c132     209.00
    c11      220.31

Tour:     14
Vehicle:  v14

Tour:     15
Vehicle:  v15

Tour:     16
Vehicle:  v16

Tour:     17
Vehicle:  v17

Tour:     18
Vehicle:  v18

Tour:     19
Vehicle:  v19

Tour:     20
Vehicle:  v20

Tour:     21
Vehicle:  v21

Tour:     22
Vehicle:  v22

Tour:     23
Vehicle:  v23

Tour:     24
Vehicle:  v24

Tour:     25
Vehicle:  v25

Tour:     26
Vehicle:  v26

Tour:     27
```

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

```
Vehicle: v27

Tour:    28
Vehicle: v28

Tour:    29
Vehicle: v29

Tour:    30
Vehicle: v30

Tour:    31
Vehicle: v31

Tour:    32
Vehicle: v32

Tour:    33
Vehicle: v33

Tour:    34
Vehicle: v34

Tour:    35
Vehicle: v35

Tour:    36
Vehicle: v36

Tour:    37
Vehicle: v37

Tour:    38
Vehicle: v38

Tour:    39
Vehicle: v39

Tour:    40
Vehicle: v40

Tour:    41
Vehicle: v41
```

```
Tour:    42
Vehicle: v42

Tour:    43
Vehicle: v43

Tour:    44
Vehicle: v44

Tour:    45
Vehicle: v45

Tour:    46
Vehicle: v46

Tour:    47
Vehicle: v47

Tour:    48
Vehicle: v48

Tour:    49
Vehicle: v49

Tour:    50
Vehicle: v50

Tour:    51
Vehicle: v51

Tour:    52
Vehicle: v52

Tour:    53
Vehicle: v53

Tour:    54
Vehicle: v54

Tour:    55
Vehicle: v55

Tour:    56
Vehicle: v56
```

```
Tour:    57
Vehicle: v57

Tour:    58
Vehicle: v58

Tour:    59
Vehicle: v59

Tour:    60
Vehicle: v60

Tour:    61
Vehicle: v61

Tour:    62
Vehicle: v62

Tour:    63
Vehicle: v63

Tour:    64
Vehicle: v64

Tour:    65
Vehicle: v65

Tour:    66
Vehicle: v66

Tour:    67
Vehicle: v67

Tour:    68
Vehicle: v68

Tour:    69
Vehicle: v69

Tour:    70
Vehicle: v70

Tour:    71
```

```
Vehicle: v71
Route:    1
Customer  Actual time
   c11      0.00
   c1012    411.00
   c11      422.60
Route:    2
Customer  Actual time
   c11      422.60
   c532     495.00
   c11      505.67

Tour:     72
Vehicle: v72
Route:    1
Customer  Actual time
   c11        0.00
   c872     410.00
   c11      422.36
Route:    2
Customer  Actual time
   c11      422.36
   c292     474.00
   c11      485.38
Route:    3
Customer  Actual time
   c11      485.38
   c62      495.00
   c11      506.71

Tour:    73
Vehicle: v73
Route:    1
Customer  Actual time
   c11      0.00
   c822     309.00
   c11      319.36
Route:    2
Customer  Actual time
   c11      319.36
   c552     384.00
   c11      395.02

Tour:     74
```

```
Vehicle: v74
Route:    1
Customer  Actual time
   c11        0.00
   c742     104.00
   c11      114.44
Route:    2
Customer  Actual time
   c11      114.44
   c692     154.00
   c11      165.47
Route:    3
Customer  Actual time
   c11      165.47
   c382     346.00
   c11      356.65
Route:    4
Customer  Actual time
   c11      356.65
   c102     385.00
   c11      395.40

Tour:     75
Vehicle: v75
Route:    1
Customer  Actual time
   c11        0.00
   c582      76.00
   c11       87.11
Route:    2
Customer  Actual time
   c11       87.11
   c222     263.00
   c11      274.38
Route:    3
Customer  Actual time
   c11      274.38
   c42      336.00
   c11      347.78
Route:    4
Customer  Actual time
   c11      347.78
   c22      366.00
   c11      376.80
```

```
Orphans: none

Total distance: 11260
```

C.2 Problem class $R2$

The following output was generated my *MATLAB* for the problem class $R2$: randomly distributed customers with a long scheduling horizon.

```
Tour:     1
Vehicle: v1
Route:    1
Customer  Actual time
   c11        0.00
   c821      25.00
   c651      37.89
   c571      80.00
   c321      91.02
   c611     102.25
   c691     113.27
   c681     124.73
   c191     134.89
   c91      145.62
   c201     216.00
   c481     240.00
   c521     411.00
   c601     891.00
   c711     903.53
   c501     1226.00
   c861     1313.00
   c732     2254.00
   c11      2265.07

Tour:     2
Vehicle: v2
Route:    1
Customer  Actual time
   c11        0.00
   c841      43.00
   c241      54.65
   c421      64.82
   c51      138.00
```

```
Orphans: none

Total distance: 11260
```

C.2 Problem class $R2$

The following output was generated my *MATLAB* for the problem class $R2$: randomly distributed customers with a long scheduling horizon.

```
Tour:     1
Vehicle: v1
Route:    1
Customer  Actual time
   c11         0.00
   c821       25.00
   c651       37.89
   c571       80.00
   c321       91.02
   c611      102.25
   c691      113.27
   c681      124.73
   c191      134.89
   c91       145.62
   c201      216.00
   c481      240.00
   c521      411.00
   c601      891.00
   c711      903.53
   c501      1226.00
   c861      1313.00
   c732      2254.00
   c11       2265.07

Tour:     2
Vehicle: v2
Route:    1
Customer  Actual time
   c11         0.00
   c841       43.00
   c241       54.65
   c421       64.82
   c51       138.00
```

c351	148.31
c751	158.58
c631	168.95
c171	180.35
c781	192.44
c461	203.89
c1011	234.00
c981	245.25
c341	255.95
c531	266.38
c951	407.00
c41	558.00
c831	1482.00
c642	2204.00
c11	2215.02

Tour: 3
Vehicle: v3
Route: 1

Customer	Actual time
c11	0.00
c741	274.00
c401	285.93
c801	298.05
c371	309.56
c921	365.00
c621	377.44
c181	389.55
c271	400.24
c961	411.56
c361	424.93
c721	438.00
c851	476.00
c511	514.00
c221	524.84
c471	765.00
c121	866.00
c151	952.00
c261	1195.00
c882	2152.00
c11	2162.69

Tour: 4
Vehicle: v4

```
Route:    1
Customer  Actual time
   c11       0.00
   c441    129.00
   c931    211.00
   c661    295.00
   c71     378.00
   c381    390.87
   c491    402.62
   c891    440.00
   c331    575.00
   c291    588.51
   c411    602.38
   c161    631.00
   c451    720.00
   c391    859.00
   c281    900.00
   c141   1009.00
   c11    1020.89
Route:    2
Customer  Actual time
   c11    1020.89
   c671   1022.42
   c431   1035.18
   c251   1047.15
   c761   1058.44
   c61    1102.00
   c971   1112.96
   c591   1124.09
   c771   1136.75
   c231   1149.56
   c991   1160.71
   c101   1173.76
   c582   1186.09
   c811   1196.95
   c901   1207.91
   c311   1219.07
   c81    1231.36
   c561   1428.00
   c32    1552.00
   c11    1563.16
Route:    3
Customer  Actual time
   c11    1563.16
```

```
   c212     1615.00
   c131     1625.55
   c111     1636.85
   c551     1691.00
   c702     1797.00
   c541     1807.91
   c912     1818.78
   c1002    1938.00
   c22      1973.00
   c872     1985.36
   c11      1995.89

Tour:     5
Vehicle: v5
Route:    1
Customer  Actual time
   c11        0.00
   c791     656.00
   c301     1030.00
   c942     1414.00
   c11      1426.11

Orphans: none

Total distance: 8722
```

C.3 Problem class $C1$

The following output was generated my *MATLAB* for the problem class $C1$: clustered customers with a short scheduling horizon.

```
Tour:     1
Vehicle: v1
Route:    1
Customer  Actual time
   c11        0.00
   c141      15.00
   c791     105.16
   c721     196.13
   c391     287.38
   c551     378.27
   c231     469.25
   c11      561.02
```

Route: 2

Customer	Actual time
c11	561.02
c801	595.00
c1011	686.33
c681	777.25
c181	867.35
c402	957.42
c82	1049.42
c11	1141.38

Tour: 2
Vehicle: v2
Route: 1

Customer	Actual time
c11	0.00
c221	72.00
c241	162.16
c511	252.67
c531	343.07
c132	435.24
c11	526.33

Route: 2

Customer	Actual time
c11	526.33
c291	527.31
c152	617.53
c751	723.00
c881	813.76
c162	904.31
c542	996.04
c992	1087.42
c11	1179.67

Tour: 3
Vehicle: v3
Route: 1

Customer	Actual time
c11	0.00
c631	23.00
c451	113.11
c201	203.44
c811	294.58
c322	384.62

```
   c261     474.67
   c11      565.49
Route:    2
Customer   Actual time
   c11      565.49
   c712     577.00
   c522     668.13
   c662     764.00
   c372     856.76
   c842     976.00
   c11     1067.45
Route:    3
Customer   Actual time
   c11     1067.45
   c642    1079.00
   c572    1169.15
   c11     1260.75

Tour:     4
Vehicle: v4
Route:    1
Customer   Actual time
   c11        0.00
   c461      19.00
   c911     109.95
   c111     219.00
   c441     314.00
   c471     405.15
   c921     496.65
   c11      587.87
Route:    2
Customer   Actual time
   c11      587.87
   c381     602.00
   c821     692.07
   c892     855.00
   c501     945.45
   c11     1037.36

Tour:     5
Vehicle: v5
Route:    1
Customer   Actual time
   c11        0.00
```

```
    c741     35.00
    c671    125.38
    c481    215.42
    c31     306.58
    c871    398.91
    c11     491.36
Route:    2
Customer  Actual time
    c11     491.36
    c71     492.25
    c342    583.51
    c101    674.51
    c22     764.55
    c1002   854.58
    c932    947.04
    c11     1039.00

Tour:     6
Vehicle: v6
Route:    1
Customer  Actual time
    c11       0.00
    c211     33.00
    c421    123.15
    c861    213.20
    c251    311.00
    c971    401.71
    c691    492.11
    c11     582.78
Route:    2
Customer  Actual time
    c11     582.78
    c772    583.33
    c412    704.00
    c312    794.49
    c431    885.36
    c11     977.02
Route:    3
Customer  Actual time
    c11     977.02
    c122    979.00
    c11     1069.91

Tour:     7
```

```
Vehicle: v7
Route:   1
Customer  Actual time
   c11        0.00
   c941      51.00
   c171     141.80
   c192     232.51
   c591     322.91
   c362     413.62
   c982     505.33
   c782     650.00
   c11      740.80
Route:   2
Customer  Actual time
   c11      740.80
   c652     741.29
   c352     833.29
   c962     923.33
   c11      1014.84

Tour:    8
Vehicle: v8
Route:   1
Customer  Actual time
   c11        0.00
   c611      77.00
   c831     167.13
   c301     285.00
   c851     378.82
   c731     561.00
   c52      651.04
   c11      743.00
Route:   2
Customer  Actual time
   c11      743.00
   c582     744.29
   c902     895.00
   c602     985.13
   c11      1076.35

Tour:    9
Vehicle: v9
Route:   1
Customer  Actual time
```

```
   c11       0.00
   c331     84.00
   c271    201.00
   c761    291.89
   c701    508.00
   c622    599.44
   c11     689.87
Route:   2
Customer  Actual time
   c11     689.87
   c62     754.00
   c92     950.00
   c11     1042.35

Tour:    10
Vehicle: v10
Route:   1
Customer  Actual time
   c11       0.00
   c491     70.00
   c281    178.00
   c951    517.00
   c11     609.44

Tour:    11
Vehicle: v11
Route:   1
Customer  Actual time
   c11       0.00
   c561     79.00
   c41     169.87
   c11     262.18

Orphans: none

Total distance: 7330
```

C.4 Problem class $C2$

The following output was generated my *MATLAB* for the problem class $C2$: clustered customers with a long scheduling horizon.

```
Tour:    1
```

```
Vehicle: v1
Route:   1
Customer  Actual time
   c11       0.00
   c791     11.00
   c891    101.31
   c381    191.91
   c411    282.95
   c991    373.42
   c741    465.58
   c391    557.40
   c671    648.07
   c191    739.51
   c271    829.64
   c171    921.25
   c91     1012.95
   c41     1104.56
   c292    1195.44
   c321    1287.13
   c701    1377.56
   c481    1469.38
   c761    1560.84
   c442    1651.24
   c542    1742.60
   c981    1883.00
   c772    3202.00
   c11     3293.18
Route:   2
Customer  Actual time
   c11     3293.18
   c62     3295.20
   c692    3387.84
   c11     3478.45

Tour:    2
Vehicle: v2
Route:   1
Customer  Actual time
   c11       0.00
   c51      57.00
   c451    147.73
   c721    238.42
   c111    330.42
   c201    422.13
```

```
   c421    513.07
   c841    604.33
   c301    694.73
   c31     784.91
   c831    877.18
   c461    968.98
   c162    1061.24
   c341    1153.42
   c492    1245.47
   c901    1336.65
   c261    1427.89
   c911    1517.91
   c942    1610.05
   c211    1702.51
   c811    1919.00
   c152    3298.00
   c11     3389.13

Tour:     3
Vehicle: v3
Route:    1
Customer  Actual time
   c11       0.00
   c371      7.00
   c681     97.62
   c331    188.13
   c431    279.56
   c641    370.80
   c1011   462.31
   c951    553.58
   c1001   643.69
   c711    733.75
   c361    824.71
   c471    915.82
   c881    1006.95
   c611    1098.45
   c141    1189.35
   c131    1279.40
   c231    1371.65
   c931    1462.67
   c921    1554.60
   c511    1646.42
   c972    1736.95
   c251    1920.00
```

```
   c11      2010.31
Route:    2
Customer  Actual time
   c11      2010.31
   c752     2012.11
   c501     2120.00
   c241     2211.00
   c622     2301.75
   c181     2392.45
   c562     2483.33
   c862     2573.49
   c82      2664.38
   c572     2756.27
   c802     2849.44
   c22      2941.40
   c531     3032.42
   c962     3123.78
   c352     3214.82
   c11      3306.00

Tour:     4
Vehicle: v4
Route:    1
Customer  Actual time
   c11        0.00
   c651      78.00
   c71      273.00
   c401     365.25
   c601     457.27
   c871     547.44
   c661     640.05
   c631     731.25
   c581     933.00
   c781     1024.35
   c311     1117.00
   c852     1421.00
   c281     1516.00
   c731     1608.11
   c521     1699.36
   c101     1817.00
   c821     2094.00
   c221     2211.00
   c591     2304.00
   c551     2395.33
```

```
   c121    2910.00
   c11     3000.58

Orphans: none

Total distance: 7240
```

C.5 Problem class $RC1$

The following output was generated my *MATLAB* for the problem class $RC1$: both randomly distributed, and clustered customers with a short scheduling horizon.

```
Tour:     1
Vehicle: v1
Route:    1
Customer  Actual time
   c11       0.00
   c441     79.00
   c492    524.00
   c11     534.87
Route:    2
Customer  Actual time
   c11     534.87
   c762    549.00
   c11     560.84

Tour:     2
Vehicle: v2
Route:    1
Customer  Actual time
   c11       0.00
   c251     52.00
   c961     63.04
   c372    550.00
   c11     561.07

Tour:     3
Vehicle: v3
Route:    1
Customer  Actual time
   c11       0.00
   c701     53.00
```

```
    c591      172.00
    c982      446.00
    c11       457.00
Route:     2
Customer   Actual time
    c11       457.00
    c402      473.00
    c682      501.00
    c11       511.29

Tour:      4
Vehicle:  v4
Route:     1
Customer   Actual time
    c11         0.00
    c431       10.00
    c471      138.00
    c212      438.00
    c11       449.53
Route:     2
Customer   Actual time
    c11       449.53
    c652      482.00
    c842      498.00
    c11       509.84

Tour:      5
Vehicle:  v5
Route:     1
Customer   Actual time
    c11         0.00
    c711      101.00
    c201      114.20
    c312      485.00
    c11       495.67

Tour:      6
Vehicle:  v6
Route:     1
Customer   Actual time
    c11         0.00
    c571       41.00
    c862      420.00
    c11       431.69
```

Tour: 7
Vehicle: v7
Route: 1

Customer	Actual time
c11	0.00
c451	206.00
c912	230.00
c11	241.29

Route: 2

Customer	Actual time
c11	241.29
c512	257.00
c172	299.00
c422	416.00
c11	426.85

Tour: 8
Vehicle: v8
Route: 1

Customer	Actual time
c11	0.00
c932	222.00
c11	232.36

Route: 2

Customer	Actual time
c11	232.36
c722	234.00
c11	245.60

Route: 3

Customer	Actual time
c11	245.60
c952	258.00
c462	268.45
c242	291.00
c11	302.73

Route: 4

Customer	Actual time
c11	302.73
c832	356.00
c582	411.00
c11	421.73

Tour: 9

```
Vehicle: v9
Route:   1
Customer  Actual time
   c11       0.00
   c851     51.00
   c121    156.00
   c882    232.00
   c11     243.13
Route:   2
Customer  Actual time
   c11     243.13
   c382    261.00
   c222    287.00
   c342    334.00
   c11     345.49
Route:   3
Customer  Actual time
   c11     345.49
   c742    347.11
   c362    357.75
   c972    372.00
   c11     382.91
Route:   4
Customer  Actual time
   c11     382.91
   c22     411.00
   c612    421.05
   c11     432.49

Tour:    10
Vehicle: v10
Route:   1
Customer  Actual time
   c11       0.00
   c781     72.00
   c91      97.00
   c142    337.00
   c11     348.56
Route:   2
Customer  Actual time
   c11     348.56
   c182    349.58
   c672    372.00
   c11     382.78
```

```
Route:    3
Customer  Actual time
   c11     382.78
   c352    383.71
   c32     396.00
   c11     407.13

Tour:     11
Vehicle: v11
Route:    1
Customer  Actual time
   c11       0.00
   c641     26.00
   c261    117.00
   c941    376.00
   c11     387.60
Route:    2
Customer  Actual time
   c11     387.60
   c772    391.00
   c11     401.95

Tour:     12
Vehicle: v12
Route:    1
Customer  Actual time
   c11       0.00
   c661     65.00
   c552    374.00
   c11     384.44

Tour:     13
Vehicle: v13
Route:    1
Customer  Actual time
   c11       0.00
   c81      57.00
   c811     90.00
   c922    263.00
   c11     274.87
Route:    2
Customer  Actual time
   c11     274.87
   c802    281.00
```

```
   c822    303.00
   c792    314.80
   c11     325.35

Tour:    14
Vehicle: v14
Route:   1
Customer  Actual time
   c11       0.00
   c531     42.00
   c321     52.09
   c752    318.00
   c11     329.98

Tour:    15
Vehicle: v15
Route:   1
Customer  Actual time
   c11       0.00
   c871    161.00
   c562    283.00
   c11     294.25
Route:   2
Customer  Actual time
   c11     294.25
   c272    299.00
   c11     310.56

Tour:    16
Vehicle: v16
Route:   1
Customer  Actual time
   c11       0.00
   c41      83.00
   c162    310.00
   c11     321.24

Tour:    17
Vehicle: v17
Route:   1
Customer  Actual time
   c11       0.00
   c291     15.00
   c151     68.00
```

```
   c72      202.00
   c11      212.45

Tour:     18
Vehicle: v18
Route:    1
Customer  Actual time
   c11        0.00
   c51        9.00
   c501      51.00
   c11       61.49
Route:    2
Customer  Actual time
   c11       61.49
   c412      70.00
   c232     192.00
   c11      203.04

Tour:     19
Vehicle: v19
Route:    1
Customer  Actual time
   c11        0.00
   c541      15.00
   c391     110.00
   c11      121.78
Route:    2
Customer  Actual time
   c11      121.78
   c61      123.33
   c301     135.93
   c112     190.00
   c11      201.98

Tour:     20
Vehicle: v20
Route:    1
Customer  Actual time
   c11        0.00
   c621     101.00
   c992     176.00
   c11      186.82

Tour:     21
```

```
Vehicle: v21
Route:   1
Customer  Actual time
   c11       0.00
   c1011    32.00
   c902    164.00
   c11     174.76

Tour:    22
Vehicle: v22
Route:   1
Customer  Actual time
   c11       0.00
   c481     30.00
   c631     40.95
   c11      52.51
Route:   2
Customer  Actual time
   c11      52.51
   c601     72.00
   c1002   104.00
   c11     116.09
Route:   3
Customer  Actual time
   c11     116.09
   c892    117.40
   c11     128.71
Route:   4
Customer  Actual time
   c11     128.71
   c282    144.00
   c11     155.47

Tour:    23
Vehicle: v23
Route:   1
Customer  Actual time
   c11       0.00
   c131     58.00
   c332    121.00
   c11     132.80
Route:   2
Customer  Actual time
   c11     132.80
```

```
    c732      135.00
    c11       145.76
Route:    3
Customer   Actual time
    c11       145.76
    c102      153.00
    c11       163.93

Tour:     24
Vehicle: v24
Route:    1
Customer   Actual time
    c11         0.00
    c521      127.00
    c11       138.71

Tour:     25
Vehicle: v25
Route:    1
Customer   Actual time
    c11         0.00
    c691       70.00
    c11        81.33
    c11        81.33

Tour:     26
Vehicle: v26
Route:    1
Customer   Actual time
    c11         0.00
    c191       82.00
    c11        93.29
    c11        93.29

Orphans: none

Total distance: 8706
```

C.6 Problem class $RC2$

The following output was generated my *MATLAB* for the problem class $RC1$: both randomly distributed, and clustered customers with a long scheduling horizon.

```
Tour:     1
Vehicle: v1
Route:    1
Customer  Actual time
   c11       0.00
   c411     26.00
   c431     38.05
   c161    136.00
   c961    146.64
   c251    209.00
   c372   2255.00
   c11    2266.07

Tour:     2
Vehicle: v2
Route:    1
Customer  Actual time
   c11       0.00
   c51       9.00
   c851    144.00
   c321    156.04
   c701    210.00
   c91     387.00
   c141    588.00
   c762   2194.00
   c11    2205.84

Tour:     3
Vehicle: v3
Route:    1
Customer  Actual time
   c11       0.00
   c481     61.00
   c631     71.95
   c291     83.15
   c801    180.00
   c441    314.00
   c492   2100.00
   c11    2110.87

Tour:     4
Vehicle: v4
```

```
Route:    1
Customer  Actual time
   c11       0.00
   c891    212.00
   c661    223.60
   c721    432.00
   c591    689.00
   c982   1783.00
   c11    1794.00
Route:    2
Customer  Actual time
   c11    1794.00
   c212   1795.53
   c652   1926.00
   c402   1936.93
   c682   2004.00
   c472   2015.64
   c11    2027.13

Tour:     5
Vehicle: v5
Route:    1
Customer  Actual time
   c11       0.00
   c931     44.00
   c81     229.00
   c911    383.00
   c201    420.00
   c711    433.20
   c451    825.00
   c842   1992.00
   c11    2003.84

Tour:     6
Vehicle: v6
Route:    1
Customer  Actual time
   c11       0.00
   c551     36.00
   c181    587.00
   c312   1940.00
   c11    1950.67

Tour:     7
```

```
Vehicle: v7
Route:   1
Customer  Actual time
   c11       0.00
   c1011    69.00
   c831     79.20
   c531    108.00
   c131    231.00
   c121    623.00
   c882    927.00
   c11     938.13
Route:   2
Customer  Actual time
   c11     938.13
   c512    1028.00
   c952    1038.25
   c462    1048.71
   c382    1059.53
   c421    1070.38
   c242    1165.00
   c342    1336.00
   c11     1347.49
Route:   3
Customer  Actual time
   c11     1347.49
   c742    1349.11
   c752    1362.71
   c972    1488.00
   c941    1503.00
   c352    1514.36
   c362    1525.44
   c11     1537.44
Route:   4
Customer  Actual time
   c11     1537.44
   c672    1538.22
   c612    1605.00
   c582    1644.00
   c22     1656.07
   c862    1682.00
   c11     1693.69

Tour:    8
Vehicle: v8
```

```
Route:    1
Customer  Actual time
    c11        0.00
    c541      15.00
    c31      197.00
    c641     207.98
    c61      324.00
    c772    1564.00
    c11     1574.95

Tour:     9
Vehicle:  v9
Route:    1
Customer  Actual time
    c11        0.00
    c731     211.00
    c501     221.89
    c811     358.00
    c521     507.00
    c922    1054.00
    c11     1065.87
Route:    2
Customer  Actual time
    c11     1065.87
    c562    1132.00
    c792    1204.00
    c222    1215.47
    c172    1226.49
    c822    1237.11
    c11     1248.36

Tour:     10
Vehicle:  v10
Route:    1
Customer  Actual time
    c11        0.00
    c781     230.00
    c71      241.51
    c191     329.00
    c601     340.60
    c261     470.00
    c272    1196.00
    c11     1207.56
```

```
Tour:     11
Vehicle: v11
Route:    1
Customer  Actual time
   c11        0.00
   c571     148.00
   c101     180.00
   c1001    209.00
   c331     328.00
   c391     439.00
   c11      450.78
Route:    2
Customer  Actual time
   c11      450.78
   c991     451.60
   c621     462.56
   c41      473.69
   c282     576.00
   c232     768.00
   c11      779.04

Tour:     12
Vehicle: v12
Route:    1
Customer  Actual time
   c11        0.00
   c151     270.00
   c901     417.00
   c691     428.65
   c301     513.00
   c871     646.00
   c11      657.56
Route:    2
Customer  Actual time
   c11      657.56
   c112     760.00
   c11      771.98

Orphans: none

Total distance: 7748
```

Bibliography

[1] *A century of transport: a record of achievement of the Ministry of Transport of the Union of South Africa.* Da Gama Publications, Cape Town, 1960.

[2] *The American Heritage Dicionary of the English Languaue.* Houghton Mifflin Company, 4^{th} edition, 2000.

[3] D. Avison and G. Fitzgerald. *Information systems development: Methodologies, techniques, and tools.* McGraw-Hill, Berkshire, UK, 3^{rd} edition, 2003.

[4] B.M. Baker and M.A. Ayechew. A genetic algorithm for the vehicle routing problem. *Computers and Operations Research*, 30(5):787–800, 2003.

[5] D. Banister. *Transport and urban development.* E & FN Spon, London, 1^{st} edition, 1995.

[6] G. Baseman. The cars that ate London, Paris, Brussels, Amsterdam, Rome, Madrid, Vienna, Athens .. *TIME Europe Magazine*, 161(8):37–40, February 2003.

[7] M. Baybars and M. Browne. Developments in urban distribution in London. In E. Taniguchi and R.G. Thompson, editors, *City Logistics III*, pages 303–317. Institute for City Logistics, Institute of Systems Science Research, June 2003.

[8] Z. Botha. First batch of taxi permits issued. *Martin Creamer's Engineering News*, 23(17):10.

[9] J. Brandão and A. Mercer. A tabu search algorithm for the multi-trip vehicle routing and scheduling problem. *European Journal of Operational Research*, 100:180–191, 1997.

[10] O. Bräysy and M. Gendreau. Tabu search heuristics for the vehicle routing problem with time windows. Report stf42 a01022, SINTEF Applied Mathematics, Research Council of Norway, December 2001.

[11] S.E. Butt and D.M. Ryan. An optimal solution procedure for the multiple tour maximum collection problem using column generation. *Computers and Operations Research*, 26:427–441, 1999.

[12] G. Cambridge. Taxi re-capitalisation project. Technical report, Department of Trade and Industry (DTI), October 2000.

[13] N. Christofides, A. Mingozzi, and P. Toth. *The Vehicle Routing Problem*. John Wiley & Sons, New York, 1979.

[14] G. Clarke and J.W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12:568–581, 1964.

[15] Government Communication and Information System (GCIS). *South Africa Yearbook 2002/03*. South Africa Oficial Yearbook. Government Communication and Information System (GCIS) and STA Publications, 9^{th} edition, October 2002.

[16] E. De Boer. *Transport sociology: social aspects of transport planning*. Pergamon Press, New York, 1^{st} edition, 1986.

[17] J.A. De Bruijn and E.F. tenHeuvelhof. *Managing complex networks: strategies for the public sector*. Thousand Oaks, London, 1997.

[18] B. de Saint-Laurent. Overview of urban transport in South Africa: Lessons from Europe. In Peter Freeman and Christian Jamet, editors, *Urban transport policy — a sustainable development tool*, Rotterdam, 1998. CODATU, A.A. Balkema.

[19] W. Dullaert, G.K. Janssens, K. Sörensen, and B. Vernimmen. New heuristics for the fleet size and mix vehicle routing problem with time windows. In *9th World Conference on Transport Research, July 22–27, 2001*, COEX Convention Center, Seoul, 2001.

[20] M. Gendreau, G. Laporte, C. Musaraganyi, and É.D. Taillard. A tabu search heuristic for the heterogeneous fleet vehicle routing problem. *Computers and Operations Research*, 26:1153–1173, 1999.

[21] F. Glover. A user's guide to tabu search. *Annals of Operations Research*, 41:3–28, 1993.

[22] B. Golden, A. Assad, L. Levy, and F. Gheysens. The fleet size and mix vehicle routing problem. *Computers and Operations Research*, 11(1):49–66, 1984.

[23] J.H. Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT Press, Cambridge, Massachusetts, 1992.

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

[24] J. Homberger. Extended SOLOMON's VRPTW instances. World wide web at *http://www.fernuni-hagen.de/WINF/touren/inhalte/probinst.htm*, September 2003.

[25] T. Ibaraki, S. Imahori, M. Kubo, T. Masuda, T. Uno, and M. Yagiura. Effective local search algorithms for routing and scheduling problems with general time window constraints. *Transportation Science*, Forthcoming.

[26] S. Kirkpatric, C.D. Gelatt, and M.P. Vecchi. Optimisation by simulated annealing. *Science*, 20:671–680, 1983.

[27] G. Laporte. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59:345–358, 1992.

[28] P.A. Leinbach and T. Stansfield. Living up to expectations. *IE Solutions*, 34(11):24–30, November 2002.

[29] J.K. Lenstra and A.H.G. Rinnooy Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11:221–227, 1981.

[30] V.S. Lipman and V.A. Monaghan. Moving South Africa - motivation and progress. In Peter Freeman and Christian Jamet, editors, *Urban transport policy — a sustainable development tool*, Rotterdam, 1998. CODATU, A.A. Balkema.

[31] F.-H. Liu and S.-Y. Shen. The fleet size and mix vehicle routing problem with time windows. *Journal of the Operational Research Society*, 50:721–732, 1999.

[32] F.-H. Liu and S.-Y. Shen. A method for Vehicle Routing Problem with Multiple Vehicle Types and Time Windows. *Proceedings of the National Science Council, Republic of China, ROC(A)*, 23(4):526–536, 1999.

[33] J. Mouton. *How to succeed in your Master's and Doctoral studies: a South African guide and resource book.* Van Schaik, 1^{st} edition, 2001.

[34] A.J. Nothnagel. Overview of the South African national land transport policy. In Peter Freeman and Christian Jamet, editors, *Urban transport policy — a sustainable development tool*, Rotterdam, 1998. CODATU, A.A. Balkema.

[35] Department of Environmental Affairs and Tourism. *White paper on integrated pollution and waste management for South Africa.* Republic of South Africa, 2000.

[36] Department of Transport. *Airports Company Act, Act 44 of 1993.* Government printer, Pretoria, South Africa, 1993.

[37] Department of Transport. *National Land Transport Transition Act, Act 22 of 2000.* Government printer, Pretoria, South Africa, 2000.

[38] SARB Chair of Transportation Engineering. *Transportation in context.* University of Pretoria, 2003.

[39] SARB Chair of Transportation Engineering. *Transportation in society.* University of Pretoria, 2003.

[40] R. Ooishi and E. Taniguchi. Effects and profitability of constructing the new underground freight transport system. In Eiichi Taniguchi and Russell G. Thompson, editors, *City Logistics I.* Institute for City Logistics, Institute of Systems Science Research, 1999.

[41] I.H. Osman. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, 41:147–167, 1995.

[42] H. Otto. 2 die as cement mixer crushes vehicles. *Pretoria News*, page 1, February 7 2003.

[43] R.L. Rardin. *Optimization in Operations Research.* Prentice Hall, Upper Saddle River, New Jersey, 1998.

[44] C.R. (Ed) Reeves, editor. *Modern heuristic techniques for combinatorial problems.* Blackwell Scientific, Oxford, 1^{st}MIT Press edition, 1993.

[45] S. Salhi and G.K. Rand. Incorporating vehicle routing into the vehicle fleet composition problem. *European Journal of Operational Research*, 66:313–330, 1993.

[46] M.M. Solomon. Algorithms for the vehicle routing and scheduling problems with time windows. *Operations Research*, 35(2):254–265, 1987.

[47] M.M. Solomon. VRPTW benchmark problems. World wide web at *http://w.cba.neu.edu/~msolomon/problems.htm*, June 2003.

[48] M.N. Spence. Western Cape Provincial Transport Policy. In Peter Freeman and Christian Jamet, editors, *Urban transport policy — a sustainable development tool*, Rotterdam, 1998. CODATU, A.A. Balkema.

[49] H.A. Taha. *Operations research: an introduction.* Pearson Education, Inc., Upper Saddle River, New Jersey, 7^{th} edition, 2003.

[50] É.D. Taillard. A heuristics column generation method for the heterogeneous fleet VRP. *Operations Research – Recherche opérationnelle*, 33:1–14, 1999.

[51] É.D. Taillard, P. Badeau, M. Gendreau, F. Guertin, and J.Y. Potvin. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science*, 31(2):170–186, May 1997.

[52] É.D. Taillard, L.M. Gambardella, M. Gendreau, and J.Y. Potvin. Adaptive memory programming: A unified view of metaheuristics. *European Journal of Operational Research*, 135(1):1–16, 2001.

[53] É.D. Taillard, G. Laporte, and M. Gendreau. Vehicle routeing with multiple use of vehicles. *Journal of the Operational Research Society*, 47:1065–1070, 1996.

[54] K.C. Tan, L.H. Lee, Q.L. Zhu, and K. Ou. Heuristic methods for vehicle routing problem with time windows. *Artificial Intelligence in Engineering*, 15:281–295, 2001.

[55] E. Taniguchi, R.G.Thompson, and T. Yamada. Modelling city logistics. In Eiichi Taniguchi and Russell G. Thompson, editors, *City Logistics I*. Institute for City Logistics, Institute of Systems Science Research, 1999.

[56] E. Taniguchi, R.G. Thompson, and T. Yamada. Visions for city logistics. In E. Taniguchi and R.G. Thompson, editors, *City Logistics III*, pages 3–17. Institute for City Logistics, Institute for Systems Science Research, June 2003.

[57] E. Taniguchi, R.G. Thompson, T. Yamada, and R. van Duin. *City Logistics: network modelling and intelligent transport systems*. Pergamon, Oxford, UK, 2001.

[58] S.R. Thangiah. *Practical handbook of genetic algorithms: new frontiers*, volume II, chapter Vehicle routing with time windows using genetic algorithms, pages 253–278. CRC Press, 1995.

[59] S.R. Thangiah, I.H. Osman, and T. Sun. Hybrid genetic algorithms, simulated annealing, and tabu search methods for vehicle routing problems with time windows. Technical report ukc/or94/4, Institute of Mathematics and Statistics, University of Kent, UK, 1994.

[60] W.I. Thomas and M. Janowitz. *W.I. Thomas on social organization and social personality: selected papers*. University of Chicago Press, Chicago, 1966.

[61] A. Van Breedam. Comparing descent heuristics and metaheuristics for the vehicle routing problem. *Computers and Operations Research*, 28:289–315, 2001.

[62] J.H.R. van Duin, P.W.G. Bots, and M.J.W van Twist. Improving strategic decision making: dynamic actor network analysis. In *IEEE*

International conference on systems, man, and cybernetics, pages 1013–1017. IEEE, 1999.

[63] A. Villa. Introducing some supply chain management problems. *International Journal of Production Economics*, 73(1):1–4, 2001.

[64] J.L. Whitten and L.D. Bentley. *Systems analysis and design methods.* McGraw-Hill, Boston, Massachusetts, 4^{th} edition, 1998.

[65] W.L. Winston and M. Venkataramanan. *Introduction to mathematical programming*, volume 1 of *Operations Research.* Brooks/Cole - Thomson Learning, Pacific Grove, CA, 4^{th} edition, 2003.