Real-time Spatial Optimization

based on the application in Wood Supply Chain management

Dipl.-Ing.(FH) Johannes Scholz

A thesis submitted in partial fulfillment of
the requirements for the degree of
Doctor of Technical Sciences
submitted to the Graz University of Technology
Faculty of Technical Mathematics and Technical Physics

Advisory Committee: Univ.-Prof. Dr. phil. Norbert Bartelme
Prof. Michael Frank Goodchild, Ph.D.
Ao. Univ.-Prof. DI Dr. Harald Vacik, MAS(GIS)

Graz 2010

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

....

Abstract

Real-time spatial optimization – a combination of Geographical Information Science and Technology and Operations Research – is capable of generating optimized solutions to given spatial problems in real-time. The basic concepts to develop a real-time spatial optimization system are outlined in this thesis.

Geographic Information Science delivers the foundations for acquiring, storing, manipulating, visualizing and analyzing spatial information. In order to develop a system that consists of several independent components the concept of Service Oriented Architectures is applied. This facilitates communication between software systems utilizing standardized services that ensure interoperability. Thus, standards in the field of Geographic Information are inevitable for real-time spatial optimization. By exploiting the ability of mobile devices to determine the own position paired with standardized services Location Based Services are created. They are of interest in order to gather real-time data from mobile devices that are of importance for the optimization process itself.

To optimize a given spatial problem, the universe of discourse has to be modeled accordingly. For the problem addressed in this thesis – Wood Supply Chain management – Graph theory is used. In addition, the problem of Wood Supply Chain management can be represented by a specific mathematical problem class, the Vehicle Routing problem – specifically the Vehicle Routing Problem with Pickup and Delivery and Time Windows. To optimize this problem class, exact and approximate solution techniques exist. Exact algorithms provide optimal solutions and guarantee their optimality, whereas approximate techniques – approximation algorithms or heuristics – do not guarantee that a global optimum is found. Nevertheless, they are capable of handling large problem instances in reasonable time. For optimizing the Wood Supply Chain Adaptive Large Neighborhood Search is selected as appropriate optimization technique – which is an approximate algorithm.

Spatial Decision Support Systems are software systems that are able to support a decision maker in a complex, semi-structured problem. Through the fact that Wood Supply Chain management can be regarded as a semi-structured problem, it is feasible to solve this task with the help of such a system. The decision maker is able to influence the results of a Decision Support System, either by selecting one of the system's recommendations or by the input of preferences and additional criteria.

A combination of all three fields of expertise – Geographic Science and Technology, Operations Research and Spatial Decision Support Systems –

results in a system that is capable of providing decision support for Wood Supply Chain management by real-time spatial optimization. Hence, relevant data are collected in real-time, and optimal routes for each truck are calculated for a specified planning horizon. With the help of such a system, timber – after it is piled up in the forest – is dynamically allocated (and thus dynamically "sold") to saw mills. Subsequently the routes of each vehicle are calculated so that the overall profit is maximized – i.e. timber sales minus haulage costs. For proving the feasibility of this concept a prototype is developed, and two test instances are created. These test instances are optimized using the prototype at hand followed by an evaluation of the obtained results.

Zusammenfassung

Mit räumlicher Optimierung in Echtzeit, die eine Kombination von Geographic Information Science und Technology sowie Operations Research darstellt, ist es möglich räumliche Problemstellungen in Echtzeit zu optimieren. Die grundlegenden Konzepte um ein System zur Echtzeit Optimierung von räumlichen Problemen aufzubauen werden in dieser vorliegenden Arbeit behandelt.

Das Fachgebiet Geographic Information Science liefert die Grundlagen zur Erfassung, Speicherung, Manipulation, Visualisierung und Analyse von geographischer Information. Um ein System zu entwickeln, welches aus mehreren unabhängigen Komponenten besteht, kann das Konzept der Service orientierten Architekturen angewandt werden. Es ermöglicht die Kommunikation zwischen Softwaresystemen mittels standardisierter Services, die Interoperabilität gewährleisten. Deshalb sind Standards – vor allem jene aus dem Gebiet der Geoinformation – wichtig für die Entwicklung eines räumlichen Optimierungssystems. Durch die Kombination von mobilen Endgeräten, wie PDA's oder Smartphones – die vermehrt die Möglichkeiten haben ihre Position festzustellen – sowie standardisierter Services, entstehen ortsbezogene Dienste (Location Based Services). Diese Services sind für die Sammlung von Echtzeitdaten von mobilen Geräten unabdingbar, was für den Optimierungsprozess von großer Bedeutung ist.

Um ein gegebenes räumliches Problem zu optimieren ist es notwendig die zu untersuchende Aufgabenstelllung entsprechend zu modellieren. Im Zuge dieser Arbeit wird die Logistikkette Holz mittels Graphentheorie modelliert. Weiters kann die Logistikkette Holz durch eine spezielle mathematische Problemklasse – dem Vehicle Routing Problem – repräsentiert werden. Im Speziellen handelt es sich um das Vehicle Routing Problem mit Pickup und Delivery und Time Windows. Um diese Problemklasse zu optimieren gibt es exakte und approximative Algorithmen. Exakte Techniken garantieren die Optimalität der gefundenen Lösungen, während approximative Techniken – Approximationsalgorithmen und Heuristiken – keine global beste Lösung garantieren. Trotzdem können die beiden letztgenannten Techniken gut mit großen Probleminstanzen umgehen. Um die Logistikkette Holz zu optimieren wurde Adaptive Large Neighborhood Search als probate Optimierungstechnik ausgewählt – was eine approximative Technik darstellt.

Spatial Decision Support Systeme sind software-basierte Systeme die Entscheidungsträger bei komplexen, semi-strukturierten Problemen unterstützen. Da die Logistikkette Holz als semi-strukturiertes Problem angesehen werden kann, erscheint es sinnvoll solch eine Aufgabe mittels Spatial

Decision Support Systemen zu lösen. Die Entscheidungsträger können das Ergebnis des Decision Support Systems durch die Auswahl einer Empfehlung des Systems oder durch die Eingabe von Präferenzen und weiteren Entscheidungskriterien beeinflussen.

Durch die Kombination der drei Fachgebiete – Geographic Information Science und Technology, Operations Research und Spatial Decision Support Systemen – kann ein System entwickelt werden, das Entscheidungsunterstützung für die Logistikkette Holz durch räumliche Optimierung bietet. Die Daten, die für die Optimierung von Bedeutung sind, werden in Echtzeit gesammelt, und in der Folge werden für einen definierten Planungshorizont optimale Routen für jeden LKW berechnet. Dadurch ist es möglich, Holz – nachdem es im Wald auf einen Polter gestapelt wurde – dynamisch Sägewerken zuzuweisen, wodurch der Verkaufsprozess eigentlich zustandekommt. Die Routen werden so berechnet, dass der globale Profit der Logistikkette maximiert wird – d.h. Erlöse aus dem Holzverkauf minus Transportkosten. Um zu zeigen, dass ein solches System funktioniert werden ein Prototyp, sowie zwei Instanzen von Testdaten erstellt. Diese Testdaten werden mit dem entwickelten System optimiert und in weiterer Folge evaluiert.

Acknowledgements

This research was carried out from January 2007 until December 2009 at the Institute of Geoinformation of Graz University of Technology. I am very thankful that a number of people accompanied and supported me on this way.

First of all I would like to thank Univ.-Prof. Dr. Norbert Bartelme for his endless encouragement, support and patience in answering my questions. Without him I – as non-TUG alumnus – would not have found my way to Graz University of Technology, and neither have finished a thesis like this. I am very thankful for his critical and honest comments on this work, that enhanced the quality significantly. Additionally, he has provided such a positive and creative working climate that made it possible to concentrate on both, interesting research projects and this thesis.

I would also like to thank Prof. Michael Frank Goodchild for his willingness to review this thesis and providing valuable hints on the work. His comments let me rethink the general direction of this work and led to a refinement of important parts of the thesis.

I am also gratefully that Ao. Univ.-Prof. Harald Vacik took over the responsibility to review this dissertation. His detailed comments, that addressed relevant issues, are greatly appreciated.

The Holzcluster Steiermark GmbH – in particular Dipl.-Ing.(FH) Erhard Pretterhofer – provided valuable information on the contemporary Wood Supply Chain and potential improvements. Hence, Erhard Pretterhofer was the essential connection to the forestry sector.

I would also like to thank my colleagues at the Institute of Geoinformation and the Institute of Remote Sensing and Photogrammetry of Graz University of Technology for their fellowship, support and "inspiring" coffee and tea breaks. Additionally, I am happy to have received so much hospitality from colleagues of the Institute of Navigation and Satellite Geodesy – whom I was honored to help out in an interesting research project.

Finally I would like to thank my parents for their support throughout my life, and for giving me a critical attitude on my way. I am also gratefully for the cordiality and support received from my parents in law. A very special thank goes to my beloved wife Susanne, who endlessly supported me on my way, lifted me up when I was down, gave me professional advice whenever necessary, and who was my link to real life.

Contents

List of Figures

List of Algorithms

List of Abbreviations

ALNS	Adaptive Large Neighborhood Search
API	Application Programming Interface
CVRP	Capacitated Vehicle Routing Problem
CRS	Coordinate Reference System
DB	Database
DOP	Dilution of Precision
DBMS	Database Management System
DSS	Decision Support System
CEN	Comité Européen de Normalisation
EPSG	European Petroleum Survey Group
ERD	Entity Relationship Diagram
GI	Geographic Information
GIS	Geographic Information System
GIS&T	Geographic Information Science and Technology
GISc	Geographic Information Science
GIService	Geographic Information Service
GIT	Geographic Information Technology
GPS	Global Positioning System
GML	Geography Markup Language
ISO	International Organization for Standardization
LBS	Location Based Service
MCDA	Multi Criteria Decision Analysis
MCDM	Multi Criteria Decision Making

MIP	Mixed Integer Problem
OGC	Open Geospatial Consortium
OpenLS	Open Location Service
OR	Operations Research
ON	Austrian Standards Institute
OWL	Web Ontology Language
PDA	Personal Digital Assistant
PH	Planning Horizon
RDBMS	Relational Database Management System
RSA	Rolling Schedule Approach
REST	Representational State Transfer
SA	Simulated Annealing
SC	Supply Chain
SCM	Supply Chain Management
SCM-system	Supply Chain Management System
SCP	Supply Chain Planning
SDBMS	Spatial Database Management System
SDI	Spatial Data Infrastructure
SDSS	Spatial Decision Support System
SES	Spatial Expert System
SFS	Simple Features Interface Standard
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SRDBMS	Spatial Relational Database Management System
TSP	Traveling Salesman Problem
UbiCom	Ubiquitous Computing
UI	User Interface

VB	Visual Basic
VRP	Vehicle Routing Problem
VRPB	VRP with Backhauls
VRPPD	VRP with Pickup and Delivery
VRPPDTW	Vehicle Routing Problem with Pickup and Delivery and Time Windows
VRPTW	Vehicle Routing Problem with Time Windows
WFS	Web Feature Service
WFS-T	Web Feature Service-transactional
WMS	Web Mapping Service
WS	Web Service
WSC	Wood Supply Chain
XML	Extensible Markup Language

Chapter 1

Introduction

This chapter comprises preliminary definitions necessary to understand the contents of this thesis. The concept of a Supply Chain is explained, followed by a detailed description of the Wood Supply Chain. Spatial optimization, which is the central methodology used in this dissertation, is introduced before the problems addressed in this thesis are mentioned. Finally, sections on the relevant literature and the organization of the thesis itself complete this chapter.

1.1 Supply Chain Management

In the last decades Supply Chain Management (SCM) emerged as a buzzword in business operations planning. Generally speaking, this concept does assume that stakeholders are not isolated islands in the market. If the market is considered as an ocean and the participating enterprises as islands that get and deliver something from/to the ocean, the market would not work as well as it does. If nobody in the ocean knows anything about the others needs, it would be hard to develop and produce goods that are really demanded. Thus, a great amount of resources is wasted due to the lack of target-oriented production and development.

The definition of the term SCM comprises the linkage between all stakeholders during the lifecycle of a product (Wannenwetsch, 2005). Generally speaking, the Supply Chain (SC) starts with the production of the raw materials and ends at the consumer. Thus, all participating institutions, enterprises or people are part of the SC (see figure 1.1). Based on this definition, also the transportation process itself is part of the SC.

Supply Chain Planning (SCP) enables every company to plan within the SC based on the market situation. Thus several processes have to be included in the planning process, based on Wannenwetsch (2005):

- SC design
- sales and requirements planning

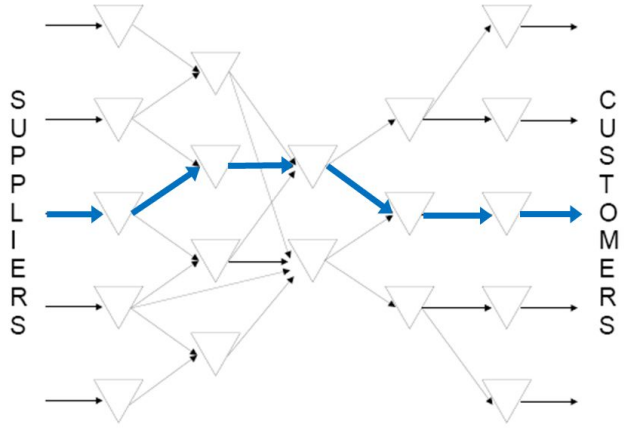


Figure 1.1: An example of a Supply Chain consisting of a number of suppliers and customers (Spitter, 2005). A number of possible paths from supplier to customer exist, of which only one – the optimal (marked in blue) – is chosen, based on a number of criteria.

- production planning
- distribution planning
- available-to-promise: guarantees reliable delivery dates in real time.

To facilitate SCP informatics may be used. The term Supply Chain Management System (SCM-system) comprises all tools that are capable of supporting the processes mentioned above, including all planning, simulation and optimization operations. Moreover the flow of commodities, information and money has to be integrated into these systems. Based on a study by Forrester Research (2001) running a SCM-system can result in an improvement of delivery reliability of 40%. At the same time the delivery time is reduced by 30% and the processing time is cut by 10%.

Generally speaking, a SCM-system is a software system that is capable of supporting business activities in four major areas/phases (see figure 1.2):

- Planning
- Execution
- Coordination
- Cooperation

A SCM-system may support planning activities by facilitating a proper SC design, as well as an exact requirements analysis. A well founded transport

planning is possible due to this preliminary work which may include network design. In the course of the execution phase of the production cycle a SCM-system is capable of supporting controlling, stock-keeping, production and transportation. Through the coordination phase the systems monitor the processes and seek for savings capacities. In case of altering conditions the production process itself may be altered - which will be handled by the system by raising an alert. In case of minor changes in the conditions that affect the process a SCM-system may adjust the performance of e.g. machines in order to guarantee an optimal production process. Cooperation takes place by the people or organizations in the SC. Thus a platform for e.g. e-commerce, registering commodities to transport from A to B, is helpful to integrate more organizations in the SC and to let them cooperate more intensely.

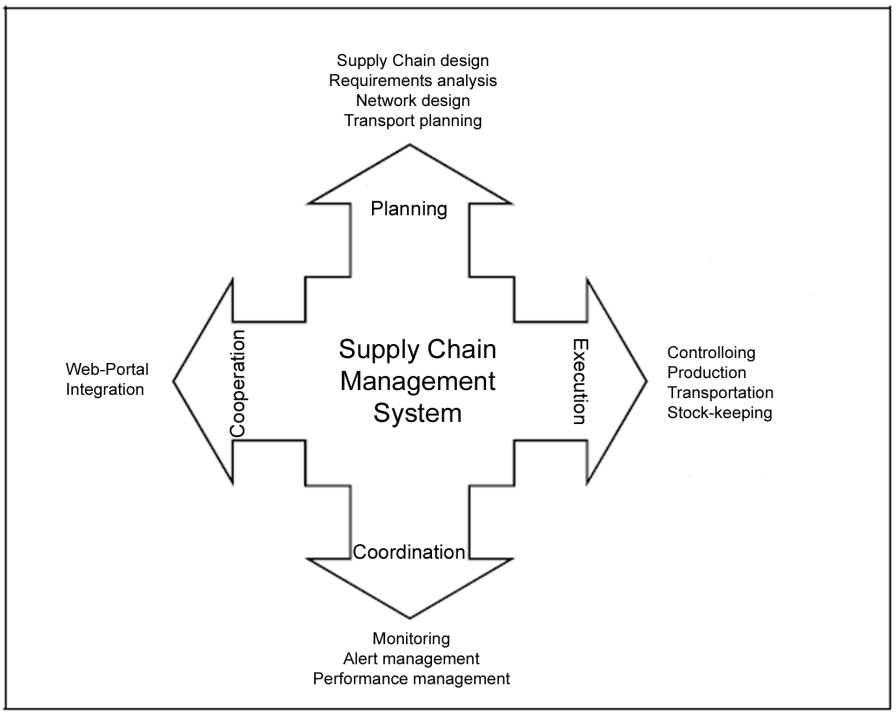


Figure 1.2: Major working areas of a Supply Chain Management system (based on Wannenwetsch (2005)).

Wannenwetsch (2005) states that Location Based Services (LBSs) help to intensify and optimize either the Business to Customer (B2C) as well as the Business to Business (B2B) relations. Typical fields of applications for LBS paired with mobile navigation in the B2B sector are according to Reichenbacher (2004):

- orientation and localization

- navigation
- search services
- identification
- event check

A description of the preceding list can be found in the following paragraph. Orientation and localization represents the position of a mobile device in relation to a reference frame (e.g. coordinate system). Route planning and guidance based on digital spatial data heavily relies on the existence of accurate position information of the mobile device. Search services provide a basic search functionality for spatial objects, i.e. "Where is the next ...?". The ability to get information on unknown spatial objects is of great importance for navigating in unfamiliar areas, and is covered by the identification ability. A number of implemented triggers, either spatial, temporal or attributive can fire events on the mobile device or the service center. Typical examples would be: "Turn left" call - in guidance; truck leaves planned route; truck has serviced all customer requests and is fully loaded; emergency button was hit - security and emergency services.

The so-called *"Mobile Supply Chain Management"* (Wannenwetsch, 2005) - the application of LBS in B2B SCM-systems - takes advantage of the combination of both fields of expertise.

1.2 Wood Supply Chain

The term Wood Supply Chain (WSC) comprises the logistics system from timber to final product that is delivered to a customer. This work will focus on the logistic operations from timber production to the first processing step - usually taking place in a saw or paper mill.

Figure 1.3 shows a brief description of the WSC part that is subject of this dissertation. Saw mills have a demand for timber with a number of special constraints, like quality, quantity, due-date and a price per cubic meter the saw mill is willing to pay. Timber arriving from a forest enterprise must meet the criteria quality and due-date. The quantity may be met by a number of forest enterprises. It can be assumed that forest enterprises produce timber if a demand exists or if they are "forced" to do so due to external conditions, like e.g. catastrophic wind breaks. Produced timber has several attributes: quality, quantity, completion date and a price per cubic meter the forest enterprise wants to receive. Haulage companies transport timber from various forest enterprises to saw mills and have the following

criteria: price per driving unit, capacity constraints. The striking questions concerning the WSC are the following:

- WHAT should be transported? e.g. timber with quality class A, from forest enterprise 1
- WHO should transport? e.g. haulage company 1
- WHEN should it be transported? e.g. picked up tomorrow at 8.30 am
- WHERE TO? e.g. saw mill 1

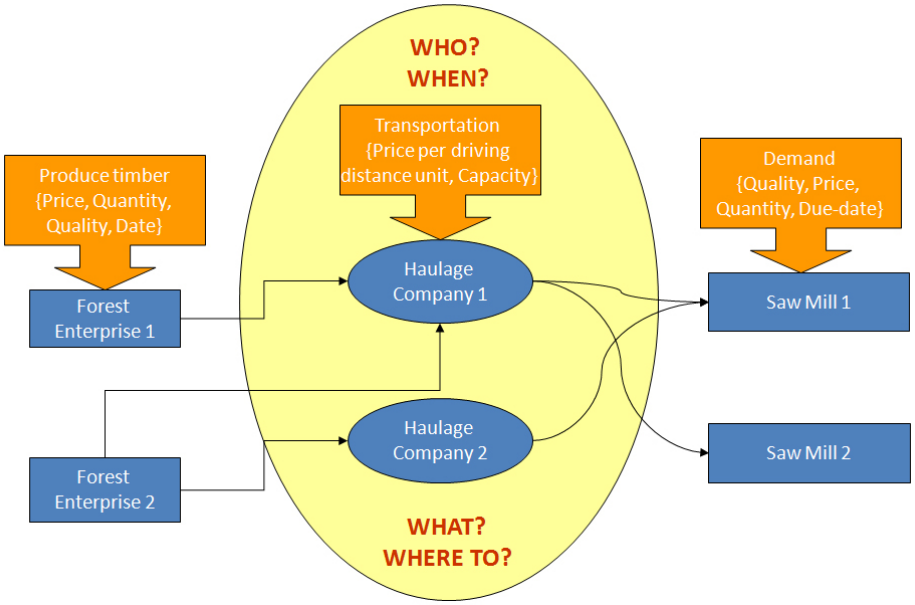


Figure 1.3: Simplified illustration of the Wood Supply Chain.

To ensure an efficient transportation process, relevant information must be passed on throughout the whole process. Due to the great number of participants, and to a great variety of information media and devices within the WSC, media disruptions occur (Gronalt *et al.*, 2005), see figure 1.4. Moreover the dispatchers hardly ever use information systems to maintain an overview of the situation. Thus, the uncoordinated handling of logistic operations is very likely. Concluding, the characteristics of the WSC are as follows:

- Great number of participants
- Bidirectional flow of products and information

- Voluminous and complex information has to be processed
- Spatial in nature
- Problem is not entirely solvable by computers - human knowledge has to be involved in managing WSC operations

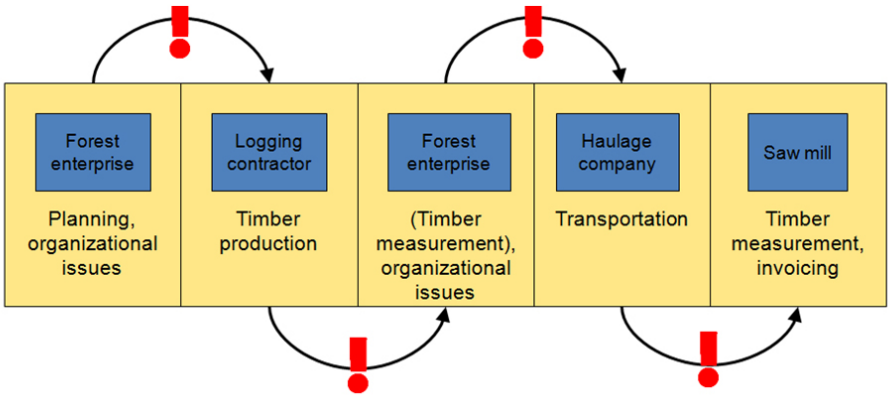


Figure 1.4: Information path and traditional shortcomings - here media disruptions marked with red exclamation marks - of the Wood Supply Chain based on von Bodelschwingh *et al.* (2003).

The actual WSC works as follows: A forest enterprise and a saw mill sign a contract that regulates the amount, due date and quality of timber delivered by the forest enterprise. Usually the timber has to be hauled to the next forest road from where it is transported to the saw mill by a haulage company that both contracting parties agree on. Thus, the destination of the timber is fixed before the timber is even harvested. In addition, trucks are assigned to timber transports without knowledge of surrounding conditions, like other timber that should be transported from nearby locations. The location of timber piles and the information that timber is available to haul has to be given by the forest enterprises. In most cases, the location information consists of an oral description of the position, which is only usable by truck drivers who are familiar with the particular area – although LBSs are introduced in this sector too. Hence, ”foreign” truck drivers are hardly ever able to find the timber to pick up.

To clarify where different components and concepts presented in this thesis will have an impact, two scenarios shall be used, which are outlined below. In both scenarios, a high degree of standardization as well as rich and well-documented geospatial data infrastructure are essential ingredients.

- Scenario 1 (long-term): A new saw mill, a new forest enterprise or new haulage company enters the market. All processes have to be

revised and for this purpose, a multitude of geospatial datasets has to be taken into account. Standardized web services allow query, overlay and analysis of up-to-date data from different sources, also ad-hoc and on-demand, providing the necessary inputs for Spatial Decision Support Systems (SDSSs).

- Scenario 2 (short-term): Varying weather conditions, road construction, traffic incidents or saw mill breakdowns call for temporary revisions of strategies. LBSs being utilized in the field and being coupled with web services in the dispatching center will improve the quality of such strategy amendments.

1.3 Spatial Optimization

Geographic Information Systems (GISs) have emerged to a type of information systems that are capable of capturing, storing, checking, manipulating, analyzing and displaying spatial data (Bartelme, 2005; Burrough and McDonnell, 1998; Longley *et al.*, 1998; Worboys, 1995; Department of Environment, 1987). Due to the fact that analyzing data in different ways is a creative and innovative task, this is a field where new methodologies from other (related) fields might be applied. Thus, the introduction of related theory and methods in Geographic Information Sciences (GIScs) has and will always be of particular interest for further research.

The analysis of spatial data has also been done by geographers, who formed the term *GeoComputation* that comprises spatial analysis added with additional modeling and simulation capabilities. GIS seems not to be a mandatory "equipment" for this field of research (Macmillan, 1997; Openshaw and Abrahart, 1996). Nevertheless, in GeoComputation and also in the field of SDSSs the fact that "external" theory and methods are necessary to advance the research field, are mentioned. Thus, a number of methodologies from e.g. mathematics, statistics or hydrology were introduced in GIS, including genetic algorithms (e.g. Lei and Li, 2009), particle swarm algorithms (e.g. van Dijk *et al.*, 2002) or kernel density estimation (e.g. Leitner and Staufer-Steinocher, 2001). With such "external" methods, like the ones mentioned, it is possible to generate "what-if" scenarios and sophisticated simulations within a GIS environment (e.g. GRASS Development Team, 2009).

For *spatial optimization*, simulation is of particular interest, as any optimization is related to it, because the "best" alternative has to be determined based on a simulation. The term spatial optimization has been mostly linked to land distribution models yet – e.g. finding the best suitable location for

[...] or finding the best distribution of [...] (e.g. Herzig, 2008; Ward *et al.*, 2003; Guerra and Lewis, 2002). In this thesis this term is used in order to optimize the WSC – i.e. defining which vehicle brings timber from a particular origin (a timber pile) to a saw mill. The prefix *real-time* denotes the ability of the system to a) gather data from all stakeholders and b) to provide assignments for each truck – i.e. where to pick up and deliver timber – in (near) real-time.

1.4 Problems Addressed – Hypothesis

The problem addressed in this work is of interdisciplinary character, due to the involved research fields: GISc, Operations Research (OR), Computer Science, SDSSs and partly Forest Science. The research questions of this thesis are as follows:

- Firstly, the concept of spatial optimization is applied to the WSC in order to maximize the profit of the WSC – i.e. maximize turnover and minimize costs. By the use of an appropriate optimization methodology, an effective algorithm is formalized.

- Secondly, by the application of Service Oriented Architectures (SOAs) with regard to international standards (International Organization for Standardization (ISO) and Open Geospatial Consortium (OGC)) a system can be designed, that facilitates communication between different modules that together form the optimization system itself. By utilizing SOA, data can be collected and delivered to clients in real-time which in turn will eliminate media breaks.

Summarizing the points mentioned above the hypothesis of this dissertation is as follows:

"Real-time spatial optimization, which is in this context reflected as a combination of Geographic Information Science (GISc) and Operations Research (OR) within a Spatial Decision Support System (SDSS), improves the profitability of the Wood Supply Chain (WSC)."

Both research questions lead to an optimization of the WSC in the end, because media breaks result in increasing costs due to a planning process based on inaccurate data. The verification of this hypothesis is done by a proof of concept implementation, and the analysis of optimization results of two distinct problem instances – comparing the profit of the optimization result and the initial solution of each instance.

Additionally, the results of this thesis imply a re-use of the generated methodology in related fields of expertise and in similar spatial problems. The implications are discussed in chapter 7.3.

1.5 Relevant Literature

Basic resources in the field of GISc are the classical books of Burrough and McDonnell (1998), Longley *et al.* (1998) as well as the work of Bartelme (2005). These publications provide basic definitions and insights in the field of GISc which are very useful for developing the theoretical part of this work. A fundamental work in the field of Multi Criteria Decision Analysis (MCDA) and GIS is the book of Malczewski (1999). This book gives detailed insight in the process of decision making paired with GIS and certainly bridges the pretended gap between GISc and mathematics. Malczewski (1999) outlines the process and feedback loops that have to be implemented within a GIS in order to support decision making, which serve as building blocks for optimizing the WSC. The foundations of SDSS are the works of Gorry and Scott-Morton (1971) using the results of Simon (1960) and Anthony (1965), who defined decision types that are still valid. The paper of Malczewski (2006) outlines the literature related to MCDA with GIS and outlines the coupling approaches of GIS and decision support. In addition, the following resources have to be mentioned when talking about coupling of GIS and decision support: Malczewski (2006); Ungerer and Goodchild (2002); Jun (2000); Fedra (1996); Jankowski (1995); Nyerges (1992). The coupling of various software components may be realized by service oriented technologies. Thus, so-called SOAs and standardization play a vital role to generate interoperable systems. A detailed description of SOAs is given in (W3C, 2009b; Organization for the Advancement of Structured Information Standards, 2006; Fielding, 2000), with specific implementations like Simple Object Access Protocol (SOAP) or Representational State Transfer (REST). Standardization issues are of great importance and are covered in the documents provided by ISO (e.g. ISO, 2005b,c, 2004, 2003a,b) and OGC (e.g. OGC, 2009a, 2008b,d, 2006c, 2005).

A detailed look into mobile cartography and their possible methods and applications is done in Reichenbacher (2004) and Zipf (2004). Navigation is covered by the book of Hofmann-Wellenhof *et al.* (2003), which lays out basic positioning and navigation methods as well as an overview on selected problems and algorithms from Operations Research - namely Shortest Paths, Traveling Salesman Problems (TSPs) and Vehicle Routing Problems (VRPs). Publications by ISO and OGC (e.g. OGC, 2008c; ISO, 2007a) list possible services and applications in the LBS context. In addi-

tion, wireless communication networks are inevitable for data transfer (e.g. Krishnamurthy and Pahlavan, 2004; Umar, 2004; Schiller, 2003) and can also be used for the determination of the position - especially indoor (e.g. di Flora and Hermersdorf, 2008; Weimann, 2008; Hofmann-Wellenhof *et al.*, 2003). Practically, satellite positioning systems, like the Global Positioning System (GPS), are of higher importance for outdoor positioning. Thus, Global Navigation Satellite Systems (GNSS) are extensively used in this thesis. A general introduction is found in Hofmann-Wellenhof *et al.* (2003) and a detailed accuracy analysis is published by the U.S. Department of Defense (2001).

VRPs are the problem class that has to be dealt with when optimizing the WSC. The VRP was first published by Dantzig and Ramser (1959), and a contemporary overview is given in the book of Toth and Vigo (2002b) and in Zimmermann (2005). Detailed descriptions of VRP subproblems as well as relevant solution techniques are published there. In order to get a general introduction in Graph Theory the book of Jungnickel (2005) is an excellent resource. To solve VRPs, heuristical techniques are discussed in literature (e.g. Talbi, 2009; Bianchesini and Righini, 2005; Kytöjoki *et al.*, 2005; Pisinger and Ropke, 2005; Bramel and Simchi-Levi, 1998; Mittenthal and Noon, 1992; Cloonan, 1966). Out of the available heuristics Adaptive Large Neighborhood Search (ALNS) (Pisinger and Ropke, 2005) is chosen because this method has promising results for VRPs with Pickup and Delivery. This heuristic relies on a number of concepts that have gained interest within the community, which are: Local Search and Simulated Annealing. Local Search is a heuristic that takes a solution as input and tries to modify the given solution based on a number of criteria. As a result a new – hopefully "better" solution – is found (Funke *et al.*, 2005; Laporte and Semet, 2002). Simulated Annealing is a technique that allows an optimization process that is similar to cooling metal. With this method is possible to overcome local optima, by accepting solutions that are worse than a given solution. Besides heuristical approaches, exact solution methods gained high attractiveness due to increasing computer power and are in depth discussed in literature (e.g. Cordeau *et al.*, 2008; Azi *et al.*, 2007; Ropke, 2005). Most publications solve the VRP using column generation methods, mostly Branch and Bound and Branch and Cut. The first articles concerning Branch and Bound were published by Dakin (1965) and Land and Doig (1960). A more historical overview of exact algorithms for the VRP is given in Laporte (1992). A number of important articles have been published that cover the optimization of VRPs with pickup and delivery and time windows (e.g. Ropke and Cordeau, iRev; Ropke *et al.*, 2007). The Rolling Schedules approach is a rather old concept, first mentioned by Wagner and Whithin (1958) and further expanded and applied to real world problems by Teng *et al.* (2006) and Spitter (2005).

In order to get an overview on the WSC and the related problems the article by (von Bodelschwingh *et al.*, 2003) is a good starting point. Kühmaier *et al.* (2007, p. 73 ff) list Decision Support Systems (DSSs) for the Wood Chip Supply Chain, and show that several companies have invested in an intelligent tool for decision support. Nevertheless, they lack of a broad approach, meaning that most of them have a navigation component as well as the possibility to monitor remaining timber located at the piles. Additionally Kühmaier *et al.* (2007) mention, that the optimization of vehicle routes is not a standard feature in such software systems. Furthermore, it is not clear if these approaches incorporate international standards, at least to a certain level, so it has to be assumed that the solutions are a piece of proprietary software.

1.6 Organization of the thesis

This thesis is organized in three parts, a theoretical part, an application and results part that analyzes the obtained result and an appendix part. The theoretical part is divided into three sections which cover the topics: GISc, optimization techniques from OR and SDSS. The second part - application and results - deals with the WSC application and the experiment setup which is followed by the results and the discussion and outlook.

The theoretical part covers the topics of GISc that are relevant for the WSC and this thesis. GISc as a basis for SDSS and spatial optimization is outlined, followed by a subsection describing the relevant standards applicable. Based on standards SOAs are possible - thus their role in GISc and in WSC will be analyzed. In order to facilitate (near) real-time information sharing LBSs are of particular interest for tracking and tracing of the stakeholders of the WSC. A description of optimization techniques starts with fundamental Graph Theory as well as VRPs. The VRP is the central problem that has to be solved in order to optimize the WSC. This basic section is followed by the description of exact optimization techniques. A discussion of approximation techniques - Heuristics and Metaheuristics - shows advantages of these methods and a detailed description. The Rolling Schedule Approach (RSA) is a technique to cope with uncertainty in optimization environment is mentioned for this reason. The part covering SDSS elaborates on the genesis of SDSS and their components and the connection to MCDA. In addition, the connection between GISc and SDSS is discussed. Furthermore, the role of optimization techniques in SDSSs is explained. The different types of SDSSs are distinguished depending on the degree of integration in GIS or the integration of mathematical models in a GIS environment.

The application part shows a discussion of the experiment setup. The system architecture is layed out, that gives an overview of the application as such. Detailed description of the underlying Spatial Database Management System (SDBMS) and the modeling of the universe of discourse is given. Tracking and Tracing with LBS and reverse LBS as well as the seamless integration in the system follows consecutively. A description of the implemented LBS on mobile devices and the according service providers shows the capabilities of the system. The Decision Engine delivers optimization results based on the optimization methods described in the theoretical part. Heuristic methods are implemented in order to prove their applicability. This is followed by a section that highlights the setup for the evaluation of the WSC optimization – i.e. describes the problem instances at hand and the system on which the results are generated. In addition, the results are displayed in this chapter. There the focus lies on the work expenditure and the optimization results of the problem instances.

The discussion and outlook chapter is analyzing the results in a general way. First, alternative system architectures are highlighted, and a justification for the architecture used here is given, based on an evaluation utilizing standardized quality parameters. Secondly, the actual results of the optimization process are analyzed in detail, followed by the implications of this thesis on related fields of expertise and/or spatial problems.

Part I

Theoretical Part

Chapter 2

Geographic Information Science and Technology

Geographic Information Science and Technology (GIS&T) is the domain of information technology that enables the researcher and technician to perform tasks that have a spatial dimension. Among them are data acquisition, storage and manipulation as well as visualization and analysis. The term GIS&T consists of two distinct parts: Geographic Information Science (GISc) and Geographic Information Technology (GIT). GIT comprises all technical tools that are necessary to build up and maintain a Geographic Information System (GIS) – and generate defined projects (e.g. cadastre). The evolution of GIS from a tool – reflected by GIT – to a legitimate scientific domain is documented by the emerging term GISc (Goodchild, 1992a). GISc was given a definition in Mark (2000):

"Geographic Information Science (GISc) is the basic research field that seeks to redefine geographic concepts and their use in the context of Geographic Information System (GIS). GISc also examines the impacts of GIS on individuals and society, and the influences of society on GIS. GISc re-examines some of the most fundamental themes in traditional spatially oriented fields such as geography, cartography, and geodesy, while incorporating more recent developments in cognitive and information science. It also overlaps with and draws from more specialized research fields such as computer science, statistics, mathematics, and psychology, and contributes to progress in those fields. It supports research in political science and anthropology, and draws on those fields in studies of geographic information and society."

Goodchild (1992a) gives a list of items under the heading *content of Geographic Information Science*:

1. Data collection and measurement
2. Data capture
3. Spatial statistics
4. Data structures, algorithms and processes

5. Display

6. Analytical tools

7. Institutional, managerial and ethical issues

In Goodchild *et al.* (1999) a number of open research questions are defined. Due to the technical formulation this condenses the term GISc. Among the mentioned research questions are:

- How to store, access and transform geographic concepts with minimal information loss?
- How can geographic phenomena be explained through applying methods and models of physical and human processes?

Moreover the University Consortium for Geographic Information Science (1996) presented a survey among their institutional members that identified the ten research topics of GISc, which reflect a number of relevant topics covered by this thesis (see table 2.1), e.g. spatial analysis, distributed and mobile computing as well as interoperability. Hence, the content of this thesis has relevance for GISc. This seems to be evident, when looking at the three distinct domains of GISc defined by Goodchild *et al.* (1999):

a) the decision maker

b) the system comprising digital GIT

c) the society with norms and standards

All three domains of GISc have relevance to this project - which will be explained in the following sections. The decision maker plays a vital role in any aspect connected with GISc. Especially for the case of Wood Supply Chain (WSC) management a person supervises the process of assigning trucks and management by the system and may alter decisions of the system. Digital GIT is the basis for all intelligent systems dealing with spatial data. Hence, GIT is necessary for routing and guidance, tracking and tracing, as well as positioning and visualization of maps (see section 2.3). Norms and standards defined by "GI-society" are the guidelines for developing a certain application. Moreover certain standards are explicitly designed for practical usage (e.g. Open Location Service (OGC, 2008b)). Thus, in most cases a reference implementation exists that reveals the potential of the standard. By building a project on standards, not each part of the planned GIT has

Acquisition
Uncertainty
Spatial analysis
Future of the information infrastructure
Interoperability
Distributed and mobile computing
GIS and society
Scale
Cognition
Representations

Table 2.1: Ten Research Topics defined by University Consortium for Geographic Information Science (1996).

to be developed from scratch, which in turn saves development time. A detailed overview of the relevant standards and norms for WSC optimization is given in section 2.1.

Thill (2000) debates the nature of GIS and mentions Goodchild (1998, 1992b), who distinguishes three different classes of GIS models – having the transportation context in mind. The classes are:

- Field models, or a representation of continuous phenomena in space.
- Discrete models, that are related to discrete entities, like points, polygons or polylines.
- Network models, that represent topological connected linear entities, like roads, railways.

The three classes are not mutually exclusive, as a network consists of discrete entities – points and lines. It is obvious that the network models are most important in the domain of transportation planning – and in particular in WSC management. Due to the complex task of network modeling Goodchild (1998) proposed several data models for discrete network planning, which are based on network and discrete models, respectively. This is necessary, because network and discrete models cannot handle the complexity that transportation networks bear. Many features mentioned in this paper are not yet available, but enhance network planning (Thill, 2000; Goodchild, 1998). A number of important concepts are mentioned in the following paragraphs.

Planar and non-planar networks: Planar networks are datasets where a vertex is present at each intersection of a line, thus resulting in

the cartographic and topologic representation to be equal. This results in a dataset that is hardly navigable, due to the fact that bridges or tunnels lead to obvious errors – e.g. traversal from a path to a motorway. In non-planar networks the cartographic and topological representation differs. Hence, the modeling of overpasses (e.g. bridges) is possible.

Turn tables: They model the turns between any pair of connected lines in a network. The attributes connected with a turn are binary when the allowed or forbidden maneuvers are modeled. Storing the delays, that are the result of a turn, result in cardinal numbers.

Dynamic segmentation: This offers the possibility to allow a variation of an attribute along an network element – i.e. vertex or edge. The original data structure is preserved, and no new vertices and edges are added. On top of this, discrete entities (0- and 1-dimensional) are placed at arbitrary positions of the network. These entities are stored in separate tables, and do not affect the original network. Dynamic segmentation is preferable when an attribute should vary over the network. An example would be speed limits or the modeling of traffic congestions.

Lanes: Due to the prior discussions, information about traversal is not stored in the data structure. Detailed data on the individual lanes reveals more potential, due to extra information on every single driving lane. With this concept it is possible to model complex street intersections in an appropriate manner, e.g. from which lane the highway exit ramp starts, or which lane is solely for turning left at a road intersection. Fohl *et al.* (1996) proposed a data model for storing lane based data, where a lane is a distinct entity that is connected with other lanes in a defined way, whereas all lanes of a road share the geometry of the road itself – i.e. an edge. Hence, there is no need to determine the absolute and relative positions of the lanes, but within the model the topology is stored, which is of particular importance. Furthermore, the topological relations of lanes, such as adjacency, can be extracted.

Flows: In order to represent "real" traffic situations the number of vehicles traveling or quantities of goods transported on an edge are of interest. Hence, network flows are discussed in literature by Fohl *et al.* (1996) and Ahuja *et al.* (1993). Goodchild (1998) distinguished three types of flows: a) flows allocated to the data model, b) flows between origins and destinations where flows exist between two places in both directions (square case) – see figure 2.1 – and c) flows between origins and destinations similar to the square case, but where the set of origins is different than the set of destinations (rectangular case) – see figure 2.2. The first type describes the mapping of e.g. measurement or traffic data on the edges and vertices of the network. Some constraints can be attached like skew symmetry and

flow conservation which guarantee a ”correct” flow. These concepts are well implemented in the field of mathematics and got incorporated in a number of GIS. For the second and third type the data structure and entities to support this flow have to be created first, in order to map the network flow accordingly. So far no GIS supports the last two types of network flows from scratch.

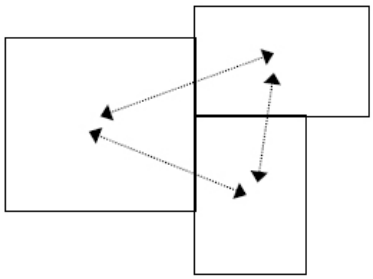


Figure 2.1: Graphical illustration of a network flow between origins and destinations – the square case. From Goodchild (1998).

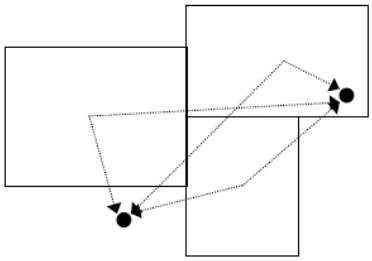


Figure 2.2: Graphical illustration of a network flow between origins and destinations – the rectangular case. From Goodchild (1998).

Temporal dimension: Common GIS and database models do not support the temporal dimension that navigation and transportation datasets have. For WSC optimization time is of crucial importance, thus, it has to be a prominent part of the data model. Hence, here is a brief look at time in GIS. Goodchild *et al.* (1993) define two distinct ways to represent time behavior:

1. **Activity events**: Time periods where any individual is connected with an activity – an event. Each event is associated with two locations (start and stop), which coincide if it is a static event. Considering the changes of entities attributes in the course of an activity, spatial attributes, such as location (position), boundary and shape, may be modified too. Based on these changes, three types of spatial-temporal behavior can be distinguished (Hornsby and Egenhofer, 2000; Renolen, 2000; Tryfonia and Jensen, 1999):

Continuous change: Entities of this type are changing constantly, like a weather system or an avalanche (see figure 2.3). There the spatial behavior, the shape and the properties are in an unsteady state.

Discrete change: Entities are in a static state and change directly through an event (see figure 2.4). An example is real estate whose owner changes immediately after an acquisition, or change in shape through partitioning the real estate into small parcels.

Stepwise change: entities have either a static state, but do change through an event (see figure 2.5). An example are vehicles movements over time, where the position is changed but the shape remains constant.

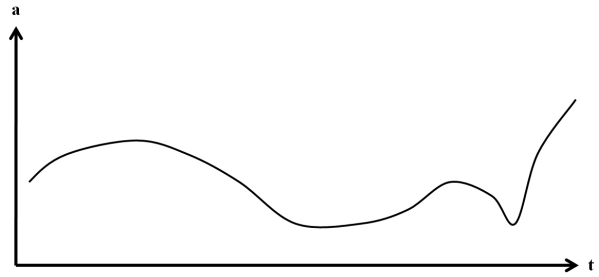


Figure 2.3: Continuous spatial-temporal behavior of attribute a over time t (after Wang and Cheng (2001) and Renolen (2000)).

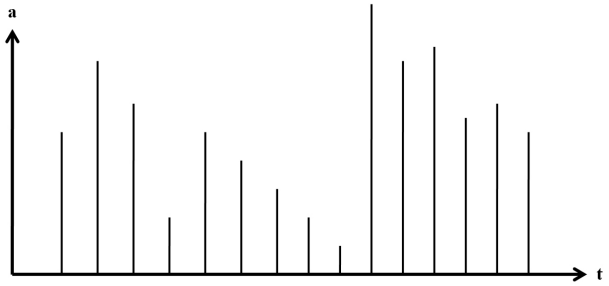


Figure 2.4: Discrete spatial-temporal behavior of attribute a over time t (after Wang and Cheng (2001) and Renolen (2000)).

2. **Space time matrices**: By the assignment of activity events to spatial entities 3D-matrices can be created consisting of three axis: time, activity and position. This allows a three dimensional analysis to identify spatial-temporal behavior, which is crucial for WSC management. Each spatial temporal movement can be evaluated using time geography (Hägerstrand, 1970). This allows the creation of *space-time paths*, which form a trajectory of an individual's movements in physical space

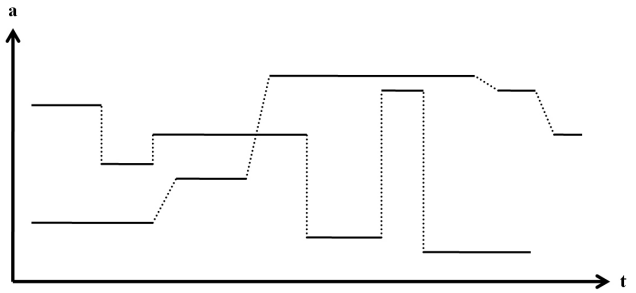


Figure 2.5: Stepwise spatial-temporal behavior of attribute a over time t (after Wang and Cheng (2001) and Renolen (2000)).

over time (see figure 2.6). The concept was enhanced by Lenntorp (1976), who incorporated the idea of transportation. Thus, it is possible to model e.g. a person's potential area that she can reach in a given amount of time starting from their actual position. This continuous space in the space-time coordinate system is named *space-time prism*. When projected onto a two-dimensional space the resulting region is called *potential path area* (see figure 2.7). According to Yu (2006), the space-time path represents historical data, while the space-time prism and potential path area allow an outlook of possible future movements by any actors – e.g. trucks participating in the WSC.

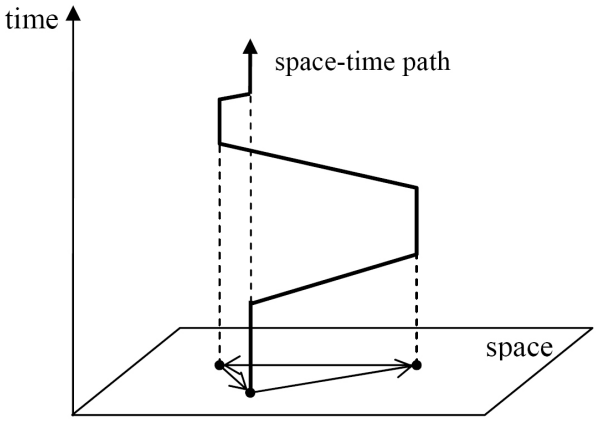


Figure 2.6: Illustration of a *space-time path*. These graphs are based on time geography (Hägerstrand, 1970). Graphic adopted from Yu (2006).

At the junction of temporal dimension and transportation in GIS ten functional requirements were defined by Adams *et al.* (2000) and mentioned in Koncz and Adams (2002) that reflect the needs of GISc in transportation. All of them have a direct or indirect link to time. To give an impression of the requirements, the most important for WSC optimization are: spatial

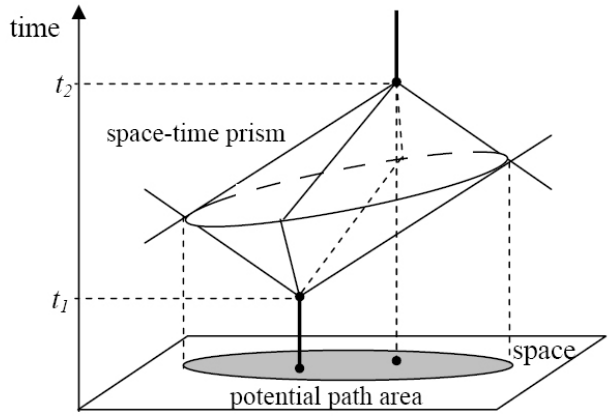


Figure 2.7: *Space-time prism* and *potential path area*. Graphic adopted from Yu (2006).

and temporal referencing methods, temporal referencing system, temporal topology, dynamics, historical databases, resolution (temporal and spatial dimension). Moreover the research activities defined by Pas (1985) support the finding, that GISc is a vital expertise in WSC management. Among the defined major areas of investigation in the field of spatial-temporal modeling Pas (1985) mentions:

- activity scheduling in time and space
- spatial-temporal, interpersonal, and other constraints
- interactions in travel decisions over time

Although the findings are related to human travel decisions, one can easily identify the correlation between the points above and WSC management, as defined in section 1.2. There activity scheduling is a vital part, due to the fact that vehicles have to pick up timber at several locations dispersed over space and deliver that to saw mills. In addition, a number of constraints are applicable to the whole problem that are mentioned in section 1.2, e.g. truck capacity or working time limits. Travel decisions of individual vehicles during a scheduled assignment are inevitable, due to changing weather conditions that result in impassable road segments. Thus, the assignment may be altered in real time, and other assignment of the vehicle have to be changed too.

The theory explained in the preceding paragraphs describes data models and additional features that are necessary for transportation modeling with GIS&T. This is of importance for the storage of spatial-temporal information, navigation issues, and creation of accurate maps with real-time

data. For these tasks a GIS – in which form ever, e.g. spatial database – is used. Goodchild (1998) mentions five application paradigms, which are also encountered in Bartelme (2005):

- digital map production
- inventory and management
- integration of data
- spatial analysis
- dynamic modeling

All listed items are of particular importance for WSC optimization, which is described in this paragraph. Digital map production is of major importance both for a logistics central office and for mobile devices, both for planning and for monitoring ("Where are the trucks now?" or "Where are the timber piles to be picked up?") and thus has a direct link to inventory and management paradigm. Data integration is of vital interest, due to emerging Spatial Data Infrastructures (SDIs) and other external datasets that have to be seamlessly incorporated into an WSC management application. The capability to review and analyze past activities is delivered by spatial analysis functionality. In this context the books of Bartelme (2005), Fotheringham *et al.* (2000), Burrough and McDonnell (1998), Bailey and Gatrell (1995) and Worboys (1995) are mentioned, that provide an introduction into spatial analysis. Dynamic modeling is one of the most important and innovative concepts concerning WSC optimization. Due to the uncertain nature of the problem itself (see chapters 3.5 and 1.2) the WSC optimization cannot deliver solutions that are valid for a long time period – e.g. one month – and has to react to changing external conditions. Thus, automatic forecasting, scheduling, as well as the evaluation of different scenarios has to be possible. In addition, the visualization of dynamic modeling results is of particular interest for stakeholders of WSC management.

2.1 Standardization in GIS

Standardization is one of the key issues of SDIs and distributed systems. Standards are inevitable in projects with a great number of participants and high data transfer rates, in order to share information between them. In addition, the intra- and interorganizational communication and coordination is possible. In the course of this chapter a brief look at interoperability is given, the term *standard* is defined followed by a number of standards that have direct influence on the thesis.

2.1.1 Interoperability

Due to the fact that in the course of this research, data are transmitted in an extensive way by various devices, the term *interoperability* is discussed first, because one set of (spatial) data has to be used by all participants of the WSC. The IEEE (1990) defines interoperability as follows:

> *"The ability of two or more systems or components to exchange information and to use the information that has been exchanged."*

The Dublin Core Metadata Initiative (2009) defines the term in a similar, but enhanced way:

> *"The ability of different types of computers, networks, operating systems, and applications to work together effectively, without prior communication, in order to exchange information in a useful and meaningful manner. There are three aspects of interoperability: semantic, structural and syntactical."*

This definition comprises different levels of interoperability that are important to GIS&T, and stresses the fact, that standards have to be defined beforehand in order to get two systems "talking". The following elaboration on interoperability levels is based on Friis-Christensen *et al.* (2005), Stuckenschmidt (2003) and Bishr (1998). *Syntactical interoperability* is related to the encoding of data, e.g. different formats that have to be harmonized. The mapping of data schemas – the logical data description (Denkers, 1992), which can be carried out in e.g. XML – is provided by *structural interoperability*. Thus, the conversion of a source into a target schema is possible at a very basic level, by resolving e.g. differences in attributes. *Semantic interoperability* is related to the meaning of a term, often in correlation with the specific context. Hence, not only technical issues are affected but also the social context has to be considered. Classical examples are extensively discussed in Harvey *et al.* (1999). There the case studies show the importance and practical problems that arise when data should be shared considering semantics. Semantics rely on ontologies, that are defined as *"formal, explicit specification of a conceptualization"* (Gruber, 1993). The result is a vocabulary for a specific knowledge domain as well as the meaning and relations between them (Yue *et al.*, 2008). To represent an ontology the Web Ontology Language (OWL) (W3C, 2009a) is one possibility besides e.g. XML. OWL is a standard in this field proposed by World Wide Web Consortium (W3C).

The first two levels of interoperability are frequently summarized as technical interoperability, due to their nature and the fact that these issues can be handled by software accurately. In contrast *semantic interoperability* is still a vital research topic and has not been solved entirely. So far a number of examples and basic initiatives have been carried out that extend capabilities of GIS&T in that direction. For Web 3.0 a number of tools exist that exploit the concept of serendipity (Ertzscheid and Gallezot, 2004), e.g. Wilbur Semantic Web Toolkit (Lassila, 2007). In addition, the Semantic Web – a part of Web 3.0 – is in development right now, which utilizes the meaning of information in the WWW (Berners-Lee, 2000). This technology will have great impact on interoperability in the field of GIS&T, due to the concept of serendipity and minimized losses in information sharing and distribution.

To come up with "fully" interoperable GISs, with respect to the three levels mentioned in the prior paragraphs, standards are inevitable. So far they provide a valuable fundament for technical and partly semantic interoperability. Thus, in the next section the standards that are of importance for real-time spatial optimization and in turn WSC optimization, are discussed.

2.1.2 Standards

Prior sections show the necessity of standards. The British Standards Institute (2009) defines a standard as follows:

> *"A standard is an agreed, repeatable way of doing something. It is a published document that contains a technical specification or other precise criteria designed to be used consistently as a rule, guideline, or definition. Standards help to make life simpler and to increase the reliability and the effectiveness of many goods and services used. Standards are created by bringing together the experience and expertise of all interested parties such as the producers, sellers, buyers, users and regulators of a particular material, product, process or service."*

Here a separation of the term "standard" itself is conducted. Considering the legal basis there are two classes of standards:

a) de facto standards

b) de jure standards

De facto standards are not created by a standards organization, like International Organization for Standardization (ISO) or Comité Européen de Normalisation (CEN), and thus are not legally binding. In the scope of this work all standards not created by a standards organization are summarized as de facto standards, although they may have become de jure standards later on. One major organization that creates and publishes de facto standards in the field of GIS&T is the Open Geospatial Consortium (OGC), whose members are companies of the GIS industry.

De jure standards are under special circumstances legally binding, which is outlined here. Globally speaking international standard organizations like ISO or CEN as well as national organizations such as the Austrian Standards Institute (ON) exist. Standards, which originate from ISO don't automatically become national standards. Even national standards may be in some points contradictory to an ISO standard. Nevertheless, a CEN standard is also a national standard if the national standards organization is a member of CEN – which is the case for ON. If an ISO standard is accepted as CEN standard then the ISO standard has to be accepted by the national CEN member organizations – which can be regarded as legally binding standards.

Such standards have to undergo an intense procedure of testing and revising through various instances before it is signed by all members and published. Thus, a de jure standard reflects recent technical terminology and actual technical rules for a specific field of expertise. Therefore, de jure standards are a valuable source of technical "know-how" due to their technical maturity and no industry driven development.

2.1.3 De facto Standards

In the following paragraphs the standards published by OGC that are strongly related to this thesis are explained. Among them there are: Web Mapping Service (WMS), Web Feature Service (WFS), Open Location Service (OpenLS), Geography Markup Language (GML), OpenGIS® Simple Features Interface Standard (SFS). Most of these standards have been – or are – in the process of being adopted by ISO and CEN.

WMS provides an HTTP interface to get georeferenced maps – images – from one or more spatial databases (located on distributed servers). The request specification comprises the layer(s) and the area of interest. The server responds with georeferenced map(s) in a defined format (e.g. JPEG, GIF, PNG) that can be displayed in any browser or geoportal application – e.g. Mapbender. For more details refer to OGC (2006c).

In order to directly access distinct features of spatial datasets the WFS interface was created. This service enables clients to send platform independent requests via HTTP, that fulfill the specifications of interfaces that are capable of manipulating and accessing spatial features. An extension to WFS specification is the Web Feature Service-transactional (WFS-T) which adds functionality to the basic WFS. Among the functionalities of WFS and WFS-T are:

- access spatial features based on constraints (spatial and non-spatial)
- create new features
- delete features
- manipulate features
- lock features

This standard requires data to be sent from spatial databases to clients. WFS data transmission utilizes another feature encoding standard, namely GML, that will be discussed in the following paragraphs. For more information concerning WFS refer to OGC (2005).

OpenLS is an umbrella for a number of standardized services in the field of Location Based Service (LBS), that have a great impact on this thesis and will be discussed in detail in chapter 2.3. The core services defined in OGC (2008d), OGC (2008c) and OGC (2008b) define a number of interfaces that are summarized in the so-called GeoMobility Server:

- Directory Service: provides a functionality to find e.g. the nearest place or service (i.e. POI's).
- Gateway Service: An interface between the GeoMobility Server and a spatial data server that contains locations of mobile users. This service enables a client to get another actual user's position.
- Location Utility Service (Geocode – getting the position out of e.g. an address – and Reverse Geocode – determining the e.g. address using the actual position)
- Presentation Service
- Route Service
- Navigation Service
- Tracking Service

The modularity of the services mentioned in the preceding list allows companies to link up with services and provide a special application on top of the standard. Well known examples are: emergency services, traffic information systems or any routing system. Thus, OpenLS has great impact on the WSC optimization, due to the fact that mobile units have to e.g. be tracked, navigate to the timber piles, and they have to send information to the logistics center.

GML is an Extensible Markup Language (XML) grammar that is capable of representing spatial features. Due to its enormous acceptance within the Geographic Information (GI)-community it has become a versatile modeling language for GIS as well as data exchange format – especially over the internet. Like almost every XML grammar, GML consists of two distinct parts: a) the schema file, that describes the document and b) the instance document with the spatial data. GML supports a set of primitives, that are the basis for the development of XML schemas, such as:

- Feature: represents a physical entity, but must not necessarily have spatial properties. E.g. a building may be a feature object with the position stored as geometry object – a point.
- Geometry: strictly stores geometry primitives
- Coordinate Reference System
- Topology
- Time
- Unit of measure
- Observations

These primitives are important for the development of application schemas, that support data exchange within a community. Such schemas consist of an XML schema file that contains relevant object classes that are of interest for a certain group of experts. Examples are CityGML (OGC, 2008a), which became a standard recently, SensorML (OGC, 2007c) or LandGML (LandXML.org Industry Consortium, 2009). More information is available in the Implementation Specification by OGC (2007a).

Simple Features Interface Standard (SFS) is a standard to ”work” with spatial data in a common and well-defined way. It facilitates object-relational databases, and thus provides an architecture for storage and retrieval of geodata and provides a number of spatial operators to manipulate data. It supports the following geometrical primitives: point, line (LineString),

polygon, $1 - n$ points in one geometry (MultiPoint), $1 - n$ lines in one geometry (MultiLineString), $1 - n$ polygons in one geometry (Multipolygon), collection of geometries (GeometryCollection). Operators that are able to manipulate spatial data can be grouped into three classes:

- Basic Operators: e.g. retrieval of geometry type, envelope.
- Spatial Relations Operators: e.g. do two features touch, or are they equal.
- Spatial Analysis Operators: e.g. buffering, difference, union.

This standard is especially interesting for real-time spatial and WSC optimization, due to the database driven system architecture, that will be discussed later. The project, that is part of this thesis, solely works with spatial data that are stored in a spatial database that supports SFS. More information on this popular standard can be found in OGC (2006a).

2.1.4 De jure Standards

Basically de jure standards are described in section 2.1.2. Here ISO standards are described that are of real importance for this thesis. In addition the OGC standards that have become ISO standards recently are mentioned here.

The Geography Markup Language (GML) as specified in ISO (2007c) is an XML dialect for the transmission and storage of geographic information. This standard is compliant with ISO 19118 (ISO, 2005a) and is equal to the OGC GML standard in the version 3.2.1.

ISO 19912 "GI - Spatial referencing by geographic identifiers" (ISO, 2003a) defines a model for spatial referencing using geographic identifiers. It also defines the components of a spatial reference system and develops components of a gazetteer. Thus it fosters the understanding of spatial references used in datasets.

ISO 19115 "Metadata" (ISO, 2003b) defines the schema for describing geographic information and services in the form of structured metadata. Hence, building a metadata catalogue based on the standard, results in information about the identification, the extent, the quality, the spatial and temporal schema, spatial reference and distribution of spatial data. It defines metadata information that are mandatory or optional, and highlights the minimal set of metadata necessary for metadata applications.

ISO 19119 "Geographic Information – Services" (ISO, 2005b), which is identical with OGC Abstract Specification - Topic 12: "The OpenGIS Service Architecture", describes the general architecture for geospatial services, that are capable of accessing and processing spatial data from distributed data sources. In addition, this standard provides the following:

- abstract framework for a service oriented architecture
- standardized interfaces which allow the development of interoperable services
- support for a service catalogue by catalogue metadata
- the usage of distributed data sources

The service architecture defined in ISO 19119 is described through four viewpoints: a) computational viewpoint, b) information viewpoint, c) engineering viewpoint and d) technology viewpoint. The computational viewpoint contains the interaction between systems through interfaces, and defines the terms *service*, *interface* and *operation*. Through the explanations, transparent service chaining is possible. The information viewpoint describes the semantics of information processing and information itself, and distinguishes six classes of services in the geospatial context - see table 2.2. In order to provide a proper distribution of data with services the engineering viewpoint documents distribution oriented aspects. The specific implementation is elaborated in the technology viewpoint. For details refer to ISO (2005b).

Geographic human interaction services	
Geographic model/information management services	
Geographic workflow/task management services	
Geographic processing services	thematic
	temporal
	spatial
	metadata
Geographic communication services	
Geographic system management services	

Table 2.2: Geographic Services as defined by ISO (2005b).

ISO 19125 Geographic Information – Simple Feature Access – Part 2: SQL Option the 2^{nd} part of ISO 19125, "Simple Feature Access", (ISO, 2004) defines an SQL schema that supports storage, retrieval, query and update of simple spatial features. In addition, this part of ISO 19125 describes a set of SQL Geometry Types together with SQL functions on those types, equal

to OGC SFS mentioned the de facto standards section (see section 2.1.3). For details refer to paragraph containing OGC SFS.

ISO 19128, "Web map server interface", (ISO, 2005c) specifies a service that produces spatially referenced maps dynamically from geographic information, equal to OGC WMS described in section 2.1.3.

ISO 19132, "Location-based services – Reference Model", (ISO, 2007a) specifies a reference model and a conceptual framework for LBSs. In addition it describes the principles and interactions between systems, which are mentioned in chapter 2.3. Typical services of LBS are discussed in detail. General principles for mobile and fixed clients are explained as well as the interactions with other ISO standards in the field of GI.

ISO 19133, "Tracking and Navigation", (ISO, 2005d) describes operations and data types for the development of tracking and navigation services. It aims to web services that are available to wireless devices.

Data types and operations for the implementation of multimodal location-based services for routing and navigation are defined in ISO 19134, "Multimodal routing and navigation", (ISO, 2007b). Like ISO 19133, its goal is to develop web services that are usable by wireless devices.

De jure standards are of particular interest for the thesis by providing the nomenclature for LBS as well as tracking and navigation, and a number of terms in the field of GI. Together with the mentioned de facto standards they form a strong set of technical fundaments this thesis relies on. Without the formulation of standards every single piece of technology would have to be "invented" or at least re-engineered (even) for the purpose of research.

2.2 Service Oriented Architectures in GIS

The term Service Oriented Architecture (SOA) is a paradigm for distributed digital capabilities able to solve a specified task (Organization for the Advancement of Structured Information Standards, 2006). Benatallah and Motahari Nezhad (2008), Papazoglou and van den Heuvel (2007) and Alonso *et al.* (2004) state that SOAs provide an architectural paradigm and abstractions that simplify the integration of various types of software within one project. Thus, software functionalities are packaged in well defined modules and exposed to externals as services and service interfaces that are independent of the state and context of neighboring services. The overall goal of SOA is the elimination of barriers of legacy systems, proprietary formats, protocols, platforms and data transmission so that various applications can be integrated and run seamlessly. Holley *et al.* (2003) state that services in

a SOA should meet the following criteria, listed here:

- All functions are defined as services.
- All services are autonomous, so that they run on an opaque level. External services do not care about the actual implementation or how results are achieved. The only thing that is visible to externals is the published service interface.
- Generally speaking, services are invocable. Hence, the protocol or infrastructure component to establish the connection are irrelevant.

If developing a SOA an architect can rely on benefits of this concept. A SOA provides a simple and scalable paradigm for systems that need interoperability. The scalability is a result of the network assumptions of SOA – as they are very low. Due to the flexibility it is easy to integrate new systems or functionalities into a grown architecture. Thus this architecture ensures a flexible and agile adaption of a system according to the user needs.

To realize SOAs a number of technologies exist. Three technologies, listed below, have gained great interest within the community. They are discussed in detail in the following paragraphs.

- Web Services – which are abbreviated as WS
- Services following the Representational State Transfer (REST) architecture (Fielding, 2000)
- Simple Object Access Protocol (SOAP) (W3C, 2009b)

WSs have gained popularity because of interoperability and standardization, due to their availability over standard Internet protocols independent of the underlying platform. Thus, they facilitate loosely coupled and decentralized applications. Web Services (WSs) heavily rely on standards, in order to ensure data transfer between two applications/computers without prior communication. Within a SOA an application/computer – generally, actor in a SOA – can have one or both states (Gisolfi, 2001):

- Service provider, that provides a Web Service with given specifications and functionalities. The implementor of a service provider has to register the service at a specific service broker. The service broker serves as clearing house or ”yellowpages” for published services (see figure 2.8).

- Service requester, that requests and invokes a service. The service requester finds an appropriate service using the service broker and binds it in order to use the service and receive the expected result (see figure 2.8).

A much simpler architecture than the one described above is found in "ordinary" web services, that can be found in a number of standards. Suppose that there are two players: the service provider and the service requester. The requester sends a request (e.g. WFS – see details for OGC WFS standard for references) and gets the result from the service (see figure 2.9).

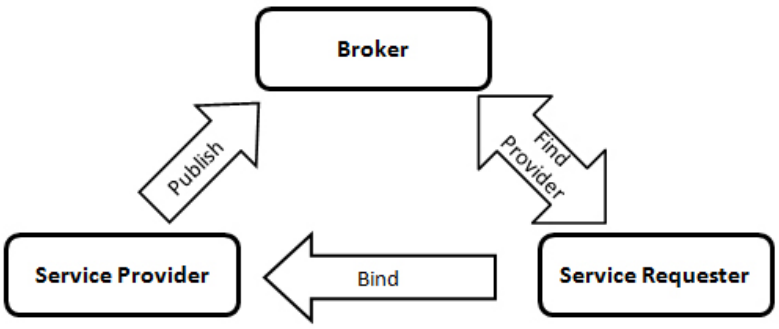


Figure 2.8: Simplified SOA architecture with web services (after Gisolfi (2001)).

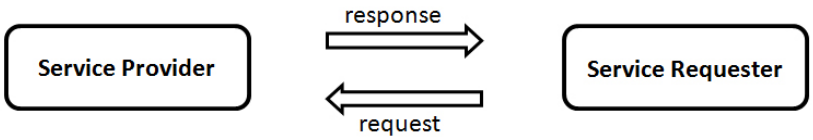


Figure 2.9: Simple web service architecture.

REST architectures are identifying how resources in a network are defined, addressed and accessed (Fielding, 2000). Consequently, applications are a set of resources, that are addressed by a Uniform Resource Identifier (URI). Each resource supports defined operations and content types (e.g. HTML, XML, CSV). REST makes use of the standard HTTP protocol and makes use of the core HTTP operations like GET, PUT, POST, DELETE. A typical client server interaction works as follows: a client application simply "knows" the URI and makes a request to the server in order to perform an operation and gets the message content in a defined format.

Generally REST architectures support stateless client server operations, which means that each request is handled as a single and independent transaction not related to prior or afterward requests. According to Fielding

(2000) they are cacheable and provide a layered nature, thus allowing intermediaries between e.g. proxies or firewalls. Advantages of the RESTful approach are that response times of servers are reduced due to caching of results. Moreover no service discovery application has to be written and maintained – services are generally discovered by their URI. Clients can be drastically simplified, due to the fact that every browser can access any application.

An example of a RESTful approach is the WWW which conforms to a great number of these principles. Due to its strict client server architecture, the stateless operations, like downloading a webpage from a resource that is identified by an URI and the fact that it relies on HTTP makes it an example for a REST approach. Some things that disturb the "perfectness" is the fact that cookies are used to facilitate session states, which clearly disrupt the example.

The SOAP protocol W3C (2009b) facilitates data exchange between spatially dispersed systems. SOAP relies on standards like XML, HTTP, FTP, SMTP or TCP. This protocol has gained high popularity not only within the GI community and has reached the status of a standard procedure for data exchange with services in nearly all possible contexts, due to its status as W3C recommendation. SOAP has a number of implementations on various platforms including PHP:SOAP, Apache CXF, Python SOAP, Microsoft .NET and mSOAP for Java which also proves the popularity within the community.

SOAP forms a lightweight protocol that exchanges data via XML. In the recommendation, the framework for the format of the data to be transmitted is defined in combination with rules to code and decode messages in a XML schema. Although possible, semantics are not covered by the recommendation per se. Due to the effort that has to be undertaken to code any information in an XML schema and to decode it at the client, the protocol needs a considerable amount of processing power – noticeable especially on mobile devices with weak processors. Moreover a great amount of metadata has to be transmitted due to the flexibility the protocol offers. Thus a single message containing a boolean variable results in a data volume up to several hundred bytes. The protocol is capable of combining complex queries and to return one single result to the client, which improves the ratio of metadata to result data, avoiding the example mentioned earlier.

This technology is used in combination with standards mentioned in chapter 2.1 in order to create a SOA in the field of GISc. Wang and Cheng (2006) describe in their literature review the shift of traditional GIS towards Geographic Information Services (GIServices). Clearly GIServices have to utilize standards and SOA as mentioned above. In particular, ISO

19119 (ISO, 2005b) as well as OGC Abstract Specification - Topic 12: "The OpenGIS Service Architecture" have to be considered in order to develop a successful SOA.

By the sharing of modularized, real-time services and data it is possible to create new information and solutions to spatial problems (Wang and Cheng, 2006; Peng and Tsou, 2003; Tsou, 2001). Rinner (2003) distinguishes four distinctive WebSDSS categories: (1) data-driven web Spatial Decision Support System (SDSS), (2) server-side web SDSS, (3) clientside web SDSS and (4) distributed SDSS. For information on SDSS refer to chapter 4. The fourth category – distributed Spatial Decision Support Services – relies on SOA. Rinner (2003) argues that complex geo-processing services with distributed Spatial Decision Support Services are one of the highest levels of SDSS (see chapter 4) and its analogue web mapping technology is represented by distributed GI services according to Peng and Tsou (2003). Yue *et al.* (2008) argue that in recent geospatial web service technologies syntactic interoperability is the main focus. In order to generate SOAs with semantic capabilities a number of enhancements to recent services have to be undertaken. Web Ontology Language (OWL) (W3C, 2009a) is a first attempt into the right direction. OWL is an approach to create ontologies and to instantiate them for the description of web services, which is described in Yue *et al.* (2008). Ostländer (2004) describes a road map towards a service based SDSS with four items that have been subject to research and development in OGC. As a result of the research agenda, two initiatives are mentioned: OGC Web Services Phase 4 (OWS-4) and OGC Web Services Phase 5 (OWS-5) (OGC, 2009a,b). OWS-4 is organized around seven topics, that have major influence on SOAs, listed here:

- Sensor Web Enablement (SWE)
- Geo Processing Workflow (GPW)
- Geo Decision Support (GeoDSS)
- Geo-Digital Rights Management (GeoDRM)
- CAD / GIS / BIM (CGB)
- OGC Location Services (OpenLS)
- Compliance Testing (CITE)

OWS-5 is organized around four topics:

- Sensor Web Enablement (SWE)

- Geo Processing Workflow (GPW)
- Agile Geography
- CITE

In this paragraph some of the terms mentioned in the lists above will be introduced, based on their importance for this thesis. SWE is a well known research item in GISc, and covers the seamless discovery, integration and delivery of sensors and their measurements in a SOA. Due to the fact that each vehicle in the WSC might be regarded as a "sensor" it is of importance for this thesis. Nevertheless, an implementation of SWE is out of the scope of this work and thus, will be just mentioned here. In particular, the GPW topic comprises the development and demonstration of interoperability among geo-processes. The interoperability is achieved through service chaining, web services utilizing OGC's Web Processing Service and SOAP. In addition, GeoDRM plays a vital role for GPW in the form of a license broker for distributed spatial models that are within a SOA. GeoDSS covers interoperable access to distributed geospatial web services for decision makers. Thus, it provides context-specific results based on information and knowledge from different communities and fields of expertise. GPW demonstrates the connection of geo-processes through service chaining. GPW delivers the basis for value-added systems that utilize interoperability and facilitating SOA. OpenLS is explained in detail in chapters 2.1 and 2.3, and has great influence on this thesis which is explained in preceding chapters.

2.3 Location Based Services

LBSs are interactive computer-based services roughly considered in the intersection of GIS, Internet and mobile devices (see figure 2.10). A number of definitions has emerged, which are similar, but stress different aspects of LBS. Some examples are given in the following list including Virrantaus *et al.* (2001):

> *"LBSs are information services accessible with mobile devices through the mobile network and utilizing the ability to make use of the location of the mobile device."*

as well as OGC (2008b):

> *"A wireless-IP service that uses geographic information to serve a mobile user. Any application service that exploits the position of a mobile terminal."*

and Ovum (2000):

> *"[...] network-based services that integrate a derived estimate of a mobile device's location or position with other information so as to provide added value to the user."*

and Reichenbacher (2004):

> *"LBSs are services accessible with mobile devices through the mobile network and utilizing the ability to make use of the location of the terminals. LBS provide specific, relevant information based on the current location to the user."*

These different definitions have some relevant aspects in common: mobile devices, location/position, network and (geographic) information, which form the basic entities that are "hiding" behind the term LBS. Basically, there are three questions that can be answered using LBS:

- Where am I?
- Where is the nearest ...?
- What is the shortest/fastest way to ...?

In research articles (e.g. Brimicombe, 2002) the temporal dimension gets more importance - as the tracking and tracing activity is definitely related to space and time. Hence, one could add another question to the list given above related to time and space, such as "Where is the nearest [...] that has opened on my arrival?".

Having that in mind, the components of a LBS can easily be determined: mobile devices, communication network, positioning component, service and application provider. In figure 2.11 the components and the architecture of the components are visualized. In this figure the service and application provider is denoted as LBS Service provider and the positioning component is represented by a Global Positioning System (GPS)-receiver. The graphic incorporates the information retrieval of a mobile device by a request via the communication network, where the service provider responds by sending the information to the client.

Mobile devices are electronical devices that allow the information request. Usually they are smartphones, Personal Digital Assistants (PDAs) or small Netbooks. In order to transmit user requests to the service provider

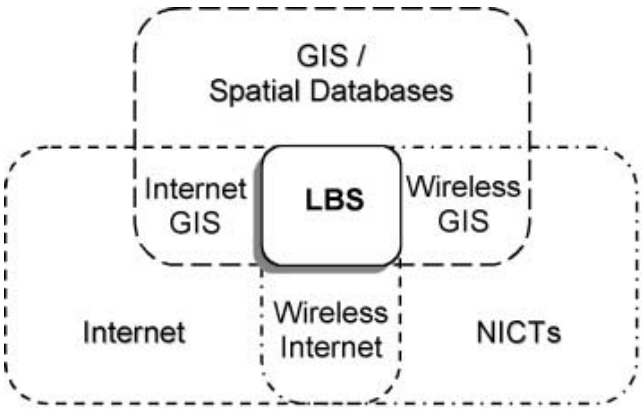


Figure 2.10: Basic niche of Location Based Services (LBSs) (Brimicombe, 2002). NICTs are portable devices that are capable of gathering their position and incorporate wireless communication – i.e. PDA's with GPS (either built-in or connected via e.g. Bluetooth).

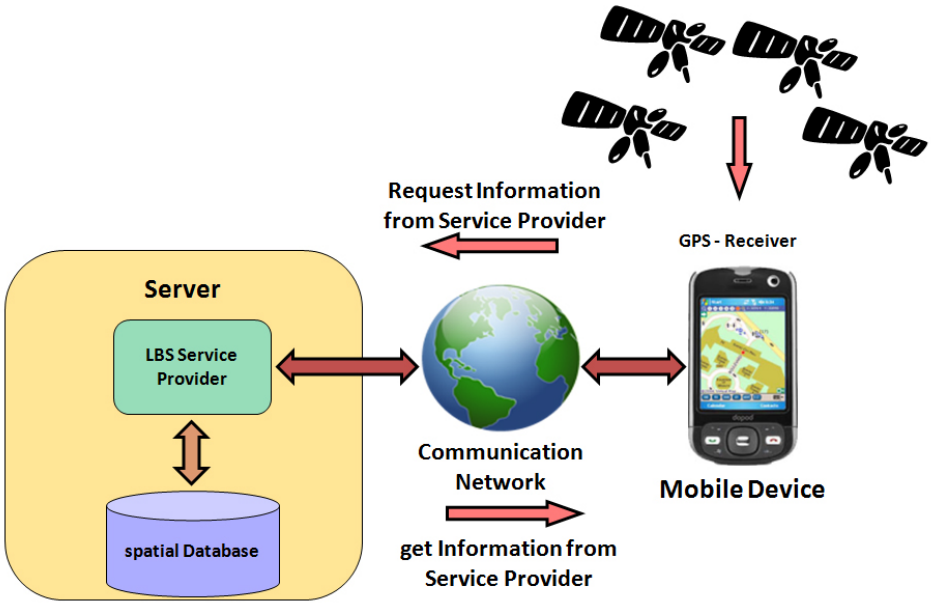


Figure 2.11: Basic LBS architecture.

as well as the requested data to the mobile device a communication network is inevitable. The communication network can be characterized by the network range and network topology (Krishnamurthy and Pahlavan, 2004; Umar, 2004; Schiller, 2003). The latter defines the network infrastructure consisting either of a large network of immobile accessible nodes or an "ad-hoc" network that is formed by the mobile devices itself. A common classification with respect to range is the following: Wireless Wide Area Network

(WWAN) – e.g. GSM and UMTS –, Wireless Local Area Network (WLAN) – e.g. IEEE 802.11 –, and Wireless Personal Area Network (WPAN) – e.g. Bluetooth. In the following paragraphs the properties of these network infrastructures are explained. This is of importance for the thesis, as mobile devices may switch between communication networks based on availability and required bandwidth for the application itself. In addition, by knowing which infrastructure will be mostly present in the "real world" it is possible to adjust the LBS application to run smoothly with the given bandwidth.

WWAN's and WLAN's are networks that have a "classical" configuration, a number of mobile terminals (=mobile devices) and fixed transceiver stations. WWAN's have been available in different generations, indicating technological evolutions of the networks. The 1st generation had a data rate of only 4.8 kbps, whereas the 2nd generation Global System for Mobile (GSM) and General Packet Radio Service (GPRS) are capable of transferring data at higher speed (GSM: 9.6-14 kbps; GPRS: 20-115 kbps). The bandwidth of GPRS is usually high enough for basic data exchange of LBS applications. The 3rd generation networks (3G), called Universal Mobile Telecommunication System (UMTS), transmit data with a speed of up to 2 Mbps which is sufficient for most multimedia applications - in particular rich data exchange of LBS applications. Due to growing coverage of 3G this is possible - even in rural areas (e.g. Mobilkom Austria, 2009).

WLAN Networks have a similar configuration like WWAN's, with Access Points (AP) as the base transceiver stations and mobile units (e.g. Notebooks, Mobile Phones, etc.). The range of WLAN covers 10-150 m within buildings and 10-300 m outside. The bandwidth usually amounts up to 54 Mbps, which is enough for all possible LBS applications.

In order to provide communication in a very short range – up to 10 m – WPAN's are used. The bandwidth is about 0.5 Mbps (Umar, 2004).

According to Krishnamurthy and Pahlavan (2004) and Schiller (2003) WLAN and WPAN are better suited for information services with high data transfer like services in facility management or consumer portal services in a shopping mall. WWAN's are suited for large scale services like fleet management, telematics or security services with limited information exchange and thus demanding a low bandwidth. Therefore, WWAN's are suitable as data transfer network for Supply Chain Management (SCM) in WSC management due to the global scale and low amount of data transferred.

The positioning component determines the position of the user in space. GPS gained a great popularity, due to mobile devices with built-in GPS sensors, but nevertheless there are a number of other positioning techniques available. WLAN access points can be used (di Flora and Hermersdorf, 2008), or dead reckoning, as well as an integration of other sensors might be

used to determine the position of the mobile device (Weimann, 2008). Another possibility for solving the positioning task is cell-oriented positioning (Hofmann-Wellenhof *et al.*, 2003), where the location is calculated based on the communication cell. Due to the fact that cells vary in size, according to the population density, the accuracy varies in wide ranges. Microcells have a radius of 100 m to 1-5 km compared to macrocells having a radius up to 10 km. In order to enhance the accuracy signal strength measurements, angle-of-arrival techniques and time-based methods (e.g. time difference of arrival) are applied.

For point positioning using standard position service (SPS) the GPS - accuracy amounts to 13 m horizontal error (95% probability level) and 22 m vertical error (95% probability level) according to the U.S. Department of Defense (2001). The SPS service is provided on the L1 signal only and takes ionospheric, tropospheric, receiver, multipath or topography effects not into account. By Differential GPS (DGPS) it is possible to reduce horizontal and vertical position errors. Hofmann-Wellenhof *et al.* (2003) state that using DGPS with C/A Code pseudorange analysis, one can reach real-time accuracy at the 1-5 m level.

The service and application provider is a piece of software that offers services to and handles requests from mobile clients. Typical services are mentioned in ISO (2007a):

- Location Services
  - Navigation: This service returns guidance instructions to a given route and the actual position of the mobile client. This topic is covered by ISO (2007b) and ISO (2005d).
  - Routing: Calculation of an "optimal" route, which is covered in detail in ISO (2007b) and ISO (2005d).
  - Tracking: Regardless of the method of position determination an interface for tracking purposes is given. Detailed Information can be found in ISO (2005d).
- Information Services
  - Data Services: Provides data and information based on client queries and delivers them in a user defined manner.
  - Event Subscription: Notifies the client if an event occurs, based on a subscription basis - hence, is dependent on user input, position, timing information.
  - Moving Object Management: Provides clients (called brokers in this context) with tracking sequences – after a successful subscription.

  - Network Data Services: In order to guarantee client side navigation as well as similar services. Thus, the clients are supplied with network data – from very small area to large area coverage – necessary to support navigation defined in ISO (2005d).

- System and User Management
  - Location Trigger Control: Enables the triggering of information.
  - User Profile Service: A data service that supports user profiles.

- Geomatics Services
  - Address Parsing: This service takes an address in any text format as input and generates an address where the elements of an address are identified and parsed in a country-specific format.
  - Gazetteer: The input of that service is a place- or landmark name and returns the position – usually as coordinates – or address. This service is described in detail in ISO (2003a).
  - Geoparsing: Takes a text with addresses or placenames as input and delivers references to locations or coordinates.
  - Location Transformation: Transforms locations of any type in any format into a new format and returns an accuracy estimation.
  - Map Service: Delivers maps to clients based on OGC WMS. For details see OGC (2006c).

- Digital Rights Management: Detailed description is found in ISO (2007a).

A number of services in the preceding list are also mentioned in OGC (2008b). Among them are: Positioning, Geographic search, Geocoding service, Reverse Geocoding, Proximity search, Routing service and Mapping service. A description is intentionally left out, due to their self-explanatory name and the given characterization above.

In LBS services are distinguished concerning information dissemination: Pull and Push services (Reichenbacher, 2004; Virrantaus *et al.*, 2001). Information of Pull services is delivered upon active user request, similar to a HTTP request by a browser. These services can be further separated into functional services, like ordering e.g. pizza, and information services (e.g. search for the next vegan restaurant) (Virrantaus *et al.*, 2001). Push services disseminate information to the user without direct request. Such services are triggered by an event, e.g. a timer. These events can have a spatial dimension, like geo-fencing. An example is walking through a historic part of a city and getting information on the sights nearby the user's position.

Nevertheless, subscription to such a service is necessary, whereas advertising messages could be delivered without subscription.

Reichenbacher (2004) defines five elementary user actions of LBS:

- orientation and localization: Delivers the position of persons and objects in context with a reference frame (e.g. coordinate system).
- navigation: Delivers routes and guidance information based on a navigable spatial dataset and a near live position of the client.
- search: Discovers available services and finds persons and objects based on user requests.
- identification: Delivers information about persons and objects which are prior unknown. This is of great importance for navigating in unfamiliar areas.
- event check: Delivers object state information and event information, based on the actual client's position. Spatial relations may be used for event triggering - e.g. has a truck been already at timber pile x, or has the truck already entered the saw mill.

These elementary user actions are consisting of the services mentioned by ISO (2007a) (see listing above). For example the orientation and localization action needs the following services: Tracking Service, Data Service, eventually Event Subscription (subscription to a service that processes the clients position, e.g. storage of positions).

For the WSC these elementary actions are of significant importance as they are basic building blocks in the WSC application, described in part II. For WSC management "orientation and localization" determines the truck's positions and transmits them in "near real-time" to a server. For navigation of trucks to timber piles and saw mills the "navigation action" is appropriate, due to providing guidance information to clients. For truck drivers, that are not familiar with a certain region, they perform an "identification action" to find the timber piles that have to be transported to saw mills. The "event check" is of necessity for the logistics center in order to see if any truck is leaving the proposed route which can be identified by a violation of a geo-fence.

2.4 Conclusion

This chapter shows that GIS&T is of inevitable importance for real-time spatial and WSC optimization. GISc and GIT provide the theoretical foun-

dation for this thesis, especially when thinking of spatial-temporal data modeling and representation. In addition, the formulation of navigable datasets is of major relevance for routing and navigation of vehicles. Generally speaking, GIS&T is the fundament for developing SDSSs, which will be discussed in chapter 4.

For the simplification of data exchange processes, in which form ever, standards are used. They provide the basis for the creation of SDIs and distributed systems. By using them, data access and exchange relies on standardized interfaces, which are well known in the community. Hence, adding other applications to the system or incorporating data produced by others is possible.

SOAs are modern concepts in GI that rely on standards. With the help of SOAs it is possible to create applications that communicate with each other by the use of services. In the field of GI a great number of possible services is defined by standards. Thus, different applications can be integrated in one project, because barriers between them can be eliminated by the use of well defined services. Especially, when LBSs are developed, these theoretical aspects are of relevance. Because every LBS relies on a combination of different systems – e.g. mobile application, service provider – the use of a SOA, that relies on standards, is inevitable. For optimizing the WSC within a real-time optimization system, LBSs are necessary to gather data from vehicles (near) live and thus the system is able to generate accurate decision support.

Chapter 3

Graph Theory and Optimization Techniques

This chapter covers relevant topics in Graph Theory that are necessary to understand the content of the following subsections. Vehicle Routing Problems (VRPs) are described in detail, due to their modeling importance in the WSC. Exact and approximate solution techniques are explained and compared thoroughly. Whereas exact techniques – often called deterministic – provide optimal solutions with guaranteed optimality, approximate algorithms – often regarded as probabilistic – provide high quality solutions in reasonable time. Due to their approximative character they do not guarantee that a global optima is found. Nevertheless, they are appropriate for practical use due to their ability to cope with large problem instances. An overview of recent optimization techniques is given in figure 3.1 and 3.2. Generally speaking, three basic classes of solution methodologies can be distinguished (Talbi, 2009; Ropke, 2005; Pisinger and Ropke, 2005):

- *Exact algorithms*: They find an optimal solution and guarantee their optimality. They are slow in comparison to other algorithm classes. Typical methods are: Branch and X (Bound/Cut/Price), as well as Constraint Programming and Dynamic Programming. Branch and X algorithms use the concept of implicit enumeration of all possible solutions and pruning of the search tree based on defined criteria – usually based on the fact that the subtree does not contain any optimal solution.
- *Approximation algorithms*: They find a solution and provide an error guarantee (Hochbaum, 1996). The error guarantee explains that the obtained solution is at worst ε times more costly than the best solution that can be found. Two classes are of interest, that can optimize a problem to any precision: the *polynomial time approximation scheme* (PTAS) and the *fully polynomial time approximation scheme* (FPTAS). It is assumed that an instance I of any problem is minimized with any $\varepsilon > 0$. Both, PTAS ans FPTAS result in a solution s $f(s) \leq (1 + \varepsilon)Opt$ – where Opt is the optimal and $f(s)$ is the objective function of the generated solution. PTAS and FPTAS differ in the fact that PTAS has polynomial time in terms of problem size, whereas problems of the FPTAS class have polynomial time in terms of

problem size and $1/\varepsilon$. Hence, the PTAS is "weaker" than the FPTAS. Details can be found in Vazirani (2001).

- *Heuristics*: These solution methods are quick compared to the classes described above and find solutions of reasonable quality. Because of their characteristic, that they have no guarantee about the solution quality, the results may be suboptimal in comparison to exact algorithms. As stated earlier, heuristics are important for solving real life problems, due to their speed and proper handling of large data sets. Metaheuristics are frameworks for the optimization of multiple problem classes, where high solution qualities can be achieved. Details can be found in this chapter and in Talbi (2009) and Voss (2001).

In the course of this thesis only one deterministic approach and metaheuristics that are important for the application itself is described. The Adaptive Large Neighborhood Search (ALNS) plays a vital role in the application part and is a combination of heuristics. It was first introduced by Shaw (1998) as Large Neighborhood Search (LNS). These topics are subject of chapter 3.4. Exact techniques are of minor importance due to their limited ability to cope with large problem instances. Talbi (2009) reports that the maximum instance size of Capacitated Vehicle Routing Problems to be solved with deterministic methods is not bigger than 60 clients. Ropke and Cordeau (iRev) mention that the Branch and Cut algorithm, proposed in their paper, is able to solve VRP instances of 500 requests to optimality whereas Ropke *et al.* (2007) declare that they solved instances with 194 requests to optimality. Thus exact optimization algorithms are of theoretical value, and have limited influence on the application itself. Nevertheless they are introduced in chapter 3.3.

Mathematical problems can be distinguished by their complexity. Therefore complexity classes are created, in order to characterize the computational time to solve the problem. Nondeterministic polynomial time (NP) is the class of the decision problems that are entirely solvable in polynomial time by a nondeterministic Turing machine. The NP-complete class denotes problems, for which results can be achieved by algorithms that have polynomial run time of the input size on a nondeterministic Turing machine – and no NP problems are more than a polynomial factor harder. NP-hard is a problem whose solution algorithm can be converted into one algorithm solving any NP-complete problem. Thus, it means, that the NP-hard problem is at less as hard as any known NP problem. NP-hard in the strong sense, denotes problems that are still NP-hard if all input is bounded in length by a polynomial function. For details refer to Garey and Johnson (1979).

An approach to cope with uncertain situations are Rolling Schedules, therefore they are described in the course of this section. The conclusion

brings all the theoretical findings together and discusses their applicability to the given problem.

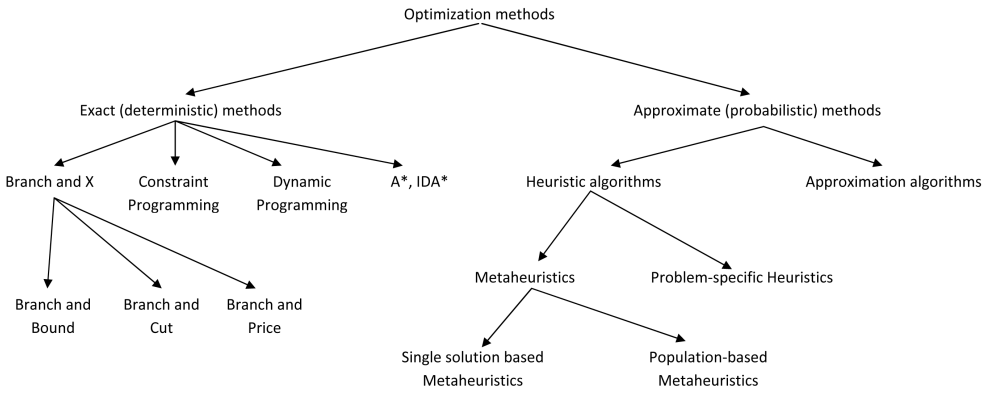


Figure 3.1: Classical optimization techniques, from Talbi (2009).

3.1 Graph Theory

Graph Theory had its very beginning in 1736 when Leonhard Euler published a paper on the problem of the *"Seven Bridges of Königsberg"*. This was the beginning of the research on topology – a term which was formed by Listing (1848). Various recent literature resources are to be found on that topic including Jungnickel (2005), Bollobas (1998), Ahuja *et al.* (1993), Piff (1991) and Chartrand (1985). Graph Theory is a subtopic of mathematics that studies graphs. Graphs are usually represented as a set of objects that are pairwise connected. This allows the modeling of complex problems and situations that are related to spatial problems – like the shortest path problem. This subsection covers the relevant theory of Graphs necessary for understanding and modeling the VRPs accordingly – which are the basic concept behind WSC optimization.

3.1.1 Graphs

An undirected graph $G = (V, E)$ is a pair consisting of a set $V \neq \emptyset$ and a set E that is defined as two-element subset of V, where elements of V are vertices and elements of E are called edges. An element of E has the form $e \in E = (a, b)$ where e denotes an edge and a and b are endpoints of the edge e. According to Jungnickel (2005) a and b are incident to e, a is adjacent to b and vice versa. In an undirected graph the edges have no particular orientation, and thus they can be traversed in both directions. In order to

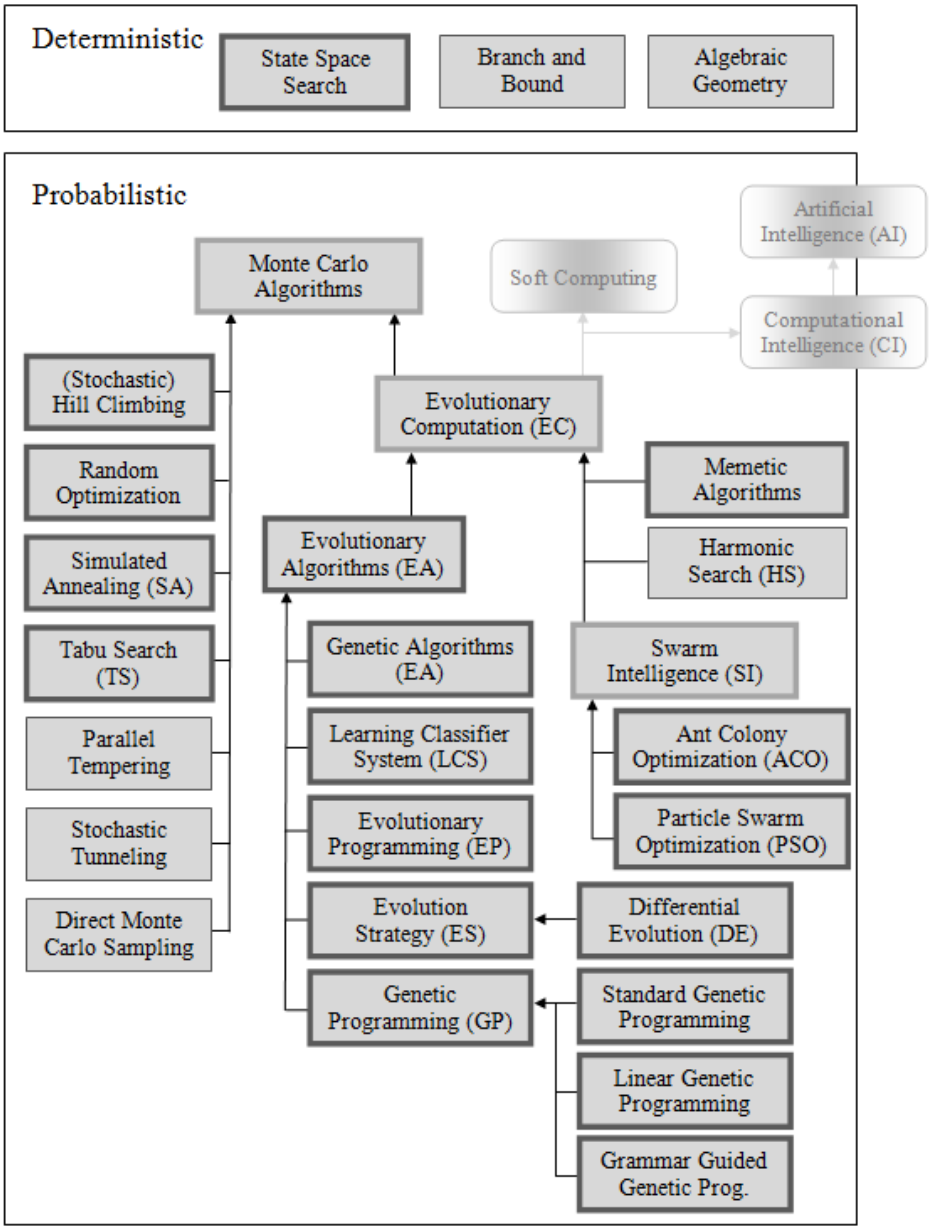


Figure 3.2: Taxonomy of optimization techniques, from Weise (2009).

support routing in a graph it is necessary to assign weights $w(e)$ to each edge e of a graph. A weight is a measure for the expense of traversing a certain edge, which allows to calculate the cost of a planned route as sum of all weights of edges that are part of the route. An induced subgraph G' of the Graph $G = (V, E)$ is a graph with $V' \subset V$ and $E|V'$ which denotes the set of all edges e with both vertices in V'. A general subgraph of G is

similar to the induced subgraph with $V' \subset V$ and $E' \subset E|V'$, whereas a spanning subgraph of G has a special set of vertices $V' = V$. The degree of a vertex v denotes the number of edges incident with v.

3.1.2 Paths, Walks and Connected Graphs

A "tour" in a graph can be defined in several ways. Due to Jungnickel (2005) there are different types of "tours" within a graph:

- walk: Assume a sequence of edges $(e_1, \ldots, e_n)$. Are there vertices $v_0, \ldots, v_n$ such that $e_i = v_{i-1}v_i$ for $i = 1, \ldots, n$, then this sequence is called a walk. In a closed walk $v_0 = v_n$.
- trail: A walk with pairwise distinct edges is called trail - if $v_0 = v_n$ it is a closed trail.
- path: If the v_j of a trail are pairwise distinct this is called a path.
- cycle: A closed trail with $n \geq 3$ where v_j are pairwise distinct.

A connected graph is a graph where a path is existing between every two vertices of the graph $u, v \in V$. According to Piff (1991) an *Eulerian cycle* exists in a connected graph where a cycle traverses every edge once. A *Hamiltonian Path* is a path which visits every vertex of the graph exactly once, thus having length $|V| - 1$. Corresponding, the Hamiltonian Cycle is a cycle that contains each vertex exactly once.

3.1.3 Digraphs

In problems related to transportation - as WSC management - there exist certain practical situations where an edge (representing a road) is only traversable in one direction. In order to model this within the scope of Graph Theory digraphs are used, that take the advantage of directed edges (see figure 3.3). According to literature, Digraphs are graphs $G = (V, E)$ with a finite set of vertices V and a set of ordered pairs $e = (u, v)$ where $u \neq v$ are elements of V. An edge $e = (u, v)$ has the start vertex u and end vertex v. The illustration of directed edges is usually done with arrows (see figure 3.3). With indegree $d_{in}(v)$ the number of edges having their end point in vertex v and by outdegree $d_{out}(v)$ the number of edges having the start point in vertex v are denoted. If $d_{in}(v) = d_{out}(v)$ is true for each vertex v the the graph is called *pseudosymmetric*.

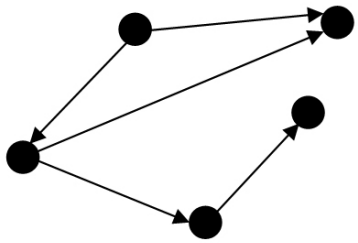


Figure 3.3: A simple illustration of a digraph.

3.2 VEHICLE ROUTING PROBLEMS

In these high mobile times an optimal distribution of goods or services is of particular interest. The definition of "optimal" varies from application to application. Nevertheless, in most cases the transportation costs have to be minimized. In section 1.1 it is argued, that by running a Supply Chain Management System (SCM-system) it is possible to achieve an improvement of delivery reliability of 40% and to reduce delivery time by 30%. In order to realize similar improvements in WSC management the WSC is modeled based on VRPs. The following explanations rely on the book of Toth and Vigo (2002c). Thus, this resource is not cited everywhere in order to maintain readability.

According to Toth and Vigo (2002b), VRPs are characterized as problems where goods are distributed between final users – the customers – and depots. The distribution relies on the existence of a set of vehicles located at one or more depots. A number of drivers pilot these trucks on a road network, which satisfies the definition of a network given in section 3.1 – i.e. a graph. The final result of a VRP is the determination of routes for each single vehicle starting and ending at a given depot. In the course of "processing" the routes, all requirements of the "customers" have to be fulfilled. In general Dorronsoro (2007) and Toth and Vigo (2002b) distinguish between the following main actors in VRPs:

- Depot(s): This are entities where goods are stored and delivered to customers and/or goods are delivered from customers.
- Customer(s): Entities that demand goods or provide goods that have to be picked up by trucks. A number of constraints can be applied according to the VRP variant. A typical constraint is the time window constraint where customers have to be serviced by trucks within a given time interval.

- Truck(s): The trucks with drivers navigate on the road network and transport goods from or to the depot(s) and the customer(s). Each truck has a predefined capacity.

- Road Network: The road network defines the connection between customers and depots. In addition, trucks have to navigate according to the road network, and route guidance is based on the network that can be represented as a graph. Several constraints concerning the road network can be formulated like speed limits, weight, turn and height constraints.

All given VRP actors can be located on the road network, in order to facilitate routing and guidance from/to customers and depots respectively. Thus depots and customers are modeled as vertices in the graph. Road connections between customers/depots are modeled as edges that can have weights. Weights in this special case can consist e.g. of driving distances or the duration to drive over the certain edge.

Given the VRP actors in the preceding list, it is possible to identify a number of related problems besides the management of the WSC. Every problem that has something to do with goods that have to be delivered to or picked up from customers and transported by a number of vehicles can be modeled by VRP. Hence, it is possible to model a number of "real world" problems, including: waste disposal services, any dispatching service (e.g. beverages) or just any repair service that has to "travel" to various customers, which is mentioned in section 7.4.

The general case of the VRP is the Capacitated Vehicle Routing Problem (CVRP) where the capacity of truck is constrained. It considers a number of customers, one depot and a number of trucks. The mathematical formulation is as follows (based on Ropke (2005); Toth and Vigo (2002a,b); Naddef and Rinaldi (2002)): Considering a Graph $G = (V, E)$ where $V = \{v_0, v_1, \ldots, v_n\}$ is a set of vertices where v_0 is the depot. Let $V' = \{V \setminus v_0\}$ be the set of n customers. $E = \{(v_i, v_j) | v_i, v_j \in V; i \neq j\}$ is the set of edges, which are also denoted as arcs. Assuming that a matrix with non-negative costs c_{ij} between each vertex (customer and depot) v_i and v_j holds the information of weights between the vertices. The use of loop arcs (v_i, v_i) is forbidden, by defining $c_{ii} = +\infty$ for all $(i \in V)$. A CVRP is called symmetric if and only if $c_{ij} = c_{ji}$ and asymmetric otherwise. Thus, in the symmetric CVRP there are undirected arcs in the arc set E. This variant is a result of representing and modeling e.g. road data in an undirected way. Thus the symmetric CVRP is likely to happen in real world problems, where appropriate directed data are not available or cannot be generated.

The graph G (of the not symmetric CVRP) must be strongly connected

and complete. In a strongly connected graph G every vertex $v \in V$ is accessible using a path between $u \in V$ and $v \in V$. A complete graph has an edge between each pair of vertices. Given a vertex set $S \subseteq V$, $\delta(S)$ and $E(S)$ denote the set of edges $e \in E$ that have only one or both endpoints in S. The cost matrix satisfies the triangle inequality,

$$c_{ik} + c_{kj} \geq c_{ij} \tag{3.1}$$

which means that it is not convenient to make deviations from the direct link between two vertices i and j. By the presence of this inequality it is guaranteed that solution algorithms for the CVRP have a feasible result. In literature (Toth and Vigo, 2002b) the Euclidean CVRP consists of points on the plane - the surface of the earth - with given coordinates and cost c_{ij} for each arc $i, j \in E$, where the cost $c_{i,j}$ is defined as the Euclidean distance, which is non-negative.

Consider that each customer has a certain demand for goods – and the demands of all customers are represented as a vector of demands d. This non-negative demand d is known in advance and may not be split. The depot has a fictitious demand $d_0 = 0$. Given a vertex set $S \subseteq V$, then

$$d(S) = \sum_{i \in S} d_i \tag{3.2}$$

denotes the total demand of the set.

A set of K identical vehicles with capacity C are available at the depot. For feasibility reasons it has to be assumed that $d_i \leq C$ for each $i = 1, \ldots, n$. Each vehicle performs exactly one route, whereas it is assumed that K is not less than K_{min}. K_{min} denotes the minimum number of vehicles necessary to serve the customers. This number can be calculated by solving the *Bin Packing Problem* (BPP). The BPP minimizes the number of bins, each with capacity C, that are required to load all n items. The BPP is NP-hard in the strong sense, but according to Martello and Toth (1990) instances can be solved up to several hundred of items. By $r(S)$ the minimum number of vehicles needed to serve all customers $S \subseteq V \setminus [0]$ is denoted. It is obvious that $r(V \setminus [0]) = K_{min}$ and the lower bound of the BPP can be estimated by

$$\lceil d(S)/C \rceil \,. \tag{3.3}$$

The CVRP results in a collection of exactly K simple circuits $\bar{r}$ – corresponding to a vehicle route)

$$\bar{r} = (v_0, v_1, \ldots, v_h, v_0) \tag{3.4}$$

where h denotes the number of customers visited on the certain route – with minimal cost, so that the sum of the costs of the arcs belonging to the circuit is minimal and that

- each circuit visits the depot vertex
- each customer is visited by exactly one truck – thus by exactly one circuit
- the sum of demands of the vertices visited by a truck (circuit) does not exceed the vehicle capacity.

The minimal cost of a route $\bar{r}$ is as follows:

$$c_{\bar{r}} = \sum_{i=0}^{h} c_{v_i, v_{i+1}} \tag{3.5}$$

Generally speaking, the CVRP is known to be NP-hard in the strong sense. Moreover it generalizes the Traveling Salesman Problem (TSP) – which searches for the minimum cost simple circuit visiting all customers/-vertices of G - a Hamiltonian cycle (see chapter 3.1) - and $C \geq d(V)$ and $K = 1$.

The CVRP can be modeled as a Linear Program (LP) with an objective and several constraints that were introduced above. Please note that R denotes the feasible simple routes, and $a_{i\bar{r}}$ is a boolean matrix with n rows and $|R|$ columns. $a_{i\bar{r}} = 1$ if and only if route $\bar{r}$ serves customer i. Based on this preconditions the CVRP can be modeled as follows:

$$\min \sum_{\bar{r} \in R} c_{\bar{r}} x_{\bar{r}} \tag{3.6}$$

subject to:

$$\sum_{\bar{r} \in R} a_{i\bar{r}} x_{\bar{r}} = 1 \qquad \forall i \in \{1, \ldots, n\} \tag{3.7}$$

$$\sum_{\bar{r} \in R} x_{\bar{r}} = K \tag{3.8}$$

$$x_{\bar{r}} \in \{0, 1\} \qquad \bar{r} \in R \tag{3.9}$$

The most important subvariant of the CVRP is the distance constrained VRP (DVRP), where the capacity constraint is replaced by the maximum length or time constraint. Each edge $(i, j) \in E$ is associated with a length $t_{i,j}$, which is the Euclidean distance. This number and the total length of

each route can not exceed a given maximum route length T. In addition, if travel times are considered, a service time s_i per customer i may be applied. The service time denotes the time period that a vehicle stops at a certain customer vertex. The service time can be added to the travel time for each arc (i, j), $t_{i,j} = \acute{t}_{i,j} + s_i/2 + s_j/2$. The original travel time of arc (i, j) is expressed as $\acute{t}_{i,j}$.

In the following sections the variants of the VRP are elaborated on. The CVRP is the general case, which is the "root" problem of VRPs. Figure 3.4 shows the VRP problem classes and their connections, respectively. Note, that the abbreviations and terms will be discussed in the following chapters. A basic overview of the VRP instances will be given here, which shall serve as starting point for the following chapters and in case one gets "lost" during reading may serve as anchor from which the reader may start over again. Here is a list of important VRP variants that are briefly described in the following paragraph:

- VRP with Time Windows
- VRP with Backhauls
- VRP with Pickup and Delivery
- VRP with Backhauls and Time Windows
- VRP with Pickup and Delivery and Time Windows
- Distance Constrained VRP

The first class is the VRP with Time Windows, where each customer is associated with a time window in which the service has to take place. The VRP with Backhauls (VRPB) is a model where customers receive or return commodities. Thus, it has to be maintained, that the remaining goods to deliver and the new returned commodities fit into the truck. The most important constraint of the VRPB is, that the deliveries nodes have to be serviced before any pickup node. In addition, goods can only be received from the depot – for delivery vertices –, and goods can only be unloaded at the depot – for pickup vertices. The VRP with Pickup and Delivery (VRPPD) represents a situation where every customer can receive and return goods from and to a given customer or depot. A subvariant of this type adds time windows in order to model the specific service times of each customer – the VRP with Pickup and Delivery and Time Windows (VRPPDTW). VRP variants mentioned in figure 3.4 but not in this thesis are the Distance Constraint VRP (DCVRP) and the VRP with Backhauls and Time Windows (VRPBTW). The DCVRP is a variant where the total Distance traveled by each vehicle is limited and must not be exceeded, and the VRPBTW is

an instance of the VRPB, which is enhanced by time windows. The last two types are of minor importance for this thesis, and because of this not described here. The VRPPDTW is essential for this dissertation and the WSC management, that is why it is explained in detail.

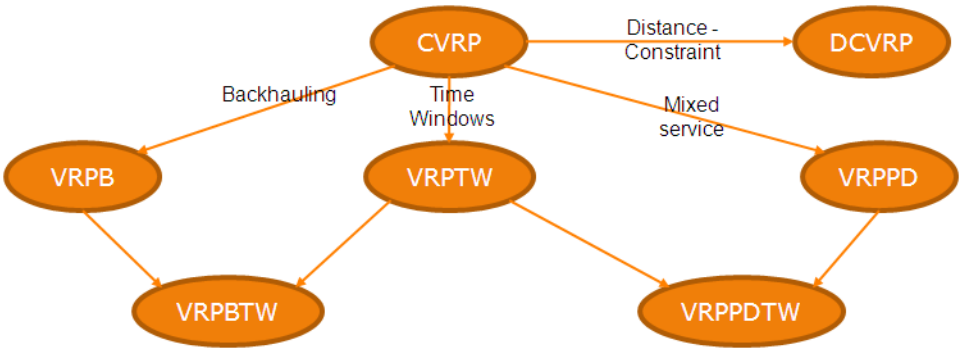


Figure 3.4: The variants of the VRP and their connections, from Toth and Vigo (2002b). The abbreviations and terms are discussed section 3.2.

3.2.1 VRP with Time Windows

The Vehicle Routing Problem with Time Windows (VRPTW) is a variant of the CVRP where each customer i is associated with a time interval - a so called time window $[a_i, b_i]$, where a and b denote start and end of the time window. There are several additional data given: the travel time $t_{i,j}$ for each arc $(i, j) \in E$, and an additional service time s_i for each customer i. Each customer has to be serviced within the given time window, and the vehicle has to stop at the vertex/customer i for the given service time s_i. If the vehicle arrives before the time window for vertex i starts the vehicle is allowed to wait at the vertex until a_i is over - i.e. the service time starts.

In the general case the cost and travel-time matrix coincide and all vehicles leave the depot at time 0. In addition, the routes have an implicit orientation due to the time window constraints even if the original matrices are symmetric. Thus, the VRPTW is modeled as asymmetric problem. According to Toth and Vigo (2002b) the VRPTW results in a set of K simple circuits with minimum cost, such that

- each circuit visits the depot vertex
- each customer vertex is exactly visited by one circuit
- the sum of the demands of the vertices visited by a circuit does not exceed the vehicle capacity C

- for each customer i the service starts in time window $[a_i, b_i]$. Moreover the vehicle stops at the vertex i for s_i time instants.

Like the CVRP the VRPTW is NP-hard in the strong sense. In addition, the Traveling Salesman Problem (TSP) with Time Windows (TSPTW) is a special case of the VRPTW, where $C \geq d(V)$ and $K = 1$.

3.2.2 VRP with Backhauls

The VRP with Backhauls (VRPB) divides the customer set V' into two subsets:

1. subset L contains n *Linehaul customers*, which require a given amount of a product to be delivered.
2. subset B comprises m *Backhaul customers*, which provide a certain amount of a product that needs to be picked up.

In the VRPB a precedence constraint is formulated: when a route needs to serve linehaul and backhaul customers, linehaul customers have to be served before any backhaul customer. A non-negative demand d_i, that has to be delivered or collected, depending on the type, is associated to each customer i, whereas the depot is assigned the demand $d_0 = 0$. An asymmetric cost matrix causes the problem to be called *Asymmetric VRP with Backhauls* (AVRPB). Both, the VRPB and the AVRPB, try to find a collection of exactly K simple circuits with minimum cost, with respect to

- each circuit visits the depot
- each customer vertex is exactly visited by one circuit
- the total quantity of goods, either backhaul or linehaul, visited by one circuit does not exceed the vehicle capacity C
- linehaul customers precede any backhaul customers.

Circuits consisting of only backhaul vertices are not allowed. K_L and K_B denote the minimum number of trucks necessary to serve the backhaul and linehaul vertices. By solving the BPP instances of the customer subsets, according to chapter 3.2, the number of vehicles can be determined. For feasibility reasons let $K \geq max(K_B, K_L)$. Both VRPB and AVRPB are NP-hard in the strong sense (Ropke, 2005; Toth and Vigo, 2002b). Both generalize the SCVRP and ACVRP arising when $B = \emptyset$ – indicating that there are no backhauls. The special case of applying Time Windows to the VRPB is called VRP with Backhauls and Time Windows (VRPBTW).

3.2.3 VRP with Pickup and Delivery

The VRPPD models a situation where each customer i is associated with two quantities d_i and p_i - representing the demand of commodities to be delivered and/or picked up at customer i. For each customer i the vertex of the origin of the delivery demand O_i, and D_i is defined - which is the destination of the pickup demand. At each customer vertex the delivery is done before the pickup. The vehicle load before arriving at a vertex is calculated by the initial load (at the depot) plus picked up demands minus delivered demands. Thus, the VRPPD tries to find a collection of K simple circuits with minimum cost, with respect to the following criteria

- each circuit starts and ends at a depot
- each customer vertex is visited by exactly one circuit
- in the course of a circuit the load is non-negative and does not exceed the vehicle capacity C.
- for each customer i, the customer O_i has to be served within the same circuit and before customer i
- for each customer i, the customer D_i has to be served within the same circuit and after customer i

Ropke (2005) calls each pair of pickup and delivery a request. The VRPPD is NP-hard in the strong sense. The VRPPD is closely connected to the Traveling Salesman Problem (TSP) with Pickup and Delivery (TSPPD) – which is a "variant" of the VRPPD where $K = 1$. A real variant of this problem is formulated by the incorporation of Time Windows, which is discussed in chapter 3.2.4.

3.2.4 VRP with Pickup and Delivery and Time Windows

The Vehicle Routing Problem with Pickup and Delivery and Time Windows (VRPPDTW) is a "mixture" of the VRPTW and the VRPPD. This variant is explained in detail, because of its relevance for this thesis. According to Ropke (2005) this problem is to be defined on a graph with $2n + 2$ nodes, where n is the number of requests. Each request is associated with vertex i and $n + i$, where i is the pickup and $n + 1$ is the delivery vertex. It is assumed that vertex 0 and $2n + 1$ represent the depot(s) where the circuits start and end. It is possible that different nodes have the same geographical

position! Thus, there is a set of pickup nodes $P = 1, \ldots, n$ and delivery nodes $D = n+1, \ldots, 2n$ where $N = P \cup D$.

For each vertex a time window $[a_i, b_i]$ is defined - analog to the VRPTW. For integrity reasons the depots may also have time windows $[a_0, b_0]$ and $[a_{2n+1}, b_{2n+1}]$. For each customer c a pickup load p_c is defined for the vertices i where $1 \leq i \leq n$. Accordingly, each customer c is assigned a delivery load d_c - defined for vertices i where $n+1 \leq i \leq 2n$. For simplicity reasons it is assumed, that if a request r contains the delivery of d_r units of goods from i to $n+i$ then let $\ell_i = d_r$ and $\ell_{n+i} = -d_r$.

Travel times $t_{i,j}$ and/or costs $c_{i,j}$ are defined on the edges (i, j), and service times s_i are defined for each vertex i (see chapter 3.2). For travel times and costs the triangle inequality has to hold. In literature an unlimited fleet of identical vehicles K with capacity C is available to serve all requests. Due to the fact that each vehicle k is not able to serve all customer requests, a set $N_k = P_k \cup D_k$ is assigned to each vehicle. N_k, P_k and D_k are subsets of N, P and D. The navigation of each vehicle is done on a network specially "designed" for each vehicle: $G_k = (V_k, E_k)$. The set $V_k = N_k \cup \{o(k), d(k)\}$ which comprises the vehicle origin and destination, i.e. the depot(s) - can be used to generate E_k as a subset of $V_k \times V_k$ which contains all feasible arcs.

The formulation of the VRPPDTW is related to the formulations given in Ropke (2005) and Desaulniers *et al.* (2002). The formulation comprises binary flow variables $x_{i,j,k}$, which is equal to 1 if vehicle k traverses arc $(i, j) \in E_k$ and 0 otherwise. The time variables T_{ik} are incorporated denoting the time when service starts at vertex $i \in V_k$. L_{ik} represents the load of the vehicle k after the service at node $i \in V_k$ is finished.

Formula 3.10 - the objective function - minimizes the total travel cost. Constraints 3.11 to 3.24 model the VRPTW accordingly. Constraints 3.11 and 3.12 guarantee that each node is visited only once by exactly one vehicle. 3.13 to 3.15 are multi-commodity flows that characterize that each vehicle starts at the origin depot $o(k)$ and terminates at the destination depot $d(k)$. 3.16 defines the "construction" of routes within the network with respect to the given schedule - given in the subsets $V_k \subset V$ and $E_k \subset E$ - and 3.17 the coordination of the routes with respect to the time windows. It has to be ensured that the pickup nodes are visited before the delivery nodes - by constraint 3.18. 3.19 defines the vehicle loads on routes. Constraints 3.20 and 3.21 describe the vehicle load ranges at pickup and delivery nodes. Each vehicle has an initial load induced by 3.22. Non-negativity of $x_{x,i,j}$ is defined by 3.23 and the binary requirement is defined by 3.24. The solution of such problems is discussed in detail in the following chapters (see 3.3 and 3.4).

$$\min \sum_{k \in K} \sum_{(i,j) \in E_i} c_{ijk} x_{ijk} \tag{3.10}$$

subject to:

$$\sum_{k \in K} \sum_{j \in N_k \cup \{d(k)\}} x_{ijk} = 1 \qquad \forall i \in P \tag{3.11}$$

$$\sum_{j \in N_k} x_{ijk} - \sum_{j \in N_k} x_{j,n+i,k} = 0 \qquad \forall k \in K, i \in P_k \tag{3.12}$$

$$\sum_{j \in P_k \cup \{d(k)\}} x_{o(k),j,k} = 1 \qquad k \in K \tag{3.13}$$

$$\sum_{i \in N_k \cup \{o(k)\}} x_{ijk} - \sum_{i \in N_k \cup \{d(k)\}} x_{jik} = 0 \qquad k \in K, j \in N_k \tag{3.14}$$

$$\sum_{i \in D_k \cup \{o(k)\}} x_{i,d(k),k} = 1 \qquad \forall k \in K \tag{3.15}$$

$$x_{ijk}(T_{ik} + s_i + t_{ijk} - T_{jk}) \leq 0 \qquad \forall k \in K, (i,j) \in E_k \tag{3.16}$$

$$a_i \leq T_{ik} \leq b_i \qquad k \in K, i \in V_k \tag{3.17}$$

$$T_{ik} + t_{i,n+i,k} \leq T_{n+i,k} \qquad \forall k \in K, i \in P_k \tag{3.18}$$

$$x_{ijk}(L_{ik} + \ell_j - L_{jk}) = 0 \qquad \forall k \in K, (i,j) \in E_k \tag{3.19}$$

$$\ell_i \leq L_{ik} \leq C_k \qquad \forall k \in K, i \in P_k \tag{3.20}$$

$$0 \leq L_{n+i,k} \leq C_k - \ell_i \qquad \forall k \in K, n+i \in D_k \tag{3.21}$$

$$L_{o(k),k} = 0 \qquad \forall k \in K \tag{3.22}$$

$$x_{i,j,k} \geq 0 \qquad \forall k \in K, (i,j) \in E_k \tag{3.23}$$

$$x_{i,j,k} \in \{0,1\} \qquad \forall k \in K, (i,j) \in E_k \tag{3.24}$$

3.3 Exact Optimization Techniques

Exact optimization techniques for WSC optimization involve a class of methods that is known as Branch and X, including e.g. Branch and Bound (Dakin, 1965; Land and Doig, 1960), Branch and Cut (Wolsey, 1998; Beasley, 1996) as well as Branch and Price algorithms (Savelsbergh, 1997). These algorithms gained high popularity and are subject of current research in the field of Operations Research (OR). They guarantee optimal solutions and no approximations. Nevertheless, they are computationally intense in terms of computing power and memory. Hence, they are limited to theoretical applications and will not be part of the WSC optimization in this thesis. First Branch and Bound is introduced, and subsequently Branch and Cut is briefly described.

3.3.1 Branch and Bound

Branch and Bound was introduced by Dakin (1965) and Land and Doig (1960). It is an exact optimization approach, that enumerates all possible solutions and discards subsets of fruitless solutions. It makes use of a decision tree where each vertex of the tree represents a subproblem of the original optimization problem. Each node of the decision tree is analyzed. Thus, the original optimization problem P is not solved in one step, but a series of problems $P_k, k = 1, \ldots, n$ with according solution spaces X_k are evaluated. The overall goals of this procedure are:

- to generate an optimal solution for P_k
- show that the "best" value of the objective function $f(P_1)$ of P_k is not better than the actual best solution
- show that the solution space X_k of P_k is empty

Branch and Bound utilizes several techniques to generate a solution of the optimization problem. Among them are Initialization, Branching and Bounding. In the Initialization phase the problem P_1 is defined which represents the root of the decision tree. P_1 can be the original problem or a LP-relaxation of it. In addition, upper and lower bounds of the optimal objective value $f(P_1)$ of P_1 are calculated. A lower bound $\underline{z}$ for maximizing problems and an upper bound $\overline{z}$ for minimization problems. If $\underline{z}$ or $\overline{z}$ cannot be defined $-\infty$ or $+\infty$ are used instead. Branching selects a previously not analyzed subproblem – i.e. node of the tree – and branches an analyzed but not terminated node. Rules exist that control how the branching is carried out and which node to evaluate next. The Bounding phase terminates a node in the search tree if the one following statements is true:

- X_k of P_k is empty
- for maximization problems: if the upper bound of $f(P_k)$ is equal or less than the actual $\underline{z}$
- for minimization problems: if the lower bound of $f(P_k)$ is equal or higher than the actual $\overline{z}$

An outline of Branch and Bound is given in figure 3.5. This method is nowadays implemented in a number of optimization software packages, both commercial and open source. Examples are CPLEX, Lingo, TOMLAB, AMPL and COIN-OR CLP, LP-solve, glpk.

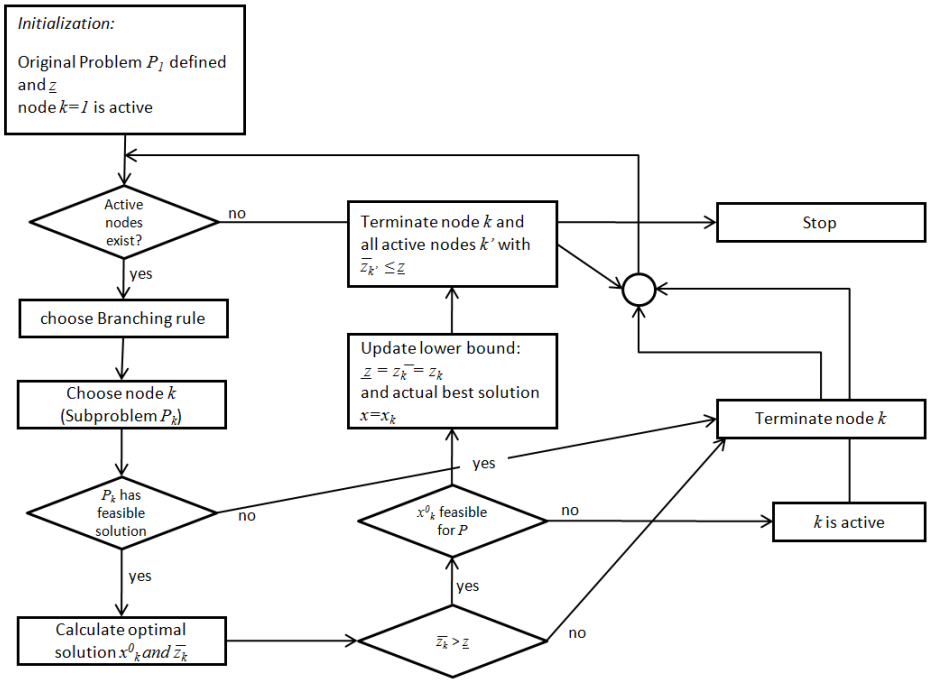


Figure 3.5: Outline of the Branch and Bound approach (based on Zimmermann (2005)).

3.3.2 Branch and Cut

Branch and Cut is a method for solving Mixed Integer Problems (MIPs) by combining Branch and Bound and Cutting Plane approaches, which were introduced by Gomory (1960). Branch and Cut was first published by Padberg and Rinaldi (1991). Branch and Cut has the ability to find integer solutions faster than Branch and Bound method, due to the applied Cutting Planes. Detailed information can be found in Wolsey (1998) and Beasley (1996).

Basically, it is similar to Branch and Bound, because the problem to be optimized is solved using e.g. the Simplex method (Dantzig, 1966) neglecting the integer constraint. The Cutting Plane algorithm adds a number of inequalities to the problem, which are fulfilled by pure integer solutions – but not by the current (non-integer) solution. By solving the problem again, using the new inequalities, a new solution is calculated. If the algorithm is not able to find new Cutting Planes to get an integer solution a Branching phase is started, that is analog to Branching process in Branch and Bound. Usually a solution with a non-integer variable z is split up into two subproblems, by adding additional constraints (e.g. constraint a: $x_1 \geq \lceil z \rceil$; constraint b: $x_1 \leq \lfloor z \rfloor$). By iteratively applying Branching to the problem a decision tree is developed, that is systematically searched (Zimmermann,

2005). Because of the great number of inequalities that are defined in the course of the algorithm it is possible to terminate nodes earlier than regular Branch and Bound.

Like Branch and Bound, it is implemented in a number of commercial and open source optimization software packages. Thus, Branch and Cut has become a standard method in the field of Mixed Integer Linear Programming (MILP). Although Branch and Bound/Cut have the advantage of being a generic solution method that does not have to be tailored to a specific problem, they have limited capabilities of solving very large instances. E.g. for the CVRP Talbi (2009) mentions that the maximum instance sizes are not bigger than 60 clients. In addition, Ropke and Cordeau (iRev) report a problem size of 500 requests and Ropke *et al.* (2007) declare that they solved instances with 194 requests to optimality.

3.4 Heuristical Optimization Techniques

Generally speaking, Heuristics – originating from the Greek word *heuriskein* – defines the technique to find and discover a new solution based on experience. Here, heuristics are optimization algorithms that re-use the information recently produced by the algorithm. This information helps to decide which candidates to search and investigate further (Talbi, 2009; Weise, 2009; Pearl, 1984). The term Metaheuristic was introduced by Glover (1986), although such algorithms existed earlier (e.g. Fogel *et al.*, 1966; Rechenberg, 1965). These methods solve general problem classes by combining several heuristics in an appropriate manner. Thus, they can be seen as *upper level* methodologies for the design of heuristics.

This chapter focuses on two main classes of Heuristics: Construction methods (greedy algorithms) and Local Search methods. In addition, the following Metaheuristics are mentioned: Simulated Annealing and Adaptive Large Neighborhood Search (ALNS), which is based on Large Neighborhood Search (LNS) and Variable Neighborhood Search (VNS). Ropke (2005) lists ALNS as "simple" heuristic, although VNS is summarized as Metaheuristic by Talbi (2009). Thus, ALNS is regarded as Metaheuristic, due to the fact that a number of competing sub-heuristics are involved in that process.

3.4.1 Construction Methods

According to Laporte and Semet (2002) Construction Methods are defined as follows:

> *"Constructive heuristics gradually build a feasible solution while keeping an eye on solution cost, but they do not contain an improvement phase per se."*

Hence, construction methods are very fast, but do not come up with an optimal solution. Although they seem "useless" at first hand, they are of practical usage. Due to the fact that these methods are very fast they are important when constructing an initial solution that is further optimized using other heuristics and/or Methaheuristics. Moreover, the quick reconstruction of a solution after a system breakdown is the main advantage of this method. Greedy algorithms are popular techniques due to their simple design, although their presence in literature is somehow fading. A typical example is shown in algorithm 3.1. A starting point is an empty initial solution and a set of Elements $E = \{e_1, e_2, \ldots, e_n\}$ that can be added to the solution. A local heuristic chooses elements from E which are not part of the solution and adds them to the solution. Moves cannot be backtracked. Construction algorithms are somehow nearsighted in developing solutions. In order to overcome this, a number of heuristics have been created that consider consequences of decisions taken (e.g. Voss *et al.*, 2005; Atkinson, 1994).

Algorithm 3.1 Greedy Algorithm Template, from Talbi (2009, p. 26)

$s = \{\}$; {initial solution is empty}
repeat
  e_i = Local-Heuristic $(E \setminus \{e/e \in s\})$; {next selected element from the set E minus the already selected elements}
  if $s \cup e_i \in F$ **then**
    {test of feasibility}
    $s = s \cup e_i$;
  end if
until complete solution found

Greedy algorithms can be subdivided into three categories:

- *Insertion heuristics*: They develop a solution by inserting just one item at a time. The choice which e.g. – in the case of VRP – customer to insert depends on the local heuristic. Typical and very simple are heuristics that choose that element that increase the overall cost least. Sequential insertion heuristics are able to construct e.g. a route at a time and parallel insertion heuristics build a number of routes or all of them.

- *Savings heuristics*: This class of construction algorithms constructs routes that serve solely one customer. In the following step the routes

are merged with respect to user defined criteria. For the case of VRPs advanced savings algorithms solve a TSP for the merged routes in order to optimize. Historically, this class of construction heuristics was introduced by Clarke and Wright (1964) – that is why it is often called the *Clarke and Wright algorithm*. Recent savings algorithms in the field of WSC optimization are proposed by Gronalt *et al.* (2004).

- *Clustering heuristics*: These algorithms have a two-phase procedure inside, where the first phase consists of a grouping of the elements that form a solution. For VRPs the customers that have to be served in one route are grouped. Subsequently the second phase generates routes for each created group. An optional third phase tries to repair errors of phase two, e.g. if a cluster of customers cannot be served by one vehicle. A "classical" clustering algorithm is the sweep algorithm (Gillet and Miller, 1974), that was created to solve CVRPs (see chapter 3.2). There sectors are created around the depot and the customers are assigned to the sectors by evaluating their polar angle with respect to the depot (see figure 3.6). The algorithm works in a very simple way: it starts with the first customer and adds more customers to the current sector as long as they can be served by one vehicle. If no more vehicles can be added to a sector a new one is opened up and filled up with vehicles. The algorithm stops if all customers have been assigned to a sector, and a TSP tour is calculated for each sector which solves the VRP. In addition, an improvement phase is done in order to enhance the result of the algorithm.

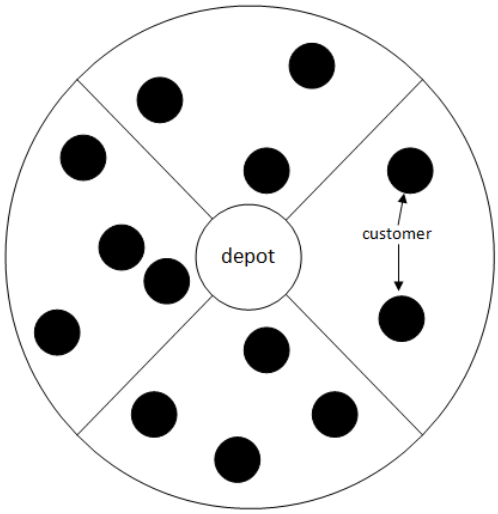


Figure 3.6: Illustration of the sweep algorithm. Customers are assigned due to their membership to a sector of a circle (based on Ropke (2005)).

3.4.2 Local Search Methods

Local Search Methods are heuristics that take an initial solution as input and modify this (now regarded as current) solution in several iterations, in order to improve the result (the objective function). The modification process itself can be seen as a selection of solutions that are neighboring the current solution. In order to select a neighbor the effects of changing the solution are evaluated in a systematic way. The neighboring solution that improves the objective function the most, is selected and is the new current solution. The heuristic stops if no neighbors are found that improve the current solution. In most cases local search results in finding a local optimum, which is – in most of the cases – not a global optimum.

In algorithm 3.2 a Local Search pseudocode is outlined. Crucial is the generation of a neighborhood and the selection of the "better" neighbor. The following description follows the explanations of Funke *et al.* (2005). In order to describe a neighborhood formally, the set of feasible solutions S have to be described followed by c, the solution cost with the function $c : S \to Q$ that maps from the solution to the corresponding cost. A neighborhood N of a solution $s \in S$ is defined as $N(s) \subseteq S$. The selection of a "better" solution (out of the neighbors) can be done using a number of strategies:

- *Steepest descent*: This strategy selects the neighbor that improves the cost function most (see algorithm 3.3 – where a minimization of the cost function is assumed). All members of a neighborhood are evaluated in terms of the cost function and the best one is selected. Due to the deterministic analysis this strategy is time consuming especially when dealing with large neighborhoods.

- *First improvement*: Here the neighbor s' is chosen, that is the first one that improves the cost function (i.e. is better than the current solution s). If such an s' is found then $s = s'$ and the algorithm starts the next iteration – creating the neighborhood and evaluating it. The first improvement strategy searches only a part of the neighborhood – in the worst case the whole neighborhood! Nevertheless it is less time consuming then the steepest descent.

- *Random selection*: This strategy randomly chooses a neighbor that improves the current solution.

As stated in the first paragraph of this page, the main disadvantage of Local Search methods is the fact that they converge towards a local optimum. Moreover, the selection/construction of an initial solution – which is the starting point of the Local Search – is very crucial for the whole process

Algorithm 3.2 Local Search algorithm template, from Talbi (2009, p. 122) and Ropke (2005, p.28)

$s = s_0 \in S$; {initial solution s_0 is current solution s}
while not Termination Criterion **do**
  Generate($N(s)$); {Generation of neighbors}
  if there is no better neighbor **then**
    Stop Local Search;
  end if
  s=s'; {Select a 'better' neighbor $s' \in N(s)$}
end while
return Final solution (local optima)

Algorithm 3.3 Steepest descent algorithm

$s = s_0 \in S$; {initial solution s_0 is current solution s}
improved = true;
while improved **do**
  $s' = min_{x \in N(s)}(c(x))$; {Generation of neighbors}
  if $c(s') < c(s)$ **then**
    {assume a minimization of the cost function}
    $s = s'$;
  else
    improved=false;
  end if
  $s = s'$; {Select a 'better' neighbor $s' \in N(s)$}
end while
return s; {Return local optimum}

and the result itself (see figure 3.7). An error estimate to measure the quality of the solution in relation to the global optimum cannot be calculated and the number of iterations is not known in advance. Thus the complexity of Local Search can be exponential in the worst case.

In order to overcome the main disadvantage of Local Search four different families of algorithms have been proposed (see figure 3.8), from which two will be directly used in this thesis and described in detail later on:

- Iterating with different initial solutions: used e.g. in greedy randomized adaptive search procedure (GRASP) (Feo and Resende, 1989);
- Accepting non-improving neighbors: With this strategies the algorithms are allowed to choose a neighbor that deteriorates the cost of the current solution. Hence, it is possible to "move out" of the "valley" (as seen in figure 3.7) and reach the global optimum. One representa-

tive – Simulated Annealing – will be content of chapter 3.4.3, and the other famous algorithm that facilitates non-improving moves is Tabu Search. Additionally Tabu Search makes use of memory structures, as it stores solutions that have been visited so far, and thus avoiding in analyzing solutions more than once. For more details refer to Glover and Laguna (1997).

- Changing the neighborhood: Here, the structure of the neighborhood is altered during the optimization. This concept is used in Variable Neighborhood Search (VNS) and Adaptive Large Neighborhood Search (ALNS), which will be described in chapter 3.4.4. VNS was introduced by Mlacenovic and Hansen (1997) and uses a family of neighborhoods. Central to VNS is the fact that a global optimum is a local optimum with respect to all neighborhoods. The pseudocode of VNS can be found in algorithm 3.4.

- Changing the objective function or the input data of the problem: Here, the optimization problem is solved by perturbing the objective function, the constraints or the input data. These approaches are not discussed in detail here – for details refer to (e.g. Talbi, 2009; Voudris and Tsang, 1999; Charon and Hudry, 1993).

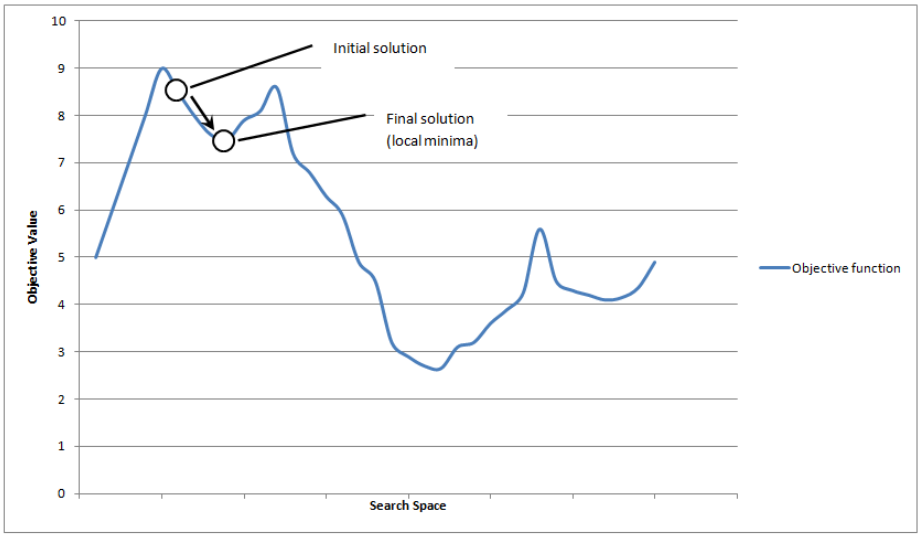


Figure 3.7: Local Search utilizing the Steepest Descent strategy. The choice of an initial solution is crucial for the result of the optimization process.

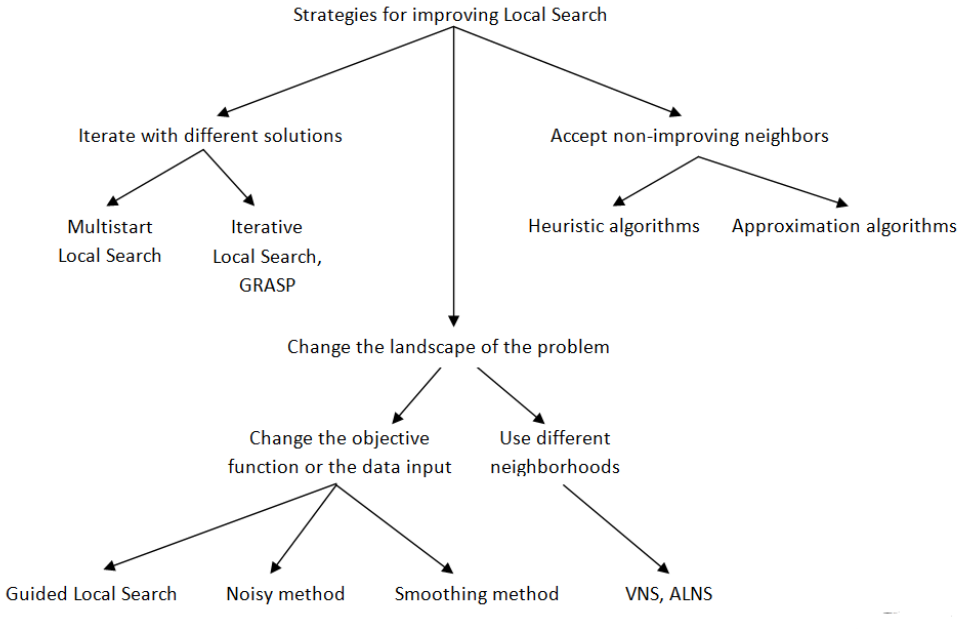


Figure 3.8: Metaheuristics for improving the Local Search – i.e. to overcome being trapped in a local optimum – from (Talbi, 2009).

Algorithm 3.4 Basic Variable Neighborhood Search, based on Mlacenovic and Hansen (1997).

Generate Neighborhood structure N_k for $k = 1, \ldots, k_{max}$
Find an initial solution s_0;
$s = s_0$; {initial solution is actual solution}
repeat
  $k = 1$;
  repeat
    Shaking: Generate a random point S' in $N_k(s)$
    Local Search: s' is the local optimum (result of local search)
    Move or not
    if s' is 'better' than s **then**
      $s = s'$;
      $k = 1$;
    else
      $k = k + 1$;
    end if
  until $k = k_{max}$
until stopping criterion is met
return s; {Return result}

3.4.3 Simulated Annealing

Simulated Annealing (SA) is a metaheuristic for large optimization problems that is based on the publications of Cerny (1985) and Kirkpatrick *et al.* (1983), which both adapt the Metropolis-Hastings algorithm (Metropolis *et al.*, 1953). SA makes use of the principle of heating and controlled cooling – called annealing – of materials in metallurgy. This technique results in increased crystal sizes and reduced defects. The metals strength, which is reached at the end of the annealing process is depending on the cooling rate and the initial temperature, that starts the melting of the material. As a result of the cooling the material converges to an equilibrium state. A comparison of the physical system and an optimization problem is given in table 3.1.

Physical System	Optimization Problem
System state	Solution
Molecular positions	Decision variables
Energy	Objective function
Ground state	Global optimal solution
Metastable state	Local optimum
Rapid quenching	Local search
Temperature	Control parameter T
Careful annealing	Simulated annealing

Table 3.1: Analogy of physical process and an optimization problem, from Talbi (2009).

SA is an algorithm that accepts worse solutions in order to escape local optima, as stated in section 3.4.2. Unlike Tabu Search it does not make use of memory structures in order to store recent analyzed solutions. SA stores only the best solution found during the optimization. The algorithm starts with an initial solution and iterates through the solution space until a stop criterion is reached. At each iteration SA develops a neighbor that is analyzed in terms of the objective function. If a neighbor is found that improves the objective function, it is accepted. For neighbors that worsen the objective function the probability of acceptance is dependent on the actual temperature T and the amount of degradation of the objective function, which is denoted as ΔE. Because of the cooling process, T is slowly decreasing which lowers the probability that such a "bad" solution is accepted. The probability strictly follows the Boltzmann distribution (see equation 3.25).

$$P(\Delta E, T) = e^{-\frac{f(s')-f(s)}{T}} \tag{3.25}$$

The control parameter that defines the acceptance probability of non-improving solutions is T. At a given temperature level a number of solutions are

explored. If an equilibrium state is found the system lowers the temperature – with respect to a cooling schedule – so that fewer non-improving solutions are accepted the further the SA algorithm advances. A pseudocode of the SA algorithm is found in algorithm 3.5, that minimizes a given optimization problem.

Algorithm 3.5 Simulated annealing algorithm, adapted from Kirkpatrick *et al.* (1983).

Require: Cooling Schedule
  Start Temperature $T = T_{max}$
  $s = s_0$; {find an initial solution that serves as actual solution}
  repeat
    repeat
      {Repeat at a fixed temperature}
      Generate a random neighbor s';
      Calculate $\Delta E = f(s') - f(s)$;
      if $\Delta E \leq 0$ **then**
        $s = s'$; {Accept the solution}
      else
        Accept s' with probability $e^{-\frac{\Delta E}{T}}$;
      end if
    until Equilibrium condition {until e.g. a given number of iterations is reached}
    $T = \alpha(T)$; {cool down the temperature}
  until stopping criterion is met {e.g. $T < T_{min}$ or given maximum number of iterations reached}
  return s; {Return best result}

The acceptance of a non-improving solution is one of the central features of SA, which is discussed in detail now. As mentioned above, such a solution is dependent on the temperature T and the change of the objective function ΔE. The probability for accepting a "bad" solution is, according to van Laarhoven and Aarts (1987, p. 10):

$$P(\Delta E, T) = e^{-\frac{\Delta E}{T}} > R \qquad R = \{x \in \mathbb{R} | (0 \leq x < 1)\} \tag{3.26}$$

where ΔE is the change in the objective function and T is the current temperature. R is a random number between 0 and 1. If T is high the probability of accepting bad moves is very high. If $T = \infty$ all solutions are accepted. The lower the temperature, the lower is the probability of accepting non-improving moves. At $T = 0$ no such solutions are accepted and SA works equivalent to a Local Search algorithm.

The cooling schedule of T is of particular importance for the success of

SA. During each iteration the temperature is lowered and thus the probability of accepting bad moves is decreasing. Crucial for the cooling schedule are the initial temperature, the equilibrium state that has to be reached at each temperature, and the cooling itself.

The choice of an appropriate starting temperature can result in an SA that behaves on the one hand like a random search when $T = \infty$, or on the other hand like a Local Search when $T = 0$. Hence, the starting temperature should be reasonably high, but not too high, in order to conduct a random search for a limited period of time. Generally speaking, there are three possible strategies that address that issue:

- Accept all: A high starting temperature allows a random search for the initial phases of the algorithm. This strategy costs computational resources.
- Acceptance deviation: T is calculated as $k\sigma$ from prior/preliminary experiments. σ is the standard deviation of the objective values and $k = -3/\ln(p)$ with the acceptance probability $p > 3\sigma$.
- Acceptance ratio: T is defined, that the acceptance probability is configured in a way that accepts at least a percentage of a_0 solutions. The formula is as follows:

$$T = \frac{\Delta^+}{\ln(m_1(a_0 - 1)/m_2 + a_0)} \tag{3.27}$$

  where m_1 and m_2 are numbers of solutions to be decreased (m_1) and increased (m_2) in preliminary experiments. Δ^+ is an average of objective values that increased in the preliminary experiments.

In order to reach an equilibrium state at a given temperature, a number of solutions have to be visited. This number can be defined in two different approaches:

- Static: A fixed number of visited solutions is defined in order to reach equilibrium state
- Adaptive: The number of visited solutions is dependent on the problem structure. Sometimes it is advisable to enforce a so-called *Non-equilibrium SA* that decreases the temperature when an acceptable solution is found. According to Cardoso *et al.* (1994) the solution quality is not affected, but the computational time is reduced.

The cooling of T can be performed using several strategies, which are outlined in this paragraph. In general T is decreased in a way that $T_i > 0$

where i denotes the i^{th} iteration, and $\lim_{i \to \infty} T_i = 0$. The simplest way is the linear cooling where $T_i = T_0 - i * \beta$. The geometric case is the most popular cooling procedure where $T_i = \alpha T$ and $\alpha \in]0, 1[$. In addition, there are a number of cooling functions including logarithmic, non-monotonic and adaptive approaches. The adaptive approach reacts to the conditions occurring in the course of the SA algorithm and the problem to be optimized. Thus, information is stored during the search which controls the cooling.

The stopping criterion mentioned in algorithm 3.5 is used to stop the algorithm after a number of iterations, in order to avoid very long computational time. Among others, the ones mentioned here, are the most popular ones:

- reaching a defined temperature, e.g. $T < 0.01$
- reaching a defined number of iterations
- performing a number of iterations without improvement of the best solution

3.4.4 Adaptive Large Neighborhood Search

The Adaptive Large Neighborhood Search (ALNS) is a local search framework that relies on Large Neighborhood Search (LNS) and Variable Neighborhood Search (VNS), which are briefly introduced in the following paragraphs. Generally speaking, ALNS enhances the Local Search by adding several heuristics to modify the current solution. A new solution is accepted if it satisfies criteria from the Metaheuristic that envelopes the Local Search. The ALNS approach was first published by Ropke and Pisinger (2006) and Pisinger and Ropke (2005). ALNS relies on LNS (Shaw, 1998), that modifies a large number of variables during an iteration of the algorithm. In order to create neighborhoods in ALNS several heuristics are applied, that rely on the *Ruin and Recreate* (Schrimpf *et al.*, 2000) and the *Ripup and Reroute* (Dees and Karger, 1982) approach. With respect to the mentioned publications, ALNS partially "destroys" the current solution to repair it instantly with the help of selected heuristics. According to the ALNS inventors it has similarities to *Very Large Neighborhood Search* (Ahuja *et al.*, 2002).

As stated in the prior paragraph VNS and ALNS share some similarities but are different in some aspects, which are outlined in the following paragraph. VNS usually operates on a family of neighborhoods resulting from of a single neighborhood that has several depths. By running the algorithm, it may reach a new local optimum and then it works with a larger neighborhood. After the procedure overcomes getting stuck in the local optimum,

it proceeds with a smaller neighborhood. ALNS instead makes use of a set of large neighborhoods that are created using destroy and repair heuristics (see figure 3.9). The pseudocode of VNS can be found in 3.4.

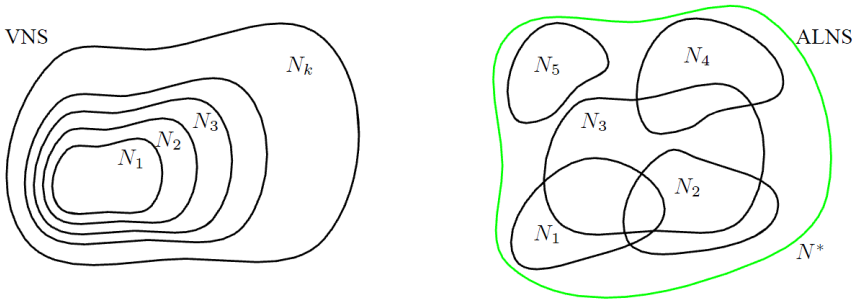


Figure 3.9: Comparison between VNS and ALNS, from Pisinger and Ropke (2005). The neighborhoods of ALNS and VNS are illustrated.

The LNS heuristic is based on an initial solution that is created using e.g. a simple construction heuristic. From this starting point it seeks to optimize a problem instance by removing and reinserting e.g. requests of a VRP instance. The pseudocode is outlined in 3.6. Here the parameter q defines the scope of the search – i.e. the neighborhood. If $q = 0$ then no requests are removed. On the contrary, if $q = n$ (where n is the number of requests) the problem is completely built up from scratch at each iteration. The bigger q the more computational effort is necessary to come up with a solution, due to the fact that the neighborhood is bigger. All solutions are accepted that improve the objective function. In addition, a more complex acceptance criterion could be employed to accept non-improving solutions in order to escape local optima.

Like LNS, ALNS is a Local Search framework but unlike LNS, it makes use of several remove and insert heuristics that are competing in the course of the algorithm. Each iteration is characterized by a selection process that results in one destroy and one repair heuristic. These heuristics alter the actual solution, and are selected with a roulette wheel selection principle based on statistics, which is described later in this section. The whole process is embedded in a SA metaheuristic in order to overcome local optima, whereas LNS is using a simple descent approach. The algorithm is described in detail in the following paragraph and the pseudocode is outlined in algorithm 3.7. First, the heuristics that destroy and repair the current solution are described. The removal heuristics are: Shaw Removal, Random Removal and Worst Removal. The insertion heuristics are: Basic greedy heuristic and the family of Regret heuristics. Secondly, the roulette wheel selection and the SA applied to ALNS are outlined.

All destroy heuristics use an integer q as an input, that defines the vari-

Algorithm 3.6 Large Neighborhood Search for VRPPDTW, from Ropke and Pisinger (2006).

Require: $s \in solutions, q \in \mathbb{N}$
  $s_{best} = s$; {initial solution is the best solution at the beginning}
  repeat
    $s' = s$;
    remove q requests from s'
    reinsert removed requests into s'
    if $f(s') < f(s_{best})$ **then**
      $s_{best} = s'$;
    end if
    if accept(s', s) **then**
      $s = s'$;
    end if
  until stopping criterion is met
  return s_{best}; {Return best result}

Algorithm 3.7 ALNS pseudocode, from Pisinger and Ropke (2005).

  construct a feasible solution s;
  $x^* = x$; {initial solution is the best solution at the beginning}
  repeat
    Choose a destroy neighborhood $\mathbb{N}^-$ and repair neighborhood $\mathbb{N}^+$ using roulette wheel selection based on previously obtained scores π_j;
    Generate a new solution x' from x using the heuristics corresponding to the chosen destroy and repair neighborhoods
    if x' can be accepted **then**
      $x = x'$;
    end if
    if $f(x) < f(x')$ **then**
      $x^* = x$;
    end if
  until stopping criterion is met
  return x^*; {Return best result}

ables that are removed. For the case of VRP variables are equal to requests that have to be served by a vehicle. In addition, Shaw Removal (Shaw, 1998) and Worst Removal need an obligatory parameter p that applies a defined degree of randomness to the heuristic. The basic approach of Shaw Removal is to remove the variables that are similar. By a strong similarity of a variable it is easy to reinsert the element at another position, and thus, create new – hopefully – better solutions. For measuring the similarity of two variables (or VRP requests) i and j a relatedness measure $R(i, j)$ is cal-

culated. If $R(i,j)$ is low then i and j are similar and more related. Ropke and Pisinger (2006) describe a relatedness measure consisting of four distinct parts (see formula 3.28): distance, time, capacity, ability of a vehicle to serve two requests. Every part is weighted using different weights φ, χ, ψ and ω. $A(i)$ and $B(i)$ are the pickup locations of a request i and T_i is the time when location i is reached by the vehicle. The parameter $d_{i,j}$ indicates the distance between request vertex i and j. l_i represents the amount of goods located at vertex i. K_i is the set of vehicles able to serve request i. The first term (with φ) measures the difference in distance and the second one (with χ) the temporal differences. The third part (with ψ) is comparing the quantity that the requests at the vertices demand. The last term provides a high relatedness if only a few or no vehicle is able to serve requests i and j. The pseudocode for the Shaw Removal facilitating the relatedness measure is given in algorithm 3.8. In the pseudocode a parameter $p \geq 1$ produces randomness in the request selection process. The lower the value the more randomness is applied.

$$\begin{aligned} R(i,j) = \varphi(&d_{A(i),A(j)} + d_{B(i),B(j)}) \\ &+ \chi(|T_{A(i)} - T_{A(j)}| + |T_{B(i)} + T_{B(j)}|) \\ &+ \psi|l_i - l_j| + \omega\left(1 - \frac{|K_i \bigcap K_j|}{\min|K_i|, |K_j|}\right) \end{aligned} \tag{3.28}$$

The Random Removal is an algorithm the picks out q requests based on randomness and removes them from the solution s. It can be seen as special Shaw Removal with $p = 1$. Indeed, a simpler implementation of the Random Removal is much faster than employing Shaw Removal for that task.

Algorithm 3.8 Shaw Removal, from Pisinger and Ropke (2005).

Require: $s \in \{solutions\}, q \in \mathbb{N}, p \in \mathbb{R}_+$
  request r = random request from S;
  set of requests $D = r$;
  while $|D| < q$ **do**
    r= random request from D;
    Array L = containing all requests from s not in D
    sort L such that $i < j \Rightarrow R(r, L[i]) < R(r, L[j])$;
    choose a random number $y \in [0, 1)$;
    $D = D \bigcup \{L[y^p|L|]\}$ {selection of a element from L at position $y^p|L|$}
  end while
  remove requests from D in s

Worst Removal is a removal heuristic that tries to remove requests that are "expensive" – i.e. worsen the objective function or require a vehicle

to travel long distances. If these requests are inserted at another position they may not be as costly anymore, thus, resulting in a better value of the objective function. Hence, the parameter $cost(i, s)$ is defined as $cost(i, s) = f(s) - f_{-i}(s)$, which represents the cost of request i in solution s, where $f(s)$ is the cost of the solution and $f_{-i}(s)$ is the cost without request i. The pseudocode is given in algorithm 3.9. As stated, Worst Removal tries to remove requests that seem to be "misplaced" in the solution, and thus are very expensive.

Algorithm 3.9 Worst Removal, from Pisinger and Ropke (2005).

Require: $s \in \{solutions\}, q \in \mathbb{N}, p \in \mathbb{R}_+$
  while $q > 0$ **do**
    Array $L =$ containing all planned requests i
    sort L by descending $cost(i, s)$;
    choose a random number $y \in [0, 1)$;
    request $r = L[y^p|L|]$ {selection of a element from L at position $y^p|L|$}
    remove r from s
    $q = q - 1$;
  end while

The first insertion heuristic, that is mentioned here, is the Basic Greedy heuristic. This simple construction heuristic performs certain – a maximum of n, the number of requests – iterations and inserts one request per iteration. $\Delta f_{i,k}$ denotes the difference of the objective function value that occurs when inserting i into route k at the position that results in the lowest increase of the objective value. If it is impossible to insert request i in route k then $\Delta f_{i,k} = \infty$. Subsequently, c_i is defined as the "cost" of inserting the request i at its best position where $c_i = \min_{k \in K}\{\Delta f_{i,k}\}$ and name this position the *minimum cost position*. Additionally the request i is selected that fulfills the function:

$$\min_{i \in U} c_i \tag{3.29}$$

U denotes the unplanned requests. The process continues until no more requests can be inserted or all requests are inserted. The Basic Greedy Heuristic has the disadvantage of inserting very "easy" – i.e. requests with low c_i – requests at the beginning and delaying the "hard" requests to the very end of the procedure. Thus, these requests may not be inserted due to routes that are already complete. In order to overcome this drawback, the family of Regret Heuristics is employed.

The Regret Heuristics tries to overcome the "problems" caused by the greedy approach, by an intelligent selection process that determines which request to insert next. $x_{i,k} \in \{1, \ldots, m\}$ is defined, where m is the number of vehicles, as the route where request i has the k^{th} lowest insertion cost – $\Delta f_{i,x_{i,k}} \leq \Delta f_{i,x_{i,k'}}$ for $k \leq k'$. Utilizing this, c_i can be expressed as $c_i =$

$\Delta f_{i,x_{i,1}}$. In this heuristic a regret value is defined as $c_i^* = \Delta f_{i,x_{i,2}} - \Delta f_{i,x_{i,1}}$, which means that the insertion cost difference of request i between the best and the second best route is calculated. In each iteration the request i is chosen that fulfills formula 3.30 and this request is inserted at its minimum cost position. As a result the request is inserted that the application would regret most if it is not done now.

$$\max_{i \in U} c_i^* \tag{3.30}$$

In addition, the Regret Heuristics – which forms a family of heuristics – is extended by the calculation of c_i^*. If the algorithm considers the k best route for request i too, there is a name for this family of heuristics: *Regret-k Heuristics*. Here, c_i^* is calculated in the following way:

$$c_i^* = \sum_{j=1}^{k} \left(\Delta f_{i,x_{i,j}} - \Delta f_{i,x_{i,1}} \right) \tag{3.31}$$

Using the definition of formula 3.31, the Regret Heuristic mentioned above – comparing best and second best route – has to be regarded as Regret-2 Heuristic. The bigger k is chosen, the earlier problems can be identified concerning the future inserting of a request i in a route.

As mentioned in this section, the remove and insertion heuristics compete with each other. Thus, a selection principle is developed for selecting the heuristic to be used in the next iteration of ALNS. To do so, the roulette wheel selection principle is appropriate. Here weights $w_i, i \in \{1, \ldots, k\}$ are assigned to each of the k heuristics and subsequently the algorithm selects the heuristic u with probability

$$\frac{w_u}{\sum_{i=1}^{k} w_i} \tag{3.32}$$

Due to the fact that the update of the weights must be performed automatically to measure the performance of each heuristic, a detailed statistic has to be established that is updated in every ALNS iteration. In order to do so, the algorithm keeps track of each heuristic and measures the score of a heuristic – high values indicate a successful strategy. According to Ropke (2005) the process has to be divided into segments of 100 ALNS iterations. At the beginning of a segment the score is set to zero. Each time a heuristic creates a new solution it is assessed and thus may increase the heuristic score by values $\alpha_1, \alpha_2, \alpha_3$ based on three distinctive cases:

- α_1: a new global best solution is found

- α_2: a solution that has not been accepted before is found, the objective value is better than the one of the current solution
- α_3: a solution that has not been accepted before is found, the objective value is not better than the one of the current solution

Because in every iteration a remove and insert heuristic is carried out, it is impossible to distinguish which one is responsible for the developed solution and the according case shown in the prior list. Thus, both heuristics are updated by the same values α_1, α_2 or α_3. At the end of a segment the weights of the heuristics are updated. There w_{ij} is the weight of heuristic i in segment j, which is equal to w_u in equation 3.32. The so calculated weight for all heuristics is used in the next segment $j+1$, which is obtained using equation 3.33.

$$w_{i,j+1} = w_{ij}(1-r) + r\frac{\pi_i}{\theta_i} \tag{3.33}$$

In equation 3.33 π_i represents the achieved score of the heuristic in the last segment and θ_i reflects the number of times heuristic i was used in the ALNS segment. $r = [0, 1]$ – the reaction factor – controls the inertia of the system. A low value of r results in a very low reaction to changes in the scores – in an extreme case if $r = 0$ then no changes in the weight values happen at all. If $r = 1$ then the last segment's score determines the weights of the heuristics. In the first iteration of the ALNS algorithm all heuristics are assigned equal weights.

ALNS is built in a Metaheuristic framework – SA – to avoid becoming trapped in a local minimum. As described earlier, SA accepts solutions that worsen the objective function. In ALNS a geometric cooling function is applied. The setting of the starting temperature T_{start} is crucial for the whole optimization process. To find a "good" value the initial solution is evaluated in terms of the objective function. T_{start} is set in a way that a solution that is v percent worse than the initial solution is accepted with probability 0.5. In order to reduce the computational time of ALNS within SA the maximum number of iterations is limited to 25000 (Ropke and Pisinger, 2006).

In addition, it is possible to reduce the number of vehicles necessary to solve the VRP instance. Due to the fact that the ALNS algorithm described here is not capable to perform this optimization, a two-stage optimization using ALNS is applied. Thus,the process starts with an initial solution that plans all requests. Then one route – i.e. a vehicle – is deleted from the solution and the requests are transferred to the set of not planned requests. Then ALNS solves the new problem. After ALNS has finished and all requests are planned then the new minimum number of necessary vehicles is found. This procedure is repeated several times until a solution has been created that leaves at least one request unplanned. Then the process steps

back to the last successful solution – which is the one that needs the lowest number of vehicles.

3.5 Rolling Schedule approach

A Rolling Schedule Approach (RSA) is scheduling approach under uncertain and dynamic conditions. In the case of WSC, future demands of customers are only known for the near future and are subject to change. In order to come up with an "optimal" solution the time is divided into:

- discrete time periods $P_1, \ldots, P_n$.
- discrete time intervals - Planning Horizons, containing one ore more time periods.

These approaches are discussed in literature and especially mentioned in Teng *et al.* (2006), Spitter (2005) and Wagner and Whithin (1958).

In systems where Rolling Horizons are appropriate, data of the phenomena to be optimized are accurate in the near future. In the far future, data are not present or not accurate. Thus, a global optimization which covers near and far future is not adequate and has to be altered due to dynamic external conditions. This results in computationally complex and time consuming calculations. RSAs work as follows (see Figure 3.10) (Teng *et al.*, 2006; Spitter, 2005):

- Step 1: Divide the time into n periods (e.g. days, months, years).
- Step 2: Define the planning horizon, i.e. the number of time periods planning ahead.
- Step 3: Solve the optimization task for the planning horizon starting at period $p = 1$.
- Step 4: Roll the horizon forward by one period and solve the optimization task for the planning horizon starting at $p = p + 1$.

The optimization task to be solved in Step 3 and Step 4 can be any optimization operation. In the case of solving the WSC the optimization task is the VRPPDTW described in 3.2.4, whereas in literature mostly machine or production operations are investigated.

For the case of SCM in the WSC timber ready for transportation to the saw mills is not known in advance - especially in far future. Thus, the

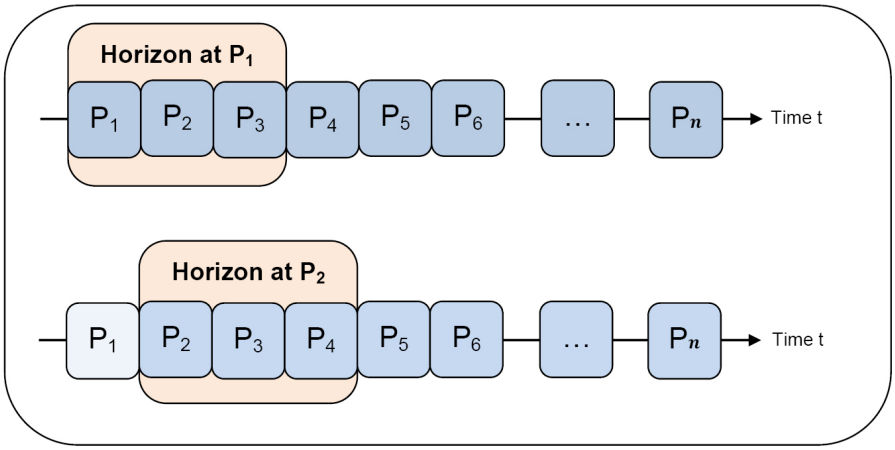


Figure 3.10: Rolling Schedules approach illustration. The timeline is divided into discrete time periods $P_1, \ldots, P_n$ and a Planning Horizon - abbreviated "Horizon" - defines the duration that the "system" planes ahead. After P_1 is over the Horizon is "rolled" forward by one time period.

accuracy of a global optimization model - as mentioned above - would be very low. If the WSC is planned ahead only for a short period of time - usually several days - the accuracy of the results is increased. In addition, weather influences the possibility of transport operations. Due to snow in mountainous regions roads may be impassable which disrupts the SCM if planned to far ahead! Due to these uncertain external conditions in the WSC it is necessary to apply the Rolling Schedules approach in the SCM process.

3.6 Conclusion

The topics discussed in the previous section are of great importance for optimizing the WSC. In order to model the process accordingly Graph Theory is used because it provides basic definitions. Furthermore, the process of WSC optimization can be seen as Graph theoretical problem – i.e. as a route problem (Jungnickel, 2005; Bollobas, 1998). Route problems include – among others – the TSP and the VRP that can be regarded as a generalized TSP. Due to the nature of the WSC optimization, that is subject of this thesis, it is obvious that it has to be modeled as VRP. Here, a detailed taxonomy of VRP subproblems is given, that identifies the VRPPDTW as the one that suits best for optimizing the WSC, according to the definition of the VRPPDTW with time windows for each vertex, pickup and delivery nodes (see chapter 3.2.4).

In order to optimize the VRPPDTW two classes of optimization techniques are applicable: Heuristics and Exact algorithms. Exact algorithms are outlined, but they have limited relevance as their capability of solving large problem instances is limited. Heuristics and Metaheuristics are applied in this study in order to solve the VRP. An overview of important heuristics for constructing initial solutions is given, and the well known Local Search is discussed. Metaheuristics, like Simulated Annealing and Adaptive Large Neighborhood Search are applied to solve the WSC in this thesis. Due to the fact that information about future events (e.g. new timber piles, weather conditions) is not accurate. Thus, the longer the application plans into the future, the more suitable is the application of a RSA. This method optimizes a given Planning Horizon (e.g. three days) that contains a number of time periods (e.g. one day) and updates the solution if a time period is over. So the system is able to consider future events – as they are already known – and is flexible enough to react to short-time changes – like deteriorating weather conditions.

The theory mentioned in this chapter is a substantial part of Spatial Decision Support Systems (SDSS) described in the chapter 4. With the help of the outlined methods the system is able to select a (near-) optimal solution with respect to a number of user defined criteria. Without optimization techniques it would not be possible to search the myriads of possible solutions and select the ”best” one, which is central to SDSS – i.e. the Model Base Management System (see chapter 4.1) and the integration of optimization techniques (see chapter 4.3).

Chapter 4

Spatial Decision Support Systems

SDSSs are software systems that support users in making decisions that are semi-structured in nature. This chapter focuses on the components of SDSSs and the genesis if these systems. Additionally, the integration of optimization methods is described followed by concluding remarks.

4.1 Definition, Components and brief history of SDSS

A Decision Support System (DSS) is a system that is used to support complex decision making and problem solving in a computerized environment utilizing modern information technology (Shim *et al.*, 2002). By adding the term spatial as prefix to DSSs the term Spatial Decision Support Systems (SDSSs) is generated, which in turn reveals that the system is capable of solving spatial problems. Wesels and Wierzbicki (2000), as well as Densham (1993), state that the main task of an SDSS is the support in solving complex, semi-structured spatial problems. According to (Malczewski, 1999, 1997) SDSSs are

> *"interactive, computer-based systems designed to support a user or group of users in achieving a higher effectiveness of decision making while solving a semi-structured spatial decision problem."*

Due to the fact that spatial problems have a great number of decision alternatives with varying properties they are regarded as complex per se (Malczewski, 1999). Due to Malczewski's definition, SDSSs are created in order to increase the effectiveness – not efficiency – of semi-structured problems. One main focus of SDSSs is the capability of making transparent and traceable decisions either by humans or the computer itself.

A competing technology to SDSSs are Spatial Expert Systems (SESs). According to Malczewski (1999, p. 283) SESs software systems are

> *"that employ reasoning methodologies in a particular [. . .] do-*

> *main in order to transfer expertise and render advice or recommendations, much like a human expert".*

It aims at being usable by non-experts to improve their capabilities, trying to replace an expert for this particular field of expertise as such. Thus, it is not as flexible as a SDSS in altering the recommendations of the system.

According to Shim *et al.* (2002) the basic idea of Decision Support Systems (DSS) was defined in Gorry and Scott-Morton (1971) using the results of Anthony (1965) and Simon (1960). Anthony (1965) developed categories of decision types that are still valid in recent publications. He distinguished between: strategic planning (executive decisions regarding overall mission and goals), management control (middle management guiding the organization to goals), and operational control (first line supervisors directing specific tasks). Simon (1960) stated that decision problems range from programmed to non programmed. Programmed meaning repetitive, well structured decisions and non programmed decisions are new, ill-structured and generally very difficult problems. Gorry and Scott-Morton (1971) use the terms structured and unstructured instead of programmed and non programmed. These two types are affecting the human decision process so that the decision maker must have detailed knowledge and insight in the unstructured case. Subsequently, the process itself is of interest in order to support the human. Simon (1960) claims that there are three problem solving phases which were named by Gorry and Scott-Morton (1971) and are mentioned in Makowski and Wierzbicki (2000):

- *intelligence phase*: the decision environment is scanned for conditions that are calling for a decision
- *design phase*: within the environment the results of possible – in the course of this phase invented or developed – actions are analyzed
- *choice phase*: selection of an action

In general, the sequence of the phases mentioned above is as follows: intelligence phase precedes design and design precedes choice phase. Although this seems very cyclic and easy to understand, the whole process tends to be far more complex and solved in an iterative manner. Hence, the design phase often requires an intelligence phase due to subproblems that arise during solving the main problem. These subproblems have to be solved in their own decision cycle which in turn results in an iterative process. Generally speaking, the human decision process covers three basic questions related to the phases (Dewey, 1910): What is the problem? What are the alternatives? Which alternative is best?

A structured problem is one where the three phases – intelligence, design and choice – are clearly structured. Hence, a number of algorithms (decision rules) exist for: specifying the problem addressed, generate candidate decisions and finding the ”best” solution. Thus, structured problems can be entirely solved by the computer. In semi-structured environments the problem definition is rather complicated compared to the structured case. Nevertheless, the definition of possible solutions and the search for optimality might be rather easy. Thus, a close interaction between decision maker – i.e. a human being – and the computer, with the implemented decision rules in the intelligence phase, is necessary. Hence, as displayed in figure 4.2, the preferences of the decision maker are taken into account in this process. The following design and choice phases are entirely solvable by the computer itself, with respect to the decision maker's preferences. In the unstructured case there are no decision rules that can be implemented, so that a system can only provide data access and visualization, but no decision support in the sense of providing the best solution according to the decision rules, which is illustrated in figure 4.1.

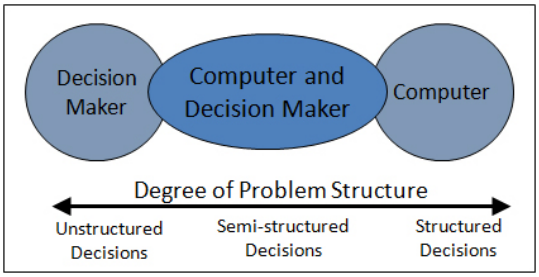


Figure 4.1: Degree of decision problems and intended problem solving environment, after Malczewski (1999).

Thus, Geovisual analytics has recently become a prominent field of research. In recent publications (e.g. Andrienko *et al.*, 2007) the topic and research agenda is layed down. There Geovisual Analytics for Spatial Decision Support tries to find ways to provide computer support to solve space and space-time related problems, by enhancing human capabilities to analyze, envision, reason and deliberate accordingly. This discipline is a subarea of Visual Analytics (Thomas and Cook, 2005). According to Andrienko *et al.* (2007) three criteria distinguish Visual Analytics from Geovisual Analytics for Spatial Decision Support:

- complex nature of spatial and temporal data
- a great number of actors involved with different roles
- tacit criteria and knowledge

These criteria are discussed in detail in Andrienko *et al.* (2007) – for details refer to this literature.

The decision process itself is complex and has been formalized by Malczewski (1999). There, a SDSS has to support a decision maker, that has to consider a great number of – often preclusive – criteria. Thus, the decision process is entitled as *Multi Criteria Decision Making (MCDM)*, see figure 4.2. It covers semi-structured problems, and incorporates the problem solving phases. Hence, the problem identification and definition is the first crucial thing, followed by a definition of the constraints, evaluation criteria and decision rules, which clearly are part of the intelligence phase. In addition, possible actions – here alternatives – are developed with the support of the computer, that are ranked utilizing the evaluation criteria and written to the Decision Matrix, which forms the end of the intelligence phase. The decision rules created in a prior stage are blended with the decision maker's preferences. With the help of this, the SDSS is able to select one or more "optimal" recommendations out of the alternatives. A sensitivity analysis completes the MCDM process by making a cross check of the evaluation criteria, the decision maker's preferences and the recommendations (Crosetto and Tarantola, 2001). There, the stability of the results is analyzed, which might have influence on the actual or future decision processes supported by the system. The results of the sensitivity analysis have direct impact on the preferences of the decision maker and the evaluation criteria. Due to the fact that the evaluation criteria have influence on the constraints and the alternatives generated by the system, the sensitivity analysis results also have an impact on the constraints and the alternatives itself, which is not displayed in figure 4.2. In order to create a SDSS, this framework should be implemented in order to guarantee a transparent and traceable decision process.

In order to get a picture of which functionalities a SDSS has to offer, see figure 4.3. There a number of basic functions and their according decision phases are displayed. Of course, the enumeration of functions could be extended. Figure 4.3 lists as first function an interactive and iterative problem solving environment, which is present for the user, and includes the possibility to input user preferences utilizing the user interface. As displayed in figure 4.3, the user interface allows the interaction with the system itself, including the user preference input, data visualization and mining, review and manipulation of evaluation criteria and constraints. The output of data and results such as maps is of great importance, due to the fact that spatial relations may be detected very easily, which is a link to Geovisual Analytics for Spatial Decision Support. Solely in the choice phase the selection of an appropriate mathematical model takes place in order to calculate the "best" alternative. Only if a number of models are available – like e.g. a choice

of a heuristic or a Branch and Cut solver for a problem – this functionality is of importance. Nevertheless, the user is able to decide how the solution is created – which can have influence on the solution itself – based on the user's knowledge. In addition, the user is able to select the alternative that is "best" in terms of the decision maker's preferences out of the solutions recommended by the system.

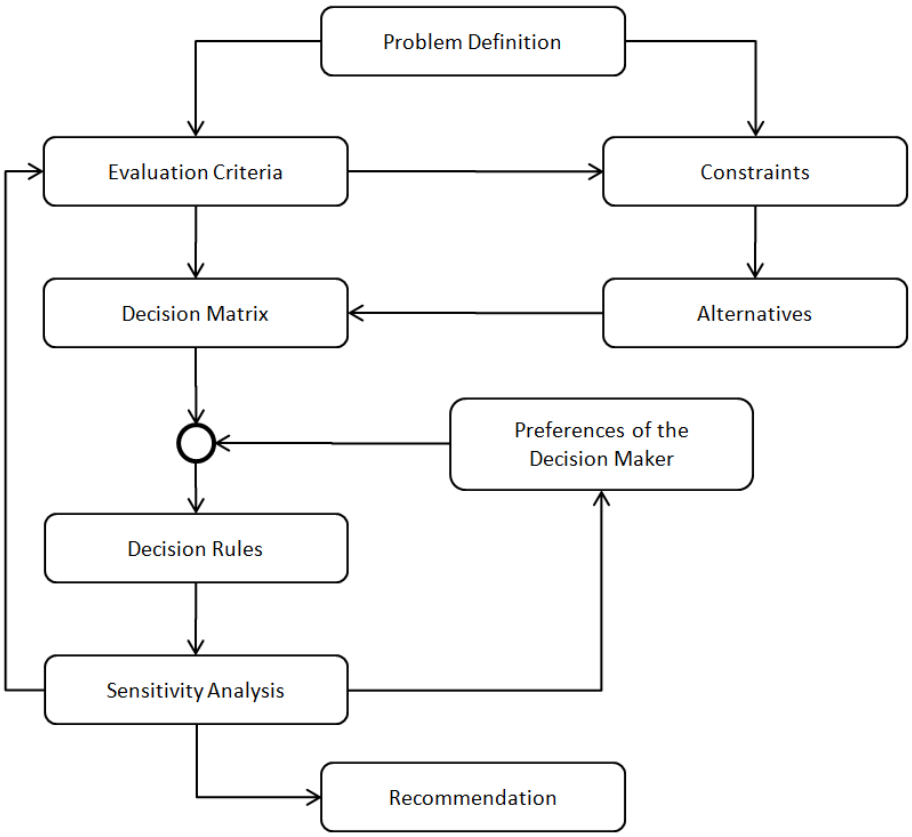


Figure 4.2: MCDM Framework, based on Malczewski (1999).

SDSSs are no single pieces of proprietary software. Due to the emergence of the Internet and SOAs, modern SDSSs have a modular architecture containing several components. Malczewski (1999) already identified the components of SDSSs that are as follows (see figure 4.4):

- Model Base (MB)
- Model Base Management System (MBMS)
- Spatial Database (SDB)
- Spatial Database Management System (SDBMS)

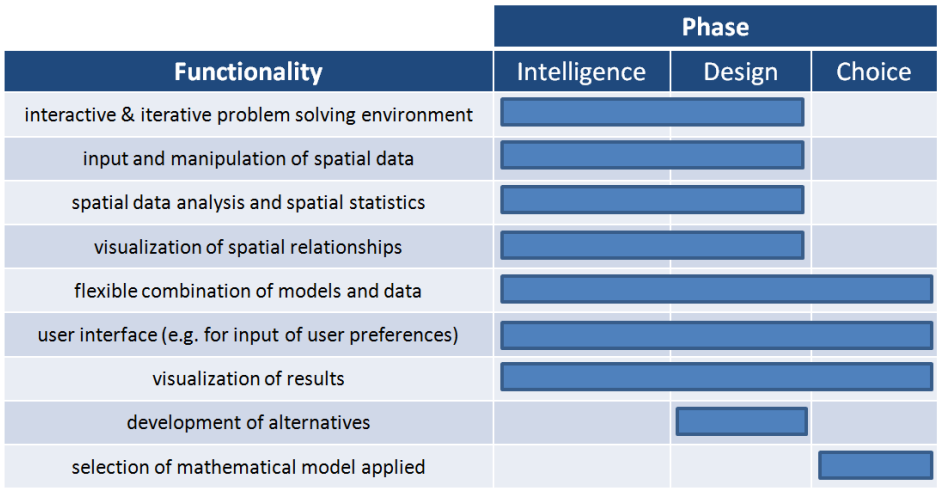


Figure 4.3: SDSS functionality and corresponding decision phases. Developed based on Herzig (2005), Makowski and Wierzbicki (2000), Malczewski (1999), Densham (1993).

- Dialog Generation and Management System (DGMS)

For reasons of completeness, a basic description of the components is given. For details refer to Malczewski (1999). The Model Base (MB) is a storage room for decision rules. In detail, the "knowledge" of the system resides there, providing the rules for: a) creating possible solutions and b) selecting the optimal solution out of the possible solutions generated before. Thus, it directly covers the decision support functionality of a SDSS. To ensure communication between the SDSS and the MB itself a Model Base Management System (MBMS) is employed.

A Spatial Database (SDB) in combination with a SDBMS is capable of storage, manipulation and retrieval of spatial-temporal data, as mentioned in chapter 2. As SOA are gaining popularity – also through the definition and application of standards in the field of GI (see chapter 2.1) – SDB and SDBMS do not necessarily have to be part of the system itself. Relevant data can also be retrieved live from data providers utilizing standards. In addition, frameworks like INSPIRE will support such architectures. Nevertheless, SDBs are relevant because spatial data – e.g. at the data provider – have to be stored in SDBs.

The Dialog Generation and Management System (DGMS) is a "collection" of User Interfaces (UIs) that offer the capabilities to interact with the system directly. With this, it is possible to alter model parameter or calibrate the system. In SOAs the DGMS is constantly loosing importance due to the fact that the SDSS functionalities are packed into Web Services. Thus, every client that is interacting with the SDSS via services can have

highly customized UIs – and no ”standard” UI. This user centered view, results in a number of UIs for each use case that has to be defined. Thus, not every user (e.g. truck driver) has the full set of SDSS-functionality on board of his device, which increases the usability for non-experts. Moreover, this approach enhances the flexibility of the whole system, by no strict dependencies between UI and SDBMS. This allows a change of e.g. DB vendor, without a ”collapse” of the whole system!

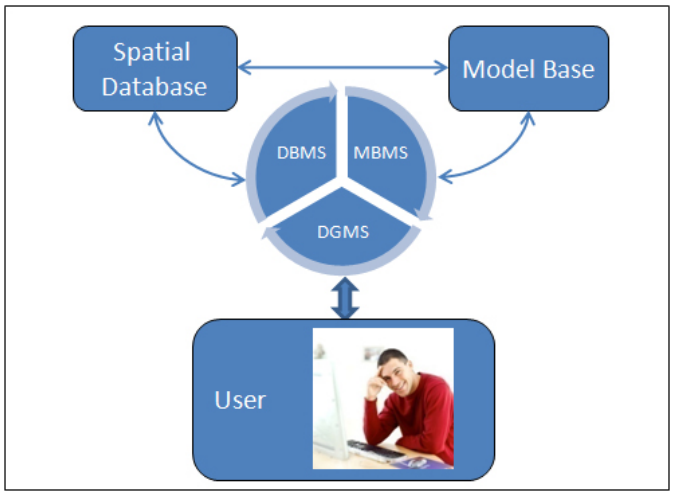


Figure 4.4: Components of an SDSS, based on Malczewski (1999, 1997).

Sprague (1980) identifies three technical levels of an SDSS framework, that have great influence even on today's papers in this field of research (see figure 4.5). The framework distinguishes between:

- Specific DSS
- DSS Generator
- DSS Tools

The specific DSS is the piece that does the ”whole” work and serves as central interface between the user and the system. Hence, it is bound to a specific problem type, and capable of providing decision support just for this special case. A DSS generator represents a packed software – i.e. components – with defined functionality that allows the quick development of specific DSS based on this components. Examples are GIS, standard DB/DBMS or geostatistical tools like R – a project for statistical computing (The R Foundation for Statistical Computing, 2009). The fundamental technological level of DSS are DSS tools. They comprise very basic software components that support the creation of DSS generators as well as specific

DSS. Examples are programming languages or application communication software like ODBC. It is important to mention that specific DSSs can be formed using either DSS tools directly, or DSS generators that rely on DSS tools (see figure 4.5). Although it is difficult to generate reusable DSS generators, they prove to be efficient in developing new specific DSS by simply "exporting" technology into a specific DSS.

Ostländer (2004) transferred the classical approach of Sprague (1980) into a new Spatial Data Infrastructure (SDI) approach utilizing SOA (see figure 4.6). There the classical desktop approach of a specific DSS is transferred into Web-based specific DSSs that represent a service chain and workflow management service for any specific problem. The user interacts with the system by parameters that are exchanged using web technologies. In this contemporary approach the classical SDSS tools are represented by geoprocessing services – as described in 2.2, see OGC Web Services Phase 4 and 5. The SDSS generator, that chains certain SDSS tools and bundles them to form a coherent piece of software, has its analogy in the SOA approach – called service chaining. This component chains a number of services to define a new "compound" service that may be used in any specific SDSS service. The services have to offer decision rules and the ability to build and save service chains accordingly. The analogy to specific DSSs in the classical desktop approach are the specific SDSS services. They are services for a well defined spatial problem. Thus, they act like a classical specific SDSSsystem as defined in Spragues specific DSS – but are fully integrated in a SOA.

The DSS tools get the prefix "Web-based" in Ostländer's approach. A general overview of Web-based DSS is given in Bhargava and Power (2001) and Bhargava *et al.* (2007). Web-based DSSs serve as information management and geoprocessing services, that store, process and apply decision rules to selected spatial data. Some of the results of Ostländer (2004) are a current research topic in OGC (OGC, 2009a,b,c). Nevertheless, it is necessary to find a connection between the "classical" DSS approach and a recent approach utilizing SOA, which is important for this thesis.

4.2 SDSS AND GISc

Given the description of SDSS in chapter 4.1, one might ask what is the difference between GIS and SDSS, and which role does GISc play in a SDSS. Concerning the first question, the answer is definitely difficult, due to the fact that both systems enable the user to analyze spatial problems utilizing spatial data with appropriate methods. The exploration of this question will be part of this chapter, followed by an explanation of where GISc methodology is necessary in SDSS.

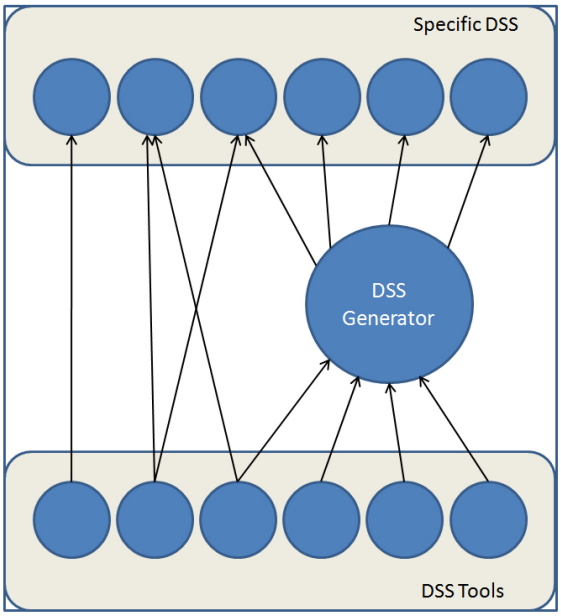


Figure 4.5: Technical levels in an DSS Framework, based on Sprague (1980).

The genesis of GIS since the 1960's – which is described in Bartelme (2005), Burrough and McDonnell (1998) and Coppock and Rhind (1993) – shows that GISs were developed based on the need to have a modern tool for spatial management. Back then, the main task was the cartographic visualization of space and land use/land cover. Data input was a vital part, due to the fact that spatial data were rare. In the course of the following enhancements until of GIS the 1990's a number of methodology and technology was integrated, including: data input and manipulation, advanced query techniques, (basic) spatial statistics, network analysis and topology. The techniques mentioned are still the core of modern GISs.

Through the integration of advanced mathematical capabilities – e.g. Monte Carlo simulations, Fuzzy Sets – as well as modeling capabilities – like application specific add-ons for e.g. waste water management – GIS evolved further and reached a broad market impact. Herzig (2005) argues that DSS functionalities – including all the relevant theory and methods – can be added to an existing GIS in order to develop a SDSS. Crossland *et al.* (1995) mention in relation to Cooke (1992) that a SDSS is a tool that "*solves specific problems efficiently, and delivers immediate results*", and heavily relies on GISc. In addition, GISs may be seen as a sophisticated toolbox that is designed to form a solid basis for a great number of different applications, where SDSSs are one of them. By adding "SDSS tools" several criticisms of GIS are addressed, that are mentioned in Chakhar and Martel (2003):

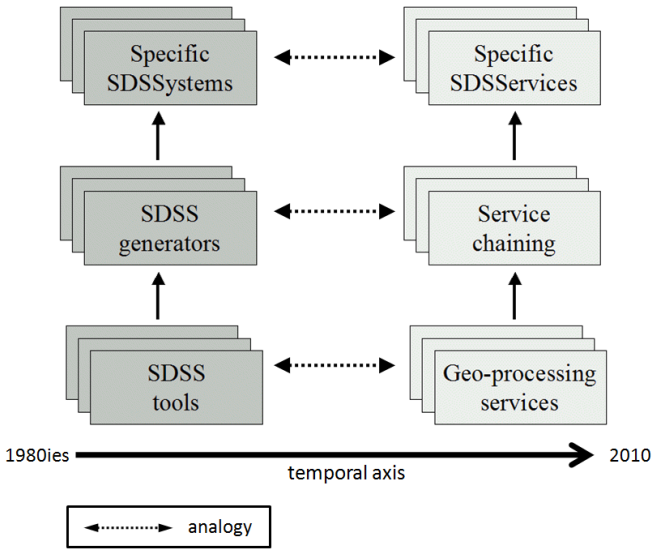


Figure 4.6: Analogy between the technical DSS framework proposed by Sprague (1980) and the analogue counterparts of a recent SDSS approach using SOA. Based on Rinner (2003). The classical desktop approach defined by (Sprague, 1980) on the left side is replaced by a contemporary approach on the right side that utilizes SOA.

- lack of incorporation of decision maker's preferences in traditional GIS
- limited analysis of multi-criteria problems, where the criteria are conflicting
- lack in scenario creation, simulation and comparison in recent GIS
- analytical functions are rather tailored towards data management than effective to analysis

Consequently, following the argumentation of Chakhar and Martel (2003), a GIS is no SDSS per se. Through the integration of specific models and components in GIS a SDSS can be created. Due to the fact that a SDSS has to provide support for the whole decision process (see chapter 4.1), Herzig (2005) argues that a GIS has to be enriched by two components: modeling and decision support (Denzer, 2002; Malczewski, 1999; Fedra, 1996; Jankowski, 1995). The degree of integration of the components will be subject of section 4.3.

Taking the findings of the prior paragraphs into account, GISc is a vital part of a SDSS. Thus, it is necessary to find out in which decision phase and to which extent GISc can be integrated. This thesis relies on the results of Malczewski (1999, 1997). In the intelligence phase GISc helps to analyze the situation. The abilities of GIS to integrate, visualize, manipulate and

analyze spatial data are beneficial in this stage. Nevertheless through GISc and GIT it is possible to come up with a coherent "picture" of the problem due to the fact that data can be imported from various data sources or integrated utilizing services (c.f. sections 2.1 and 2.2). In addition, GIT is able to serve complex spatial information to citizens by means of e.g. Web Mapping Services (WMSs). Thus, the public is able to participate in the decision making process by making their annotations directly on a map. This approach is commonly known as Public Participatory GIS (PPGIS), and has been recently covered by literature (e.g. Drew, 2003; Jankowski and Nyerges, 2003; Howard, 1999). In the design phase GISs play only a minor role, due to their shortcomings described in the preceding paragraph. Especially, the lack of generating and evaluating possible solutions/alternatives is one of the major drawbacks of recent GIS. Thus, a separated modeling module, capable of generating and evaluating alternatives is inevitable. Nevertheless, basic GIS functionalities like overlay, map algebra (Tomlin, 1990), connectivity or proximity, should be used as basis by the modeling modules. In the "technology-provider" role a GIS plays an important part when it comes to developing modeling parts of an SDSS. Even in the choice phase GISs play only a minor role due to the lack of an interactive incorporation of decision makers preferences with respect to predefined evaluation criteria. Although GIS is necessary for e.g. visualization, data manipulation tasks, every other system could not handle.

In chapter 2.2 and 4.1 the impacts of SOAs on GISc, and especially on SDSS are discussed. There the papers of Ostländer (2004) and Rinner (2003) are mentioned that give detailed insight in SDSS within SOAs, which are described in chapter 2.

4.3 Integration of Optimization Techniques

To develop a SDSS the MBMS has to be incorporated – which is represented by the optimization techniques and methods, that are described in chapter 3. In this thesis mathematical models that create and rate alternatives are "hidden" behind the optimization tools with according mathematical models. In literature, a number of approaches to integrate optimization methods in an SDSS exist. Malczewski (2006) mentions a basic classification of an integration: a) the extent and b) the direction of integration. Although the integration issues discussed in literature (Malczewski, 2006; Ungerer and Goodchild, 2002; Jun, 2000; Fedra, 1996; Jankowski, 1995; Nyerges, 1992) cover the integration of GIS and DSS, the theory covered in this resources is extended towards the incorporation of optimization methods in a SDSS environment.

First, the extent of integration is covered. This characteristic is crucial for the communication of the components, which will be described in the following chapter. Nevertheless, the integration has direct influence on the implementation of an SDSS. Malczewski (2006) mentions four categories of integration:

- no integration or isolated components (according to Ungerer and Goodchild (2002))
- loose coupling
- tight coupling
- full integration

If the components are *not integrated* there is no way to exchange data between the different systems, which is hardly ever the case. Denzer (2002) argues that this *null integration strategy* is tolerable in science but is generally not acceptable for other user groups. The *loose coupling* approach allows the systems to exchange information in a file based manner, utilizing predefined formats (see figure 4.7(a)). The interacting components have to be able to read and write the exchange format. In this approach the user has to do the data exchange "work" and handle the programs that "form" the SDSS – which is not present as a homogeneous piece of software and is rather a collection of software packages. Such systems can be created very fast, because of the little amount of extra coding work. The user pays for the coder's convenience by time consuming data import and export and a not intuitive handling of the system. *Tight coupling* results in a combination of different systems with an automated data exchange (see figure 4.7(b)). Clearly, there are a number of different software packages, or components like *.dll's, that provide a special functionality. In order to ensure a simple user interaction there is only one single user interface that allows the control of all software components from one single place, which simplifies user interaction and acceptance of such a system even by non-experts. Nevertheless, it takes more resources to develop a SDSS with tight coupling approach. To create such a system it is necessary to define the data exchange process and to generate a user interface that is able to communicate with the system components. Hence, it is possible to store relevant data of the SDSS in a single DB, which serves as central data exchange platform. A SDSS with a *full integration* approach does not consist of different components that exchange data, and all functionality is delivered by one single application (see figure 4.7(c)). To create a 'single' application, user-defined routines are necessary that are written in any programming language. In order to make a re-use of existing applications they can be integrated by utilizing the Application

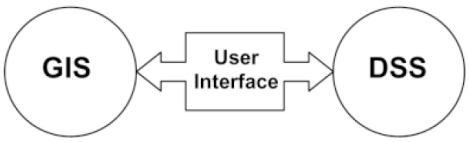
Programming Interface (API). For the user's eyes there is just one single piece of software that has all the functionality of the SDSS.

Secondly, concerning the direction of integration and communication Anselin *et al.* (1993) distinguish three different scenarios:

- one-directional integration (see figure 4.8(a))
- two-directional integration (see figure 4.8(b))
- dynamic integration (see figure 4.8(c))

One-directional integration results in a data flow in only one direction. E.g. data generated in the GIS module may serve as input data for the DSS module in the following process. The two-direction approach involves a more interactive data exchange. There data are transmitted from e.g. a GIS to the DSS module and backwards as needed. Dynamic integration enables data to be exchanged between modules desired by the user or the system. Thus, a user interface is necessary that allows the interaction with both systems. This allows the generation of flexible SDSSs for e.g. iterative analysis steps within a SDSS.

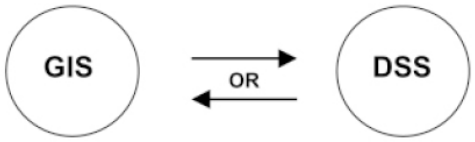
(a) loose coupling approach



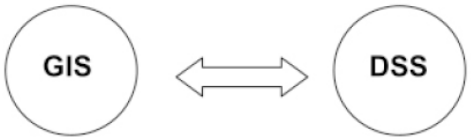
(b) tight coupling approach

(c) full integration approach

Figure 4.7: Integration of GIS in a SDSS – distinguishing the extent of integration (Jun, 2000).



(a) one-directional integration



(b) two-directional integration

(c) dynamic integration

Figure 4.8: Integration of GIS in a SDSS – distinguishing the direction (Jun, 2000).

4.4 Conclusion

SDSSs are of vital importance for optimizing the WSC in general. Although it might seem reasonable to use Spatial Expert Systems (SESs) instead of SDSS, the latter is used due to the flexibility and possibility of user interaction and selection of the chosen alternative. In addition, the problem of WSC optimization is still regarded as semi-structured problem because of the number of external influences on this process (e.g. an immediate snow storm). Thus, a logistics operator should have the ability to overrule the computers decisions based on real-time data – which is a fundamental functionality of a SDSS. Nevertheless, a SDSS with an appropriate modeling system is inevitable in optimizing vehicle movements in both – long term and short term horizons (see section 1.1). By the application of SOAs in SDSS the real-time optimization system supporting WSC management can be created as "Web-based specific SDSService". SOAs are of major importance because a complete new implementation of all features to generate a "desktop specific SDSSystem" would be very complex and time consuming – a re-use of existing components is a far more economic way. In order to ensure communication between the service oriented components SOAP and RESTful approaches are applied.

Part II

Application and Results

Chapter 5

System Architecture and Implementation of WSC Prototype

This chapter is intended to discuss the implementation in detail. The discussed topics rely on the theory of the previous chapters, and reference to relevant content where necessary. In addition, the differences between theory and the actual realization are described. Especially the optimization itself, which forms the core module, will be highlighted.

5.1 Overview of System Architecture

The system that optimizes the WSC is divided into several modules that are working independently, and share a common data basis – see figure 5.1. The central database serves as data pool where all modules receive and deliver information using services. The modules are explained in detail in the following chapters. Here an overview that describes how the modules are "sticked" together is given followed by an explanation how the exchange of data works.

The system consists of a number of modules that fulfill a specified task. These modules are based on the components of Spatial Decision Support Systems (SDSSs) as described in chapter 4.1 and are connected using Service Oriented Architectures (SOAs). The modules are as follows:

- The **Database Management System (DBMS)** (Mod_DBMS) saves all information and serves as central data exchange platform. This module is equivalent to the SDSS components Spatial Database and Spatial Database Management System (SDBMS). For details refer to chapter 5.2.
- The **Tracking Engine** (Mod_Tracking) takes the delivered position data from mobile devices and stores the data in the Database (DB). For details refer to chapter 5.3.
- The **Web Mapping Engine** (Mod_WME) is the central visualization engine. It serves maps for the Mobile Devices and the Desktop

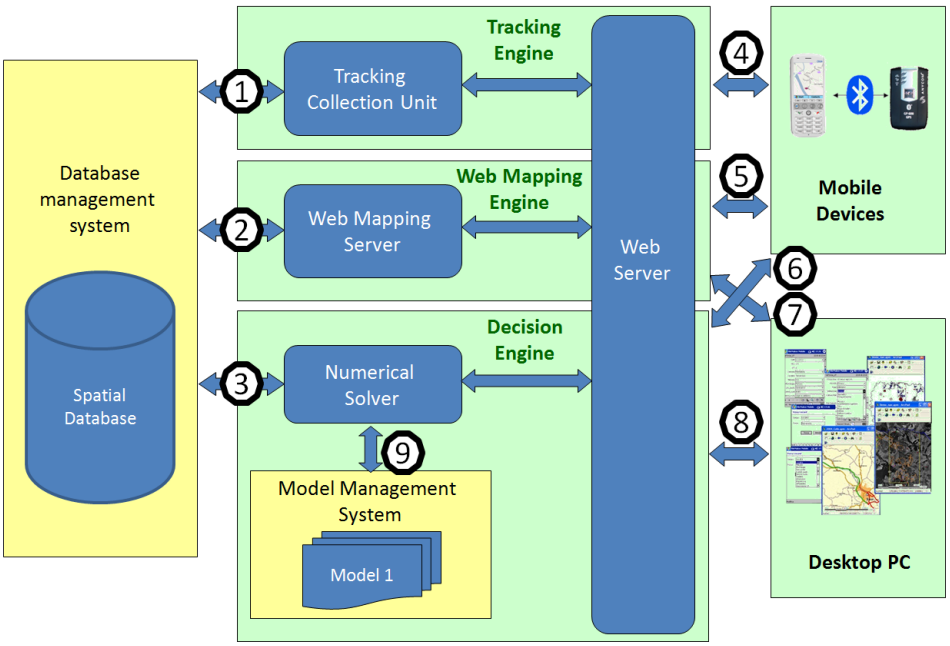


Figure 5.1: Overview of the system architecture for optimizing the WSC. The arrows are indicating the connections between the modules – each connection with a number is explicitly explained in the text of section 5.1.

Engine. This is roughly a part of the Dialog Generation and Management System as it produces vital parts of the User Interface (UI) itself. By using standardized Services (OGC® Web Mapping Service (WMS) and Web Feature Service (WFS)) it is possible to come up with an interoperable system in terms of Geographic Information Science and Technology (GIS&T). For details refer to chapter 5.4.

- The **Decision Engine** (Mod_DE) is able to model the Wood Supply Chain (WSC) accordingly – as Vehicle Routing Problem with Pickup and Delivery and Time Windows (VRPPDTW) – and solve it using an extended Adaptive Large Neighborhood Search (ALNS) algorithm described in chapter 3.4.4 and 5.5.5. In addition, to the heuristic solution method also other solution models or altered model parameters can be used. Thus, this module is equivalent to the Model Base and the Model Base Management System. Details can be found in chapter 5.5.

- The **Mobile Device(s)** (Mod_Mobile) are handheld devices capable of establishing mobile internet connections, having a sufficient large screen for displaying UIs and are capable of determining their position (by e.g. Global Positioning System (GPS)). In addition they have to be ”programmable”. Hence, Personal Digital Assistants (PDAs)

or smartphones (with built-in GPS/with external GPS) fulfill these requirements. Details are in chapter 5.6.

- The **Desktop System** (Mod_Desktop) is a simple system for setting the model parameters, and monitoring the whole system (vehicles, truckloads, etc.). It is operated by the staff of a logistics center. A detailed description is in chapter 5.7.

As mentioned earlier, the connections between the modules utilize SOA (see 2.2). Generally speaking, all three types of service approaches are used: Web Service (WS), Simple Object Access Protocol (SOAP) and Representational State Transfer (REST). Explicitly this thesis is restricted to SOAP and REST, but as e.g. WFS and WMS are prominent representatives of WS there is no excuse to "escape" WS. Nevertheless, WSs are restricted to the necessary International Organization for Standardization (ISO) and Open Geospatial Consortium (OGC) services, and all other web services rely on SOAP or REST. In order to go into detail the following "connections" can be identified, which are marked in figure 5.1 with the according number:

1. Mod_Tracking – Mod_DBMS: This connection relies on a SOAP web service. The functionality of this service is, that it takes the position information provided by the Mod_Mobile and stores it in the spatial database. The following parameters are transmitted by the vehicles, which are subsequently processed by Mod_Tracking: Position (in EPSG:4326 – lat/lon WGS84), vehicle number, date and time of the gathered position. In order to support position updates of the vehicles (i.e. where the Mod_Mobile are located on) which have no direct connection (e.g. no internet due to a very remote location) the positions could be locally stored on the mobile devices and then sent as soon as a connection is available. The service returns a message if the data was successfully stored in the DB.

2. Mod_WME – Mod_DBMS: The connection of the Web Mapping Engine and the SDBMS is no "real" web service as it uses a direct connection that is provided by OGR implemented in the web mapping software – that could be UMN Mapserver, Geoserver or any other mapserver.

3. Mod_DE – Mod_DBMS: These two modules are connected via a REST web service, due to the fact that there are a number of stateless queries against the database. Among them are e.g. lengths of shortest paths between two vertices or queries to fill the data model. Due to these requests with low data volume, it is advisable to use the REST approach in order to get rid of the SOAP data overload. In addition, very small

data packages speed up the transmission process and, thus speed up the optimization as a whole, especially where a great number of such simple queries is necessary.

4. Mod_Mobile – Mod_Tracking: They are connected by a SOAP protocol that is capable of reading the data from the mobile devices and returning if the transmission was successful. The data that this service receives are: Position (in European Petroleum Survey Group (EPSG):4326 – lat/lon WGS84), vehicle number, vehicle status (e.g.: Driving, Pickup, Delivery, Breakdown), date and time of the gathered position. Date and time are necessary in order to support position updates that are not carried out in real time, due to internet connection problems – as mentioned in the connection description of Mod_Tracking – Mod_DBMS.

5. Mod_Mobile – Mod_WME: The connection between these models is realized using a classical WS approach. A read/write connection utilizing WMS and WFS(-T) – see chapter 2.1 for details – provides the delivery of maps for mobile devices as well as the editing of geographical features. These services rely on the implementations of web mapping servers that are used here.

6. Mod_Mobile – Mod_DE: This connection is created in the form of a SOAP service. A number of data are transmitted from the mobile devices to the Decision Engine that concern the actual progress of the timber transport process. Among them are: the actual node that the vehicle is driving to, or where actually timber is picked up or delivered, a successfully processed node or if a node could not be reached. In addition, the data of the actual route are sent to the mobile device on demand, including the data of which timber to pick up (quantity and quality). The data sent by the mobile devices are directly stored in the database after a proof of plausibility. In addition, new timber piles can be inserted into the system using this connection. There the relevant data are transmitted that include: position of the pile, start date, due date, sub-piles with according timber qualities (i.e. assortments), quantities and the date of availability.

7. Mod_Desktop – Mod_WME: The connection is like connection nr. 5 (Mod_Mobile – Mod_WME), a classical WS. Similarly to the connection between Mod_Mobile and Mod_WME, this one relies on ISO and OGC services – especially WMS and WFS – in order to display the situation of trucks and timber (delivery as well as pickup).

8. Mod_Desktop – Mod_DE: The desktop is able to alter model parameters, input new or manipulate existing trucks, haulage companies or

saw mills as they desire. Thus, the system is able to communicate this using SOAP services.

9. Numerical Solver – Model Management System (i.e. mathematical models): The connection between the numerical solver and the mathematical models is dependent on the solver used. Using a standard numerical solver inside a solver system like MPL (Maximal Software Inc., 2009) has the opportunity that models can be stored in simple text files that can be exchanged as desired. Especially using e.g. the Optimax component library (Maximal Software Inc., 2009) or AMPL (Fourer *et al.*, 2003) it is very easy to exchange the optimization model. Regarding the optimization model used in this thesis – ALNS –, there is a number of model parameters which alter the model strongly.

The integration of modules containing Geographic Information (GI) technology and Mod_DE can be described as *dynamic tight coupling* approach, as defined in chapter 4.3. This integration strategy is realized in the combination of GI and DSS modules which can be controlled with one user interface. These modules dynamically exchange data in both directions utilizing (standardized) services, as desired by the user or needed by the system.

As described earlier, the system architecture supports every module being placed on geographically dispersed machines. Thus, the database could be a database cluster, which is suitable for safety reasons. The machines in the cluster communicate via web services. Mod_DE could be any numeric solver, which is upgraded to process the data coming from Mod_Mobile. In order to communicate, the numeric solver must be reachable via web services. Due to the fact that most numerical solvers do not provide communication via web services (except for COIN-OR), they have to be enveloped within services by hand.

A typical workflow of the system is described in the following paragraph. First, different users working with the system are distinguished, according to the stakeholders of the WSC (see chapter 1.2 and figure 1.3). Secondly the workflow itself follows.

- **Saw mills**: They are using the system to put in their timber demand. This includes the input of quantity, quality, the time frame when the timber should be delivered as well as a location where to transport the timber.
- **Haulage companies**: These stakeholders mainly input their vehicles, the corresponding data like working times per day, workdays during the week, the capacity and the length constraints of the transported timber. The location of the depots – in the sense of VRPs, i.e. where

the vehicles start and end their tour – has to be defined too. In addition, the vehicles of each haulage company alter the timber quantities of a pile after picking up timber.

- **Forest enterprises**: This stakeholder provides data on newly or recently harvested timber piles. Here, the following data are transmitted using LBS and mobile devices: position of the timber pile, start date (when they are ready for transportation), due date (denotes the date when the pile has to be completely "removed"), as well as quality and quantity of the timber in the piles.
- **System administrator**: Administrators are able to alter system and/or mathematical model parameters according to changing surrounding conditions. E.g. the system administrator can change parameters that alter the "speed" of timber pick up, which indicates that at "high speed" timber is transported to the saw mills on the fastest way – using all resources available.

The typical – but though very simplified – workflow of the system looks as follows:

1. Input/altering of **vehicle data**, which is a continuous process.
2. Input of **timber demand**, which is also an ongoing process, resulting in demand/delivery profiles.
3. Input of **timber supply** – i.e. timber piles, which is an ongoing process.
4. **Create a Vehicle Routing Problem (VRP)** according to chapter 3.2.
5. **Optimize VRP** according to chapter 3 using ALNS.
6. Every truck has access to assignments using Mod_Mobile and **picks up/delivers timber according to the assignment**s calculated by the optimization phase before.
7. **Update** of **timber pile quantities** as well as **demand quantities** due to picked up and delivered timber.
8. **Track vehicles position and status** constantly.

Points 4 and 5 are processes that are started each night, when no vehicles are driving. They are calculating a solution for the VRP according to the Rolling Schedule Approach (RSA) discussed in chapter 3.5.

5.2 Database Management System

A DBMS is a software system to store, manage and retrieve data. The findings of Codd (1982) identified that mathematics is crucial for database technology, due to the introduced approach of relations to databases – the database terminus technicus is rephrased to Relational Database Management System (RDBMS). All connections of data records are represented as relations, these functions can be defined based on the relations in order to retrieve a well defined subset of data. For the case of WSC optimization RDBMS are used that are capable of working with spatial data – the term Spatial Relational Database Management System (SRDBMS) is created which is mostly denoted as SDBMS. For that purpose standards have to be implemented in the SRDBMS, specially the OpenGIS® Simple Features Interface Standard (SFS) (see chapter 2.1.3) and ISO 19125 Geographic Information – Simple Feature Access – Part2: SQL Option (see chapter 2.1.4).

To create and visualize a database model, which represents the "Universe of Discourse", a Entity Relationship Diagram (ERD) is appropriate (IEEE, 1998; Chen, 1976). The ERD contains all relevant entities, attributes and relations. The ERD for WSC optimization is shown in figure 5.3, making use of the IDEF1X notation (IEEE, 1998). This notation is capable of visualizing relations and attributes in a user friendly way. In addition, the cardinality of relations as well as the relationship type (identifying or non-identifying) can be depicted very easy. Non-identifying relations are visualized with a slashed line and identifying relations with a full line. In order to depict "optional" relations – meaning $0 \ldots 1{:}0 \ldots n$ – the relationship sign used in figure 5.4 is used.

The data model itself is capable of storing and analyzing spatio-temporal data, which is of particular interest for WSC optimization. Spatio-temporal data are "produced" by the following entities: vehicles and the according transport routes (either planned or done), timber piles as well as timber demand. The spatial data in this SDBMS are in CRS UTM-33N – EPSG:32633. In order to clarify the connections between relevant stakeholders modeled in the ERD, figure 5.2 visualizes them accordingly. There the stakeholder saw mill is explicitly denoted that defines a certain timber demand. The forest enterprises, which are not mentioned in figure 5.2, produce timber supply that is transported to the saw mills by trucks of haulage companies. The vehicles of the haulage companies are necessary to form planned and finished transports (i.e. transport done), whereas the transports planned represents the solution of the VRP that is solved utilizing ALNS in the Mod_DE. The vehicles submit the actual position to the according table truck_geom and the "real" transport of timber is stored in the transport_done group.

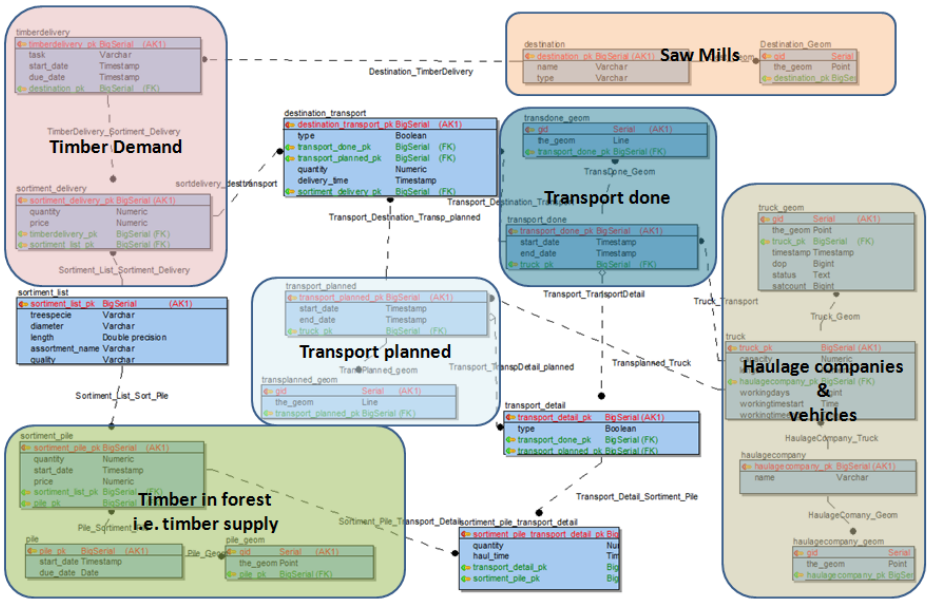


Figure 5.2: ERD with entity groups according to a subgroup of the identified stakeholders of the WSC.

In the following paragraphs the entities shown in the ERD (see figure 5.3) are described thoroughly including the attributes and the relations between them. First, the timber supply is the starting point, followed by timber demand and saw mills, haulage companies and transport tables, ended by the ”connecting” tables that cannot be assigned to a specific group. The tables connected to the ones holding spatial information – denoted with suffix ”_geom” – could be incorporated into the father table (e.g. pile and pile_geom), due to a 1:1 relation. This is done in order to have very flat spatial tables which are faster to transmit over the internet, e.g. for web mapping purposes. Additional information should be requested as needed and not sent beforehand. This concept is to be found in the all entities holding spatial data throughout the ERD.

The table **pile** holds relevant information of the piles itself and a start and due date indicating when a pile is available for pickup and to which date it should be removed completely. Due to the fact that every pile has a position this can be modeled with a separate table **pile_geom** that solely holds the spatial data for each pile – i.e. a point. Due to the fact that each pile can consist of at least one timber quality it is assumed that this can be modeled by table **sortiment_pile**. In case of more timber qualities per pile they are stored in this table. Thus, there is only one vertex for one or more timber qualities and associated quantities, which seems quite unusual. This approach is feasible though, because timber of different quality is usually

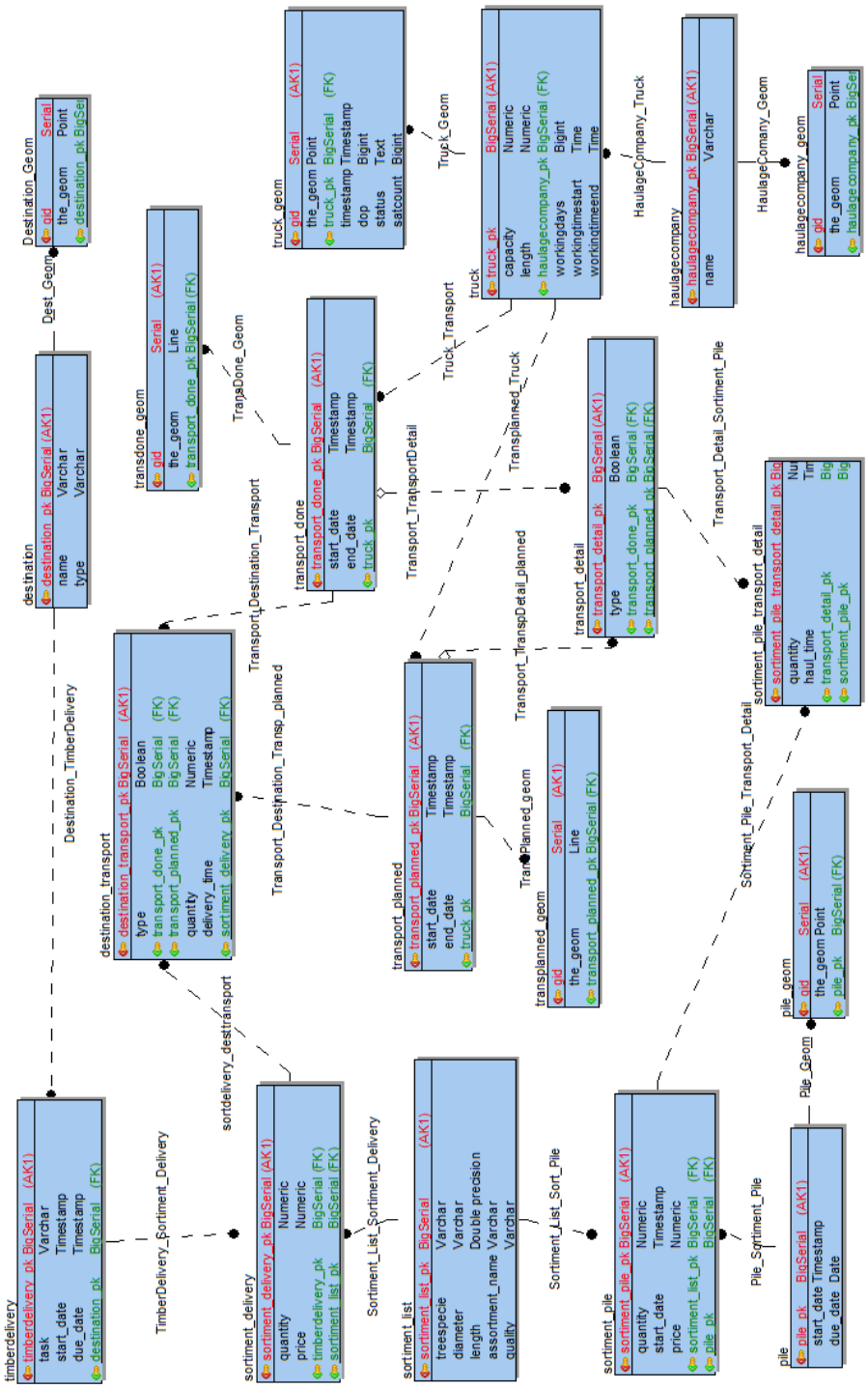


Figure 5.3: ERD in the IDEF1X Notation.

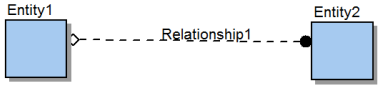


Figure 5.4: Vizualizing $0 \ldots 1{:}0 \ldots n$ relations in ERD using IDEF1X notation.

piled up very close to other timber qualities of the same pile. In addition, to the quality the attributes of sortiment_pile are start_date which indicates the date when the assortment is ready for transportation, price represents the price per cubic meter timber the forest enterprise wants to achieve. The parameter quantity shows the actual quantity that is initially present in this pile. It is not updated, due to the fact that every pick up operation can be calculated using the group transport_done. The data for the parameter quantity could – besides the possibility of an input by the forest enterprise – originate from harvester log files that are transmitted to the central database using services.

The tables of the group timber demand and saw mills represent the demand side of the WSC. Thus, the tables **destination** and **destination_geom** store the relevant data of saw mills, analogue to the tables pile and pile_geom. Nevertheless, every timber demand is modeled by two tables: **timberdelivery** and **sortiment_delivery**. Table timberdelivery includes the long term demand profiles that are valid over e.g. a quarter of a year. Thus, they have a start and end date indicating the beginning and the end of possible delivery operations for this entity. For sure, every saw mill can have more such profiles – even parallel! Each demand profile has to be filled up with precise information on which timber qualities and according quantities are needed – in table sortiment_delivery. In addition, the attribute price denotes the price the saw mill wants to pay for a cubic meter of wood.

The haulage companies and corresponding vehicles are modeled using the tables: **haulagecompany**, **haulagecompany_geom**, **truck** and **truck_geom**. The tables haulagecompany and haulagecompany_geom include relevant data on the companies, including their positions which serve as depots in the VRP – the location where the vehicles start and end their tour each day (see chapter 3.2 for details). There is a 1:1 connection between haulagecompany and the corresponding spatial table, as discussed in the preceding paragraphs. Each truck in the table truck is connected to one haulagecompany. In this table several data are stored that are of interest for the optimization process itself:

- *capacity* of the vehicle in cubic meter
- *length* of timber that can be transported in meter

- *workingdays* denoting the days of the week on which the truck can be used as an integer – e.g. 12345: meaning that the vehicle is used on Mondays until Fridays
- *workingtimestart* indicating the time the vehicle will start from the depot
- *workingtimeend* the time the vehicle has to be at the depot at latest – which might be connected to workingtimestart in conjunction with legal working time regulations, which are not subject of this thesis.

The actual position – delivered as a point – with a sending time stamp of each truck is stored in table **truck_geom** with some additional data on the quality of the GPS signal like: *DOP*, *satcount* – indicating the number of satellites in view and used for position calculation. The attribute *status* is used to store messages of the truck driver – e.g. if he is actually picking up/delivering timber or if he has a severe breakdown.

The transports themselves can be separated in two general categories: planned transports and done transports – which is represented in figure 5.2. The group planned transports consists of the tables **transport_planned** and **transplanned_geom** which stores the spatial data of the planned transports – i.e. a line. The transport_planned has two attributes indicating the start and end time of the tour for a particular vehicle. The entity group transport done consists of two tables: **transport_done** and **transdone_geom**. Transport_done is similar to transport_planned, and just holds the data on start and end of the tour for a certain vehicle. Transdone_geom stores the spatial information of tours as lines. These lines are updated each time a vehicle, which has to do a transport, sends new position data. Thus, the position is added as new point to the line.

The connecting tables, which cannot be clearly assigned to a certain group, serve as connection between the entity groups. The entities are described in the following list:

- **sortiment_list**: This table holds data on the different assortments (i.e. timber qualities) available in the system. Each assortment has a defined *treespecie*, *length*, *diameter* and a *quality* – indicating the quality class e.g. A, B, C, C*. All this data have to be defined according to Kooperationsplattform Forst Holz Papier (2006).
- **transport_detail**: This table defines the connection between the tables transport_planned/transport_done and the table sortiment_pile_-transport_detail. The attribute *type* indicates if a certain record of sortiment_pile_transport_detail is a planned or a transport that is done

– i.e. it is set to true for done transports. In addition, the two foreign keys to transport_planned and transport_done are not mandatory. Either one of the two has to be set.

- **sortiment_pile_transport_detail**: This table models the picked up timber from the piles. It contains the attributes *quantity* – which indicates the quantity to be picked up, *haul_time*, which denotes the arrival time at a specific vertex (see chapter 3.2 for details). Thus, both transport types (planned and done) are possible records of this table. In order to differentiate between planned and done transports the table transport_detail is necessary. The table sortiment_pile_transport_-detail monitors the quantity of timber present at a certain pile, i.e. quantity of a specific quality of a pile.

- **destination_transport**: This entity holds the data on the delivered timber either from planned and done transports. Thus, if the attribute *type* is true this indicates that the record in this table is related to a done transport, otherwise it is related to a planned transport. In addition, this table has the attributes *quantity* – representing the quantity that is delivered to a destination (for fulfilling a specific profile using the foreign key *sortiment_delivery_pk*) by a specific transport. The attribute *delivery_time* indicates the arrival time at the delivery vertex as described in chapter 3.2. The foreign keys *transport_done_pk* and *transport_planned_pk* are optional attributes, but either one of the two attributes has to be set – similar to the table transport_detail mentioned above.

The development of the ERD was done in TOAD Data Modeler (Quest Software) due to its capability to export Data Definition Language, which are SQL statements to create the database. They can be specially tailored to the needs of a specific database (e.g. PostgreSQL 8.2 or Oracle 10g). The implementation of the ERD is done utilizing PostgreSQL/PostGIS 8.2, due to the fact that PostGIS offers full support of OpenGIS® and ISO 19125. In addition, it possible to combine this specific SDBMS with other software from the GIS&T sector e.g. Mod_WME, which will be subject to chapter 5.4.

5.3 Tracking Engine

The tracking engine is of particular interest for the Mod_Mobile, as it takes the position sent by the Mobile Devices (especially of the vehicles), performs an accuracy check and enters the data into the database. The tracking engine is set up as SOAP web service (see chapter 2.2). To be able to create and

run a SOAP web service fast the Internet Information Service (IIS) is used as an appropriate platform, where Visual Basic and C# developed SOAP web services can be deployed. In particular, the Tracking Engine is written in Visual Basic (VB) 8 due to its simplicity and readability.

This web service takes several input parameters that are as follows:

- Latitude: in coordinate reference system WGS84 (EPSG:4326) as decimal degrees;
- Longitude: in coordinate reference system WGS84 (EPSG:4326) as decimal degrees;
- DOP: indicating the Dilution of Precision (DOP) value;
- Satcount: the number of satellites that are in view and used for calculation of the position;
- Timestamp: which is the exact date and time when the position is gathered at the mobile device;
- Truck_ID: indicates the primary key of the vehicle from the table truck, in order to specify a single vehicle;
- Transport_ID: key indicating the actual transport that is carried out, which corresponds to the primary key of the table transport_done – if no transport is carried out then a value equal to *nil* is submitted;
- status: a message indicating the status of the vehicle, e.g. pickup, delivery, breakdown, etc.;

The data are, if passing the accuracy check, inserted in the database using the appropriate web service of the Mod_DB. The accuracy check is a simple query that allows positions being inserted into the DB only if the number of satellites is ≥ 4 and the DOP is < 7. If the submitted value of Transport_ID is not equal to *nil* then the actual position is added to the geometry of the transport done in the table transport_done_geom, which can be visualized in the Mod_Desktop. Although the Tracking Engine has very limited functionality, it has to bear a great workload given a great number of vehicles sending their positions parallel. Thus, it has to be developed with respect to cope with a great amount of data.

5.4 Web Mapping Engine

The web mapping engine is the central data sharing component of the system. The web mapping engine makes use of the ISO and OpenGIS® stan-

dards wherever possible, in order to support interoperability. In particular the WMS and WFS standards are supported. The freely available standard web mapping server is used – the UMN mapserver (University of Minnesota, 2009). Through the various programming interfaces UMN mapserver offers, it is possible to adapt the functionality to the needs of WSC optimization. In order to support Web Feature Service-transactional (WFS-T), which UMN Mapserver does not support (University of Minnesota, 2009), Geoserver (OpenGEO, 2009) is used in parallel.

The Module_WME has a number of functionalities that are as follows:

- serve maps for Mod_Desktop including the visualization of trucks and their status, road network, and piles with additional information. These data are available as OGC compliant WMS.
- provide maps for Mod_Mobile (i.e. the vehicles) including their own position, the shortest way to the next vertex to be processed (either pickup or delivery), the piles around the actual position – with the pile where a pick up process should take place being highlighted.
- provide a WFS-T interface to alter data of piles and timber demand using the map interface.

In order to provide maps for Mod_Mobile – the vehicles – the application relies on data originating from the SDBMS. To generate such specific maps for each vehicle, the mapserver's functionality has to be extended facilitating a Application Programming Interface (API) offered. For UMN mapserver, php/Mapscript enables the generation of a dynamic web mapping service suitable for that purpose. This simple WS returns a map in a picture format of desired size if the following parameters are submitted:

- **Size of output picture** – width and height – in pixel, in order to support a number of mobile devices with different display resolution and/or size. This technical fact is crucial, because mobile devices like smartphones are exchanged on a frequent basis with changing displays and according specifications.
- **Truck_ID**: denoting the primary key of the table truck
- **Nextnode_ID**: identifies the next vertex (either pickup or delivery) to be processed. Based on this information it is possible to calculate the shortest path from the actual node to the next node and display it on the map.

The service queries the last submitted position of the specific truck and (if desired) centers the map to this point with a given zoom level. Thus, it

is possible to have (near) - live maps on the mobile devices located on the vehicles. Due to the fact that it is not possible to formalize this special map query as OGC compliant WMS request a WS is created that is capable of fulfilling the required needs. Nevertheless, all data can be accessed using WMS.

In order to serve maps to Mod_Desktop an OGC compliant WMS is opened including all data necessary for system administrators and the logistic center. Thus, several layers are offered in this WMS, among them are: the vehicles positions, vertices (either pickup or delivery), road network, etc. Through WFS-T it is possible to alter spatial and non-spatial data in a standardized way. In particular, it is possible to alter the following:

- timber piles and the related assortments (of table sortiment_pile): quantities, price, start and due dates as well as the position itself
- timber demand and the related assortments (of table sortiment_delivery): quantities, price, start and due dates.

The data of the Mod_WME are offered in two Coordinate Reference Systems (CRSs): WGS84 (EPSG:4326) and UTM-33N (EPSG:32633). In order to come up with a CRS that results in a "natural" visualization of Austria UTM-33N is used as the main CRS.

5.5 Decision Engine

The Mod_DE is an application wrapped in a web service that is able to solve the WSC optimization, according to the theoretical discussions in chapter 3 – using graph theory, heuristical optimization techniques and Rolling Schedule Approachs (RSAs) as basic foundation. In this chapter this module is described thoroughly, with particular attention towards applying ALNS (see chapter 3.4.4) to WSC optimization. This will result in modifications of the original algorithm due to the specific problem of WSC optimization. In addition, RSA is applied to the solution process in order to speed up the optimization and to avoid inaccurate results for events far ahead.

In general, the ALNS algorithm is implemented in VB 8, which seems not appropriate for a mathematical algorithm. But, due to the fact, that it has to be wrapped within a SOAP web service it is implemented in a object oriented manner in VB. VB has a disadvantage in speed. On the contrary, it offers a number of advantages, as readability and the ability to be easily integrated into a web service and object oriented design instead of monolithic software development principles.

5.5.1 Rolling Schedule for WSC optimization

The overall procedure of Mod_DE can be found in algorithm 5.1. There the combination of the basic foundations – graph theory, heuristical optimization and RSA – can clearly be seen. First, the Rolling Schedule Approach (RSA) is used to select which vertices – either pickup or delivery – have to be processed in the present planning horizon. Every vertex that fulfills the following requirements is chosen:

- $duedate_{vertex} \geq startdate_{planninghorizon}$
- and $startdate_{vertex} \leq enddate_{planninghorizon}$

where it is assumed that $\forall v \in V | v_{duedate} > v_{startdate}$. The selection of the timber quantity of a quality q to be delivered to a vertex v in Planning Horizon (PH) is done using equation 5.1, which is a simple linear distribution of timber not delivered so far. The equation has the following parameters: $quantity_{v,q}$ denotes the quantity of timber at a vertex v with a given quality q, $duedate_v$ indicates the due date of vertex v – when all timber has to be removed or delivered–, PH_{start} is the start date of the planning horizon and $PH_{duration}$ is the duration of the planning horizon. Notice, that $PH_{duration}$ and the result of subtraction $duedate_v - PH_{start}$ must be measured using the same time unit, e.g. days. With the factor $f > 1$ you can force a more immediate delivery to the sawmills, e.g. if weather conditions are deteriorating. Thus, the saw mills get more timber than in comparison to the linear distribution.

For very small timber quantities and/or a long duration where timber can be picked up or delivered, the euquation 5.1 results in very small timber quantities. In order to avoid such small-volume operations, a resulting quantity barrier value of at least 30 cubic meter is introduced. Thus, the quantity is never below this value, except the whole timber (with defined quality) at a node is below this barrier value. For these cases, the whole timber quality of a node has to be picked up or delivered.

$$quantity_{v,q}^{PH_x} = \frac{quantity_{v,q}\ not\ delivered\ yet * f}{\frac{duedate_v - PH_{start}}{PH_{duration}}} \tag{5.1}$$

The duration of the PH itself can be of various length, due to user defined input. For WSC optimization the time intervals, defined in chapter 3.5, should be set to whole days, as hours would not make any sense. As a default setting the planning horizon duration is set to three days, which is applicable given fixed external conditions – e.g. weather – during the half of a week.

Algorithm 5.1 Pseudocode for the Decision Engine.

Require: set of pick up vertices, set of delivery vertices, planning horizon and definition of time period, set of vehicles, road network;
map-match every new vertex with the road network; {*this can be done by a simple approach setting the next road network vertex as position for the new vertex*}
select pick up and delivery vertices that are affected by the current planning horizon
calculate proportional quantity of timber to be delivered to each delivery vertex with respect to rolling schedule – by using equation 5.1;
generate a graph using the vertices extracted above;
create an initial solution using a construction heuristic;
run ALNS using the initial solution;
return ALNS solution;

5.5.2 Graph theory for WSC optimization

In order to generate a VRP an undirected graph has to be defined (see chapter 3.1) using the vertices extracted by RSA, including the vertices of the depots – i.e. the haulage companies. Thus, a pre-processing step is performed that intensively makes use of the road network stored in the SDBMS. With the help of PostGIS and pgRouting (Orkney, Inc., 2009) it is possible to perform shortest patch calculations within PostGIS, using a number of algorithms e.g. Dijkstra–Algorithm (Dijkstra, 1959) or A*–Algorithm (Hart *et al.*, 1968). With this a graph is created where each pair of vertices is connected using the shortest path possible (with respect to the road network). Alternatively, the application is able to create paths between each pair of vertices using other criteria as well. Possible would be e.g.:

- path with the least inclination overall: which is especially interesting for trucks and ecological issues, as trucks emit most green house gases if driving uphill fully loaded
- path with least driving time: is of economical interest, but not every time ecologically reasonable
- path with least emissions: would be of ecological interest, but the emissions of a truck are hard to determine; this topic is out of the scope of this thesis.

5.5.3 Object oriented design for handling of optimization data

Having created a graph, it is stored in an appropriate structure. As stated in the heading, an object oriented approach is employed to design the Mod_DE. Thus, a number of VB–classes shown in figure 5.5 are developed. There, the relevant data are stored, and manipulated. In order to manage the data in the classes, a number of methods are developed.

The class **Node** stores relevant information on the vertices, including the distinction between depot, pickup or delivery vertex. In addition, the quantity timber initially present at the pile (if pickup), or the timber to be delivered (if delivery) is stored. For the case of delivery vertices the quantity is negative. It is obvious that the quality, start and due dates have to be inside the model too. The class **Nodes** is a collection of various instances of the class Node, where single nodes can be inserted and removed.

All data on a single vehicle are the content of the class **Vehicle**, including load limit, working time start/end, working days, to which depot the vehicle belongs, as well as the actual vehicle load – which will be necessary during ALNS. The class **Vehicles** is again a collection of Vehicle instances.

Class **Route** represents one route, meaning the tour that one vehicle makes in the course of one day. Hence, an instance of Route holds the following parameters: vehicle, node sequence, quantity sequence, quality sequence, price sequence and the time of arrival sequence. The sequences in the class indicate that lists are stored that reflect the time flow of events. At every vertex that is part of the route, the quantity of timber picked up or delivered – negative in that case –, and the according quality and price is stored. In addition, the time of arrival at each vertex is part of the class, which facilitates arrival time calculation in ALNS. Furhtermore, every route gets a parameter indicating the exact time frame (e.g. day) for which it is valid. The class Route has a number of functions, among them two, which are important for the evaluation of VRP solutions using the objective function defined. The mentioned functions together are capable of calculating the value of the objective function for a route. This is necessary in the ALNS process, in order to compare solutions found by the heuristic.

The class **VRP** is the father class, holding data of a solution of a VRP. Hence, a collection of nodes, a collection of vehicles that transport the goods from pickup to delivery vertices are needed. A number of routes represents the tours of the vehicles on the days in the PH. Using the functions to calculate the value of the objective function of each tour, it is possible to come up with the objective value for the whole VRP instance, which is of fundamental importance for the ALNS process.

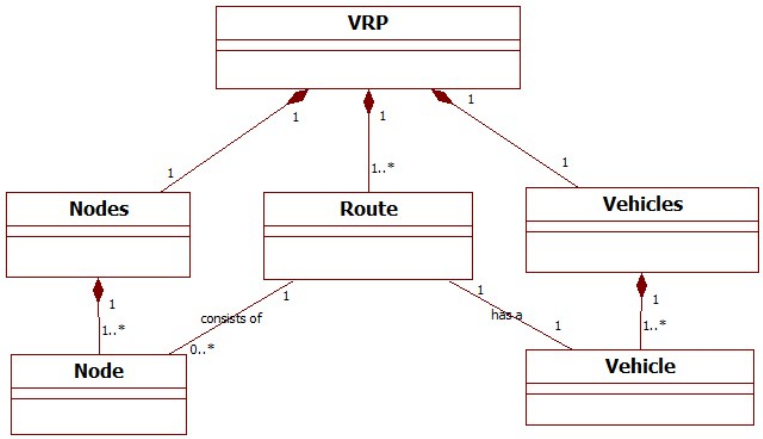


Figure 5.5: Class Diagram of VRP classes. The classes Nodes and Vehicles are collections of the classes Node and Vehicle respectively.

5.5.4 Retrieval of an initial solution

Using the classes defined in section 5.5.3, the Mod_DE proceeds to optimize the solution. Before ALNS optimization starts an initial solution has to be found using a construction method. The construction method employed here is similar to saving heuristics, mentioned in chapter 3.4.1, which results in a number of tours. Firstly, the algorithm creates routes that pick up timber at a timber pile and deliver it to a saw mill at first. Secondly, routes that fit are merged, until routes cannot be merged anymore. If the number of available vehicles is exceeded the routes with the lowest objective value are deleted. The pseudo code is shown in 5.2, where the procedure described in this paragraph is condensed.

Algorithm 5.2 Pseudocode for finding an initial solution.

Require: set of pick up vertices, set of delivery vertices, set of vehicles, road network;
create routes serving one pickup vertex and one delivery vertex; {starting and ending at a depot;}
repeat
merge routes based on start and due dates of vertices, vehicle driving time and working time limits;
until no routes can be merged
if $|routes| > |vehicles|$ **then**
delete n routes so that $|routes| \leq |vehicles|$;
end if
return initial solution;

5.5.5 ALNS for WSC optimization

After having found an initial solution the Mod_DE starts ALNS in order to optimize the initial solution. The general procedure of ALNS is to be found in algorithm 3.7. For the purpose of WSC optimization a number of alterations to the original algorithm proposed by Ropke and Pisinger (2006) and Pisinger and Ropke (2005) is made. Due to the changes in ALNS the adapted version is denoted as ALNS_{WSC}. Before the description of the ALNS procedure is started, detailed insight in the mathematical problem definition for WSC optimization is given. This model serves as guideline and "anchor" when implementing ALNS_{WSC}.

Mathematical problem definition

The mathematical model is based on the definition of VRP given in chapter 3.2.4. Thus the WSC is regarded as a VRPPDTW. The VRPPDTW is appropriate due to the fact, that there are pickup and delivery "places" – timber piles and saw mills – which have to be serviced by vehicles. Due to the fact that timber has to be delivered within a certain time period to saw mills on the one hand, and that a pile has to be removed by a certain date, each vertex has to be visited – at least once – within a given time window. Thus, the formulation shown starting with 5.2 until 5.20, is similar to the VRPPDTW formulation given in equations 3.10 until 3.24.

In the VRPPDTW that is constructed here, each request consists either of a pickup or a delivery node. A request, as defined in section 3.2.3 consists of a pickup and delivery node. In the requests defined here, solely the pickup or delivery demand is known. For each pickup node the associated delivery demand is calculated in the course of the optimization. Thus, a request – a pair of pickup and delivery nodes – is defined in the course of ALNS_{WSC} to facilitate a dynamic timber allocation. In contemporary WSC timber allocation is done before timber is harvested with contracts between forest enterprises and saw mills.

To model the WSC problem several abbreviations and definitions are necessary. The graph developed in a prior stage (see 5.5.2) is an undirected weighted graph $G = (V, E, w)$. The weight is defined as the length of edge $e \in E$, which can be also denoted as $(n, m) = e$ with $(n, m) \in V$. Hence, $w_{n,m}$ denotes the weight – i.e. the length – of edge $(n, m) \in E$. From the set of vertices V three subsets can be distinguished: V_P denoting the pickup vertices, V_D representing the delivery vertices and V_{De} which are the depots. As described in chapter 5.2 there is a set of assortment classes – timber qualities, defined in the table sortiment_list (see chapter 5.2 for

details) – Q from where a single assortment q can be identified. Pickup vertices $i \in V_P$ have for each assortment a defined quantity $p_{i,q}$, as well as a start date/time $pl_{i,q}$ and due date/time $pe_{i,q}$, and a defined price $pp_{i,q}$. Analogous, every assortment q in a delivery vertex $j \in V_D$ has a quantity $d_{j,q}$, a start date/time $dl_{j,p}$ and due date/time $de_{j,p}$ and a price $dp_{j,q}$. From the set of trucks T a vehicle $t \in T$ can be identified with the following parameters: capacity c_t, load time per m^3 timber d_t, working time start ws_t, working time end we_t. As stated above, a number of routes $r \in R$ exist, where r represents a route of a vehicle t which starts and ends at the according depot. Hence, a route can only be performed during a time period of the PH, where the time period has the length of one day. Every tour generates a subgraph $G_r = (V_r, E_r, w)$, which is a subset of the complete graph $G = (V, E, w)$. Additionally, every vehicle t forms a subgraph G_t of the original graph: $G_t = (V_t, E_t, w) \subseteq G = (V, E, w)$, where $G_r \subseteq G_t$. For formal reasons r is broken down into a smaller unit – the truckload $f \in F$, which indicates the following timespan of a vehicle:

- starting from having delivered timber and being empty
- loading timber at at least one pickup vertex
- ending being fully loaded and
- delivering the timber to a delivery vertex

In order to connect vehicles and timber located at piles and timber to be delivered to saw mills the following definitions and abbreviations are necessary: $tl_{i,q,t,f}$, denoting the timber loaded at pickup vertex $i \in V_P$. In order to cope with time windows the arrival time at a vertex $i \in V$ $A_{t,f,i}$ and the corresponding leaving time $B_{t,f,i}$ have to be defined. As mentioned in the prior list, at the end of a truckload, vehicles drop timber at delivery vertices $j \in V_D$, which is abbreviated as $td_{j,q,t,f}$. The actual load of a truck – with respect to a given quality – at a vertex $v \in V$ is $l_{v,q,t,f}$. To model the VRP accordingly, a decision parameter is necessary that models the traversal of a vehicle v from vertex $m \in V$ to vertex $n \in V$: $x_{m,n,t,f} \in (0, 1)$. The parameter $dt_{m,n}$ denotes the driving time from vertex $m \in V$ to vertex $n \in V$.

The mathematical problem formulation can be found in equation 5.2 until 5.20. The objective function 5.2 optimizes the profit of the WSC by maximizing the profit defined as sales revenue for delivered timber minus haulage costs. The haulage costs are the weight defined in G – i.e. the costs of the edge weight sum traversed by the vehicles. In order to generate costs in the monetary sense, the average haulage costs are assumed to be at a level of 0.95 EUR per km – which is an EU - average value for truck costs per

km (Herry Verkehrsplanung, 2001). For future applications it seems obvious to calculate with the consumed time to haul the timber multiplied by the vehicle cost per time unit, e.g. on an hourly basis. Recent publications show that the costs of trucks are about 65-77 EUR per hour (e.g. Linko, 2008; Friedl *et al.*, 2004).

The first half of the formula 5.2 reflects the sales volume of delivered timber, whereas the second part represents the haulage costs. The VRP is defined with the help of constraints (5.3 – 5.20). 5.3 limits the number of vehicles available for the solution of the VRP. In order to limit the loaded timber on a vehicle, formula 5.4 is appropriate. This assures that each truck does not load more timber as its capacity. In addition, the loaded timber at a pickup vertex has to be limited – it is not possible to load more timber than present in a pickup vertex, defined in equation 5.5. 5.6 models the leaving time and 5.7 the arrival time of vertices. As described above, time windows are applied to each vertex, which sets the time frame in which the timber must be delivered (for delivery vertices) or hauled (for pickup vertices). Thus, the application has to take care of time window constraints defined in 5.8 for pickup and in 5.9 for delivery vertices. In order to limit the timber delivered to saw mills, 5.10 regulates that there is not more timber arriving in saw mills than demanded. The additional variable s – denoting a slack variable – allows the delivery of more timber than demanded. This case is necessary in order to cope with oversupply, which might be the case e.g. after catastrophic windbreaks.

$$\max z = \sum_{t=1}^{|T|} \sum_{f=1}^{|F|} \left[\sum_{j=1}^{|V_D|} \sum_{q=1}^{|Q|} td_{i,q,t,f} * dp_{j,q} - \sum_{n=1}^{|V|} \sum_{m=1}^{|V|} x_{n,m,t,f} * w_{m,n} \right] \quad (5.2)$$

subject to:

$$\sum t \leq |T| \tag{5.3}$$

$$\sum_{q=1}^{|Q|} \sum_{i=1}^{|V_P|} tl_{i,q,t,f} \leq c_t \qquad \forall t, f \tag{5.4}$$

$$tl_{i,q,t,f} \leq x_{m,n,t,f} * p_{m,q} \qquad \forall q, t, f, m, n \tag{5.5}$$

$$B_{i,t,f} \geq A_{i,t,f} + \sum_{q=1}^{|Q|} (x_{m,n,t,f} * tl_{i,q,t,f} * d_t) \qquad \forall t, f \quad \forall (m,n) \in E \tag{5.6}$$

$$A_{m,t,f} \geq B_{n,t,f} + x_{m,n,t,f} * dt_{m,n} \qquad \forall t, f \quad \forall (m,n) \in E \tag{5.7}$$

$$pl_{i,q} \leq B_{i,t,f} \leq pe_{i,q} \qquad \forall i, q, t, f \quad i \in V_P \tag{5.8}$$

$$dl_{j,q} \leq B_{j,t,f} \leq de_{j,q} \qquad \forall j, q, t, f \quad j \in V_D \tag{5.9}$$

$$d_{j,q} \geq \sum_{t,f}^{|T|,|F|} td_{j,q,t,f} - s \qquad \forall j, q \tag{5.10}$$

$$\sum_{i,q,t,f}^{|V_P|,|Q|,|T|,|F|} tl_{i,q,t,f} = \sum_{j,q,t,f}^{|V_D|,|Q|,|T|,|F|} td_{j,q,t,f} \tag{5.11}$$

$$p_{i,q} = p_{i,q} - (x_{i,n,t,f,} * tl_{i,q,t,f}) \qquad \forall q, i, n \quad n \in V, i \in V_P \tag{5.12}$$

$$\begin{aligned} l_{m,q,t,f} &= l_{n,q,t,f} * x_{n,m,t,f} \\ &+ tl_{n,q,t,f,} * x_{n,m,t,f} \\ &- td_{n,q,t,f} * x_{n,m,t,f} \end{aligned} \qquad \forall q, t, f \quad (n,m) \in E \tag{5.13}$$

$$\sum_{m,f}^{|V \setminus V_{De}|,|F|} x_{n,m,f,t} = 1 \qquad \forall t, f \quad n = v_t \tag{5.14}$$

$$\sum_{m,f}^{|V \setminus V_{De}|,|F|} x_{n,m,f,t} = 1 \qquad \forall t, f \quad m = v_t \tag{5.15}$$

$$\sum_{n,m \in V_t}^{n<m,m<|V_t|-1} (x_{n,m,f,t} - x_{m,m+1,f,t}) = 0 \qquad \forall t \tag{5.16}$$

$$ws_t \leq A_{n,t,f} \leq we_t \qquad \forall n, t, f \quad n \in V \tag{5.17}$$

$$ws_t \leq B_{n,t,f} \leq we_t \qquad \forall n, t, f \quad n \in V \tag{5.18}$$

$$p_{i,q} \geq 0 \qquad \forall i, q \quad i \in V_P \tag{5.19}$$

$$x_{m,n,t,f} \in \{0,1\} \tag{5.20}$$

5.11 indicates that no timber gets "lost" after the pickup process and generally in the WSC. Thus, the amount of timber flowing out of pickup

vertices has to be equal to timber amount flowing in delivery vertices. Timber picked up reduces timber present at a pickup vertex, which is modeled by 5.12. The actual load of trucks is represented by 5.13. This calculates the actual load of a certain quality q on vehicle t located at vertex m. In order to cope with loading and unloading processes, the second line of 5.13 denotes a pickup process and the third line reflects timber delivery.

5.14 and 5.15 ensure that each route starts and stops at the vehicles's depot. By using the subgraphs created by each truck $G_t = (V_t, E_t)$ it has to be ensured that the according truckloads f are connected. This is guaranteed by 5.16. 5.17 and 5.18 limit the working time of each vehicle accordingly. 5.19 defines that the parameters that are representing timber at a vertex must not be lower than zero – in order to prohibit that timber is removed from piles although nothing is left. 5.20 states that the decision variable can only have the values 0 or 1.

In order to avoid numerous unloaded drives of vehicles, there is no direct formulation. As the objective function of the optimization problem itself is maximized (see equation 5.2), there are two competing components: profit and cost. A maximization of the objective function is reached by lowering the cost and/or increasing the profit. Due to the fact that an unloaded or partly empty drive of a vehicle would result in a cost increase, while the profit is not or only marginally increased, the system would rate the change in the objective function not as an improvement and refuse this result. Hence, the optimization system itself cares about the rejection of empty and partly empty drives, as they result in additional costs not maximizing the objective function.

ALNS implementation

As described in section 5.5.4, after having found an initial solution based on rolling schedule approach and graph theory,the problem is optimized using ALNS_{WSC}. The basic ALNS procedure is described in detail in chapter 3.4.4. This section focuses on the alterations made to ALNS in order to fit to WSC optimization defined in the mathematical problem definition. The general principles of ALNS is defined in algorithm 3.7, which will serve as foundation for ALNS_{WSC}. The ALNS pseudocode is printed repeatedly (see algorithm 5.3) in order to enhance readability.

In order to bridge the gap between theory and implementation, the theoretical concept of requests is altered to fit to ALNS_{WSC}. Due to the fact that every pile and timber demand has at least one assortment with according quantity, price, etc. (see chapter 5.2) every record in the table sortiment_pile

and sortiment_delivery is modeled as request. Thus, a single position may have several requests nodes, which is not common, but reasonable in this context. With this arrangement it is possible to model e.g. multiple requests at one pile/saw mill. Therefore, if thinking of requests it has to be thought of:

- timber with defined quality and quantity located at position – for the case of pickup vertices
- timber with defined quality and quantity to be delivered to a position – for the case of delivery vertices

Algorithm 5.3 ALNS$_{WSC}$ pseudocode, based on Pisinger and Ropke (2005).

construct an initial VRP s;
$s^* = s$; {initial solution is the best solution at the beginning}
repeat
  Choose a destroy neighborhood $\mathbb{N}^-$ and repair neighborhood $\mathbb{N}^+$ using roulette wheel selection based on previously obtained scores π_j;
  Generate a new solution s' from x using the heuristics corresponding to the chosen destroy and repair neighborhoods
  if s' can be accepted **then**
    $s = s'$;
  end if
  if $f(s) < f(s')$ **then**
    $s^* = s'$;
  end if
until stopping criterion is met
return s^*; {Return best result}

For the purpose of removing requests from a VRP solution there are three techniques:

- Shaw Removal
- Random Removal
- Worst Removal

Shaw Removal removes the requests that are similar or have a certain similarity value. In the ALNS described in chapter 3.4.4 this value is calculated using formula 3.28. Due to the fact that all vehicles are able to serve requests equation 3.28 is modified and equation 5.21 is formed, which omits the vehicle part. Nevertheless, the remaining three parts affect distance,

time and capacity. For clarification, the variables used are described as follows: $R(i,j)$ is the relatedness measure of request i and j. The variable $d_{i,j}$ denotes the distance between the requests. T_i indicates the time when location i is visited. l_i denotes the quantity of timber located at request i. $q_{i,j}$ denotes a factor regarding if timber quality of request i is equal to j (explained in 5.22), and $t_{i,j}$ if the request type is equal (see 5.23). C is a constant that has a high numeric value, in order to increase the similarity value $R(i,j)$. The higher $R(i,j)$ is, the lower is the similarity between two requests i and j. Random Removal and Worst Removal are implemented as described in chapter 3.4.4 with no alterations.

$$\begin{aligned} R(i,j) = \varphi(d_{i,j}) + \chi(|T_i - T_j|) \\ + \psi(|l_i - l_j|) + q_{i,j} * C + t_{i,j} * C \end{aligned} \tag{5.21}$$

$$q_{i,j} = \begin{cases} 0 & \text{if } q_i = q_j, \\ 1 & \text{if } q_i \neq q_j \end{cases} \tag{5.22}$$

$$t_{i,j} = \begin{cases} 0 & \text{if request types } i,j \text{ are equal (e.g. pickup)} \\ 1 & \text{if request types } i,j \text{ are not equal (e.g. pickup and delivery)} \end{cases} \tag{5.23}$$

The insertion heuristics mentioned by Ropke and Pisinger (2006) are: Basic Greedy heuristic and Regret heuristics. The Basic Greedy heuristic implemented in ALNS_{WSC} differs from ALNS, in the form that it is purely interested in increasing the overall profit – sales volume minus haulage costs. Thus, $\Delta f_{i,k}$ denotes the difference of the objective function value that occurs when inserting request i into route k at the position that results in the *highest* increase of the objective value. If no insertion is possible then $\Delta f_{i,k} = \infty$. c_i is defined as the profit of inserting the request i at its best position where $c_i = \max_{k \in K}\{\Delta f_{i,k}\}$ and name this position the *maximum profit position*. Additionally, the request i is selected that fulfills the function:

$$\max_{i \in U} c_i \tag{5.24}$$

U denotes the set of unplanned requests. The process continues until no more requests can be inserted or all requests are inserted.

The family of Regret heuristics are a class of heuristics that incorporate a look into the future, by inserting requests that are "hard" – which requires an intelligent selection process. The definition of "hard" requests cannot be given within a sentence, thus, the following explanation should clarify this

term. $x_{ik} \in \{1, \ldots, m\}$ is a variable that represents the route for which request i has the k^{th} highest increase of the objective value, and m is the number of vehicles. Mathematically defined, this can be denoted as $\Delta f_{i,x_{ik}} \geq \Delta f_{i,x_{ik'}}$ for $k \leq k'$. As mentioned in chapter 3.4.4, c_i can defined as $c_i = \Delta f_{i,x_{i1}}$. Subsequently, the regret value c_i^* is defined as $c_i^* = \Delta f_{i,x_{i1}} - \Delta f_{i,x_{i2}}$. Hence, the regret value is the difference in the profit of inserting a request in its best and second best route. In each iteration the request i is chosen that fulfills equation 5.25 and this request is inserted at its maximum profit position. As a result the request is inserted that is regretted most if it is not done now.

$$\max_{i \in U} c_i^* \tag{5.25}$$

As mentioned in the prior paragraph Regret heuristics form a family of inserting heuristics – Regret-k heuristics. In the prior paragraph the Regret-2 heuristics is mentioned, which calculates c_i^* with respect to the 2^{nd} best $\Delta f_{i,x_{ik}}$ value. By considering the k^{th} best route c_i^* has to be calculated in the following way:

$$c_i^* = \sum_{j=1}^{k} \left(\Delta f_{i,x_{i1}} - \Delta f_{i,x_{ij}} \right) \tag{5.26}$$

The ALNS$_{WSC}$ is restricted to the Regret-3 and Regret-2 Heuristic, due to the fact that it is important to save computational resources. For future applications Regret-k Heuristics could be expanded to $k > 3$.

For the selection of an appropriate remove and repair heuristic, the roulette wheel selection process has been described in section 3.4.4. For ALNS$_{WSC}$ the selection process is implemented in the way described in the theoretical part. Iteratively weights are assigned to each heuristic, due to the performance during the last 100 iterations. Depending on the weights the roulette wheel configuration is chosen, which significantly influences the roulette wheel selection process.

As described in the theoretical part, the insert and remove heuristics are enveloped in a metaheuristic, that controls the acceptance of solutions – see algorithm 5.3: line *[...] if s' can be accepted [...]*. Generally speaking, "good" moves are always accepted, which means that if $f(s) < f(s')$ – then s' is accepted anyway. For the case of $f(s) \geq f(s')$ the metaheuristic is applied that decides if the new solution s' is accepted. The metaheuristic used is Simulated Annealing (SA) (see chapter 3.4.3 and 3.4.4 for details). For clarification reasons, SA is necessary in order to escape local maxima. A local maxima is valid for the case of ALNS$_{WSC}$, due to the fact that the objective function is maximized (see equation 5.2). Thus, the metaheuristic applied accepts solutions that are worse than the current solution s. In order to do so, a new solution s' is accepted if it fulfills the following equation –

where R is a random number:

$$e^{-\frac{f(s)-f(s')}{T}} > R \qquad R = \{x \in \mathbb{R} | (0 \leq x < 1)\} \tag{5.27}$$

As described in the theoretical part, SA starts with a given temperature T_{start} and decreases the temperature T with a given cooling rate $0 < c < 1$, so that $T = T * c$. For ALNS$_{WSC}$ a cooling rate of 0.99975 is chosen, according to the given parameter set by Ropke and Pisinger (2006, p. 14). Important for the whole process is the initial T_{start}, which is dependent on the initial solution. By evaluating the objective function of the initial solution the solutions profit is calculated. T_{start} is chosen that a solution that is w worse than the initial solution $s_{initial}$ is accepted with a probability of 0.5. w the start temperature control parameter is set to 0.05 – indicating that new solutions are accepted that are 5% worse than the current solution with probability of 0.5. The calculation of the T_{start} is as follows:

$$T_{start} = \frac{-(f(s_{initial}) - f(s_{initial}) * (1 - w))}{\ln 0.5} \tag{5.28}$$

The stopping criteria of ALNS$_{WSC}$ is met if a number of iterations have passed. The number of iterations is set to 25000, in order to reduce computational time. As mentioned in chapter 3 and more detailed in chapter 3.4 results obtained by ALNS$_{WSC}$ do not represent an exact optimal solution. Due to the fact that heuristics are used, the results may be suboptimal in comparison to the exact solution. Thus, it cannot be guaranteed that the ALNS$_{WSC}$ process does not result in a local maximum. But, as the objective of the process is, to find a "better" solution than the initial one, it can be stated that finding a "better" solution is independent of the solution property, i.e. being a local maximum. The reduction of vehicles used in the VRP is not implemented, as it is only necessary to show that the system itself works properly.

5.6 Mobile Devices

The Mod_Mobile are handheld devices capable of establishing mobile internet connections and having a sufficient large screen for displaying a UI. In addition, they are capable of determining their position. Hence, PDAs or smartphones (with built-in GPS/with external GPS) are technical devices which form the technical basis for Mod_Mobile. From a user's perspective it has to be distinguished between two applications of this module: a) for the use on vehicles, b) and for the purpose of managing of piles by forest enterprises. Smartphones and PDAs are chosen due to their growing market share and economic advantage in comparison to other mobile computers (e.g. toughbooks).

Both applications share the development platform, which is Windows® Mobile 6 for smartphones and Windows® Mobile 6.1 for PDAs. These very similar platforms are ideal for creating mobile solutions, by exploiting the capabilities of the Compact Framework of Windows®. Both applications are developed using VB 8, due to its simplicity and ability to integrate a number of external programs. In order to connect with the external or internal GPS the third-party software GPS.Net 2.0 (Geoframeworks LLC, 2009) is employed, which has been released to the public domain for the benefit of the open source development community in 2009.

5.6.1 Mobile Device for vehicles

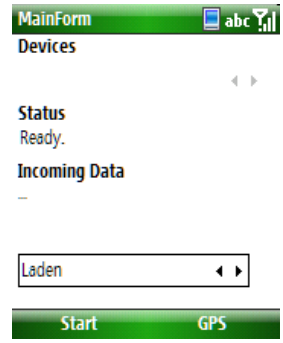
The Mobile Device application for vehicles provides functionalities that send and receive data from Mod_Tracking, Mod_WME and Mod_DE (see figure 5.6). Crucial is the ability to determine the own position, as well as a connection to the Internet. The data transmitted and received are described in chapter 5.1. Important functionalities of this application are:

- Assignment management
- Dynamic mapping functionalities
- Tracking functionalities

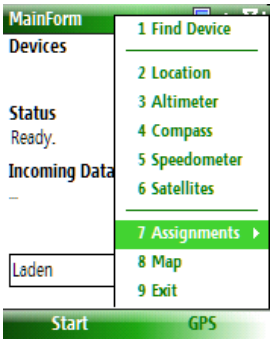
Assignment management includes the automatic receiving of the tour the vehicle has to perform on the present day. The driver can accept the assignment and then starts the tour, using the Location Based Services (LBSs) offered by the system architecture. Subsequently, the shortest path to the next pickup or delivery node is mapped, and the attributes can be visualized using the mobile device. If a request was serviced, it can be marked as completed and the timber amount hauled or delivered is entered in the system, in order to monitor the actual quantity of the timber pile or timber already delivered. The vehicle operator is able to cancel a request, which will be in most cases a pickup node, due to bad weather. In case of cancellation of a vertex, the truck driver can submit a reason why the node could not be served.

In the paragraph above the mapping of the shortest path to the next node is mentioned. Dynamic mapping functionalities utilizing Mod_WME provide this functionality. The application updates the map in a defined time interval and centers the map to the position of the truck if desired by the user. For sure, the standard operations like zoom, pan, etc. are available.

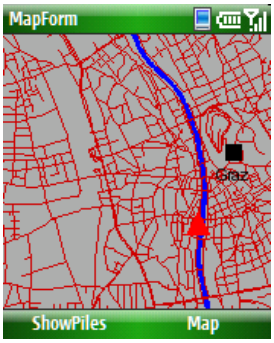
Tracking functionalities make use of two sub-functionalities: self positioning and submitting the own position and status. The determination of the own position is done using GPS technology (see chapter 2.3 for details), which can be either built in or external (usually connected via Bluetooth). Using this data, the mobile application sends this position in a defined interval to the Mod_Tracking via a web service. In addition, the driver can alter the actual vehicle status, that is sent in conjunction with the position updates, using the UI. Possible values are: Pickup, Delivery, Driving and Breakdown. For the case of a not existing internet connection, the tracking data are stored locally and transmitted in a bundled way if the connection is available again. Due to the fact that every position is marked with a timestamp there is no consistency problem when inserting the data in the database.



(a) start screen with status message



(b) options tab



(c) visualization of the map

Figure 5.6: Screenshots of mobile device for vehicles. In 5.6(a) the start screen of the application is visualized and in 5.6(b) the available program features are displayed. In figure 5.6(c) the red lines indicate the road network and the gray areas show the settled areas. The red triangle marks the position of the mobile device, i.e. the vehicle.

5.6.2 Mobile Device for forest enterprise

The mobile device for forest enterprises has similar capabilities as the one for vehicles, but has a limited set of functionalities. Nevertheless, it relies on self positioning using GPS and transmitting data using mobile Internet. The functionalities of this application are as follows:

- "Creation" of timber piles
- Alteration of timber piles
- Mapping of timber piles

The "creation" of timber piles provides an application to input data of new timber piles including their position. The position can either be entered as WGS84 lat/lon coordinates, or gathered from the connected GPS device. The data is then sent via a web service to the SDBMS. The inserted piles can be managed by the forest enterprises as desired, which includes an alteration and monitoring of e.g. quantities and due dates. Nevertheless, it is possible to map the piles using the Mod_WME in an appropriate way. The mapping function centers the map to the actual position of the device and visualizes the relevant data concerning timber piles for the forest enterprises.

5.7 Desktop System

The Desktop System is a system for expert users, like people working in the logistics center and the system administrator. It is intended to run on an desktop PC and is developed using VB 8 using .Net technology (see figure 5.7). Nevertheless is utilizes OGC and ISO standards respectively, which are served by Mod_WME. In particular WMS, WFS and WFS-T are of importance for this application, which has the following functionalities:

- Alter model parameters
- Manage WSC data
- Dynamic mapping of vehicles, pile, delivery data

The alteration of model parameters consists of setting the parameters of ALNS via the UI including: the maximum number of iterations, q which represents the number of requests to be removed, p the degree of freedom in the removal process, r the reaction factor of the roulette wheel selection (see chapter 3.4.4 for details), f that controls how immediate timber is hauled to

the sawmills (see chapter 5.5.1), the duration of the PH, the time interval duration, c the cooling rate of Simulated Annealing (SA) and w the start temperature control parameter for SA.

The management of WSC data involves the creation and management of haulage companies, according trucks as well as saw mills, and their detailed demand profiles. As described in chapters 5.2 and 5.1 this data has to be put in manually by an expert. All this can be done using the delivered UI of the application, and does not require database experts. The mapping of vehicles,

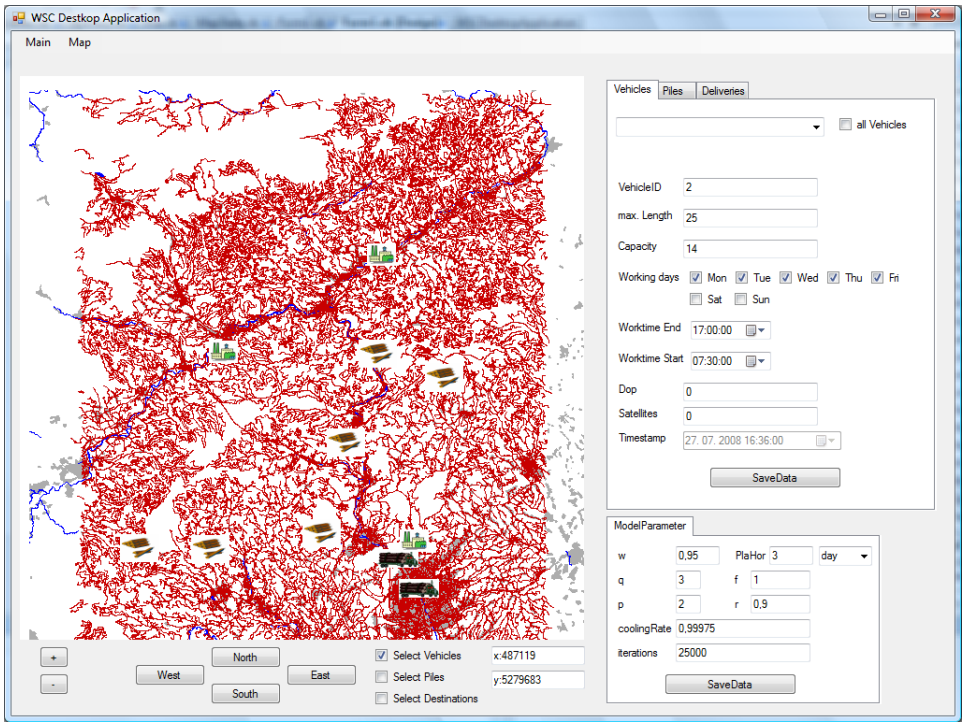


Figure 5.7: Screenshot of the desktop application.

piles data and delivery data in a dynamic map requires the connection of the Mod_Desktop with the Mod_WME. Every map is served via a WMS that is enhanced by php/Mapscript. The map presented in the Desktop application refreshes itself in a given time interval, in order to present near live information. With the help of this tool the trucks and their status – including their actual load –, the piles and their according assortments with quantities as well as the status of delivery nodes can be visualized. By using WFS it is possible to have access to entity data as desired.

The Desktop System is of great importance for WSC management, as all data of the WSC are merged in this view. Thus, a complete monitoring of data is possible, which avoids timber getting lost in the Supply Chain.

5.8 Conclusion

This chapter elaborates on the implementation of a real-time optimization system for WSC management. By using the theory described in chapters 2, 3 and 4 it is possible to create such a system. In detail, an overview of the system – utilizing SOA – is given, that shows the components and how they are connected. Subsequently, all components are described thoroughly. The Decision Engine, as most innovative part, makes use of ALNS, that is tailored towards optimizing the WSC. Because of these amendments, the resulting algorithm is called ALNS$_{WSC}$. In addition, the mobile devices for vehicles and forest enterprises are shown that are used to collect real-time spatial data. A desktop system is available for experts in the logistics center, that visualizes all relevant data.

Chapter 6

Setup and Results

This chapter discusses the experiment setup and the results of WSC optimization using ALNS. The experiment setup subchapter summarizes the data used and gives insight in the problem instances that are analyzed. The obtained results, with respect to the settings are subject of the 2nd subchapter. There, also the time expenditure is highlighted, which is necessary to develop a WSC optimization prototype.

6.1 Experiment Setup

The experiment setup consists of the following components that will be discussed in detail in the following paragraphs. First, the hardware components are described, followed by the software and the (spatial) data that form the basis for the results. In addition, the area of interest, and the problem instances created, are discussed. These problem instances are solved accordingly, and the results are published in chapter 6.2.

6.1.1 Hardware

The hardware components are selected in order to fulfill the needs of optimizing the WSC. Hence, the project relies on "standard" equipment, that is comparatively cheap and easy to handle. In addition, when choosing hardware it is necessary to have an eye on market acceptance, in order not to create legacy systems nobody is willing to use. Generally speaking, the hardware can be divided into the following basic categories, which are visualized in figure 6.1:

- Mobile
- Server
- Desktop

Mobile hardware consists of the devices for the vehicles and forest enterprises. Generally speaking, hardware is used that has a certain market share and is affordable. Thus, it seems advisable to choose the smartphone HP iPAQ 514 Voice Messenger for the trucks and a PDA HTC Touch for the forest enterprises due to their greater flexibility with a touch screen. The smartphone has only limited capabilities of data input, as there is no touch screen available, but truck drivers have only to submit a few data. Hence, a simple UI with keyboard input is sufficient.

The server hardware may consist of a number of servers that are geographically dispersed holding the modules described in chapter 5.1. In the course of this thesis, all of the modules are located on one machine. A standard desktop computer with an Intel Pentium D processor is used. Also for the desktop system a standard hardware is chosen – in this case the Intel D Processor computer. For this system any standard desktop hardware system that is available, is appropriate.

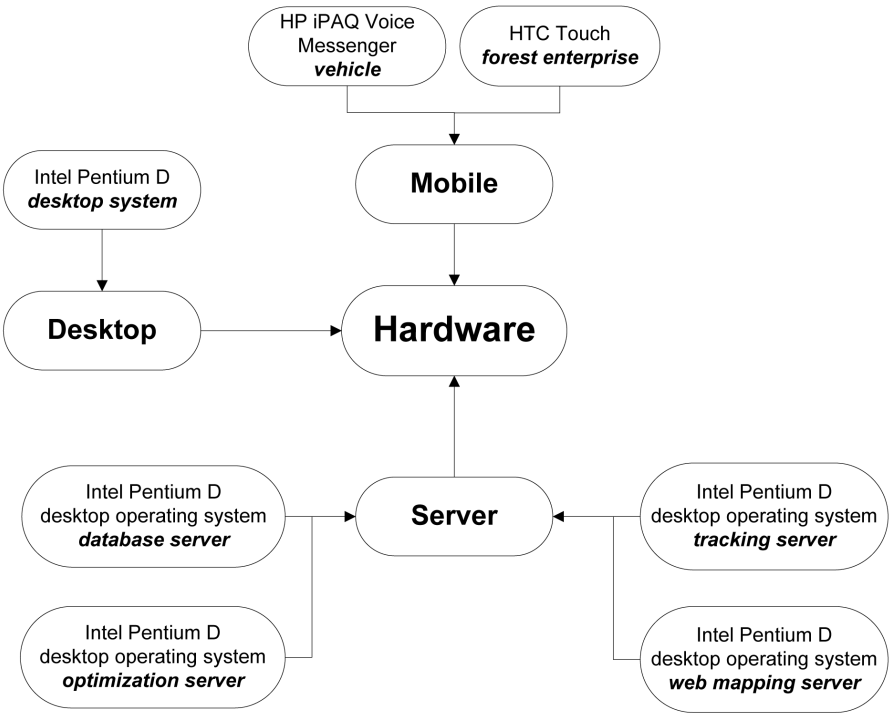


Figure 6.1: Hardware overview for WSC optimization used in this thesis.

6.1.2 Software

The software used in this work – besides the developed solutions described in chapter 5 – relies mainly on free and open source software. Analog, to the previous chapter the software used in this thesis is divided into three categories: mobile, server and desktop. An overview can be found in figure 6.2.

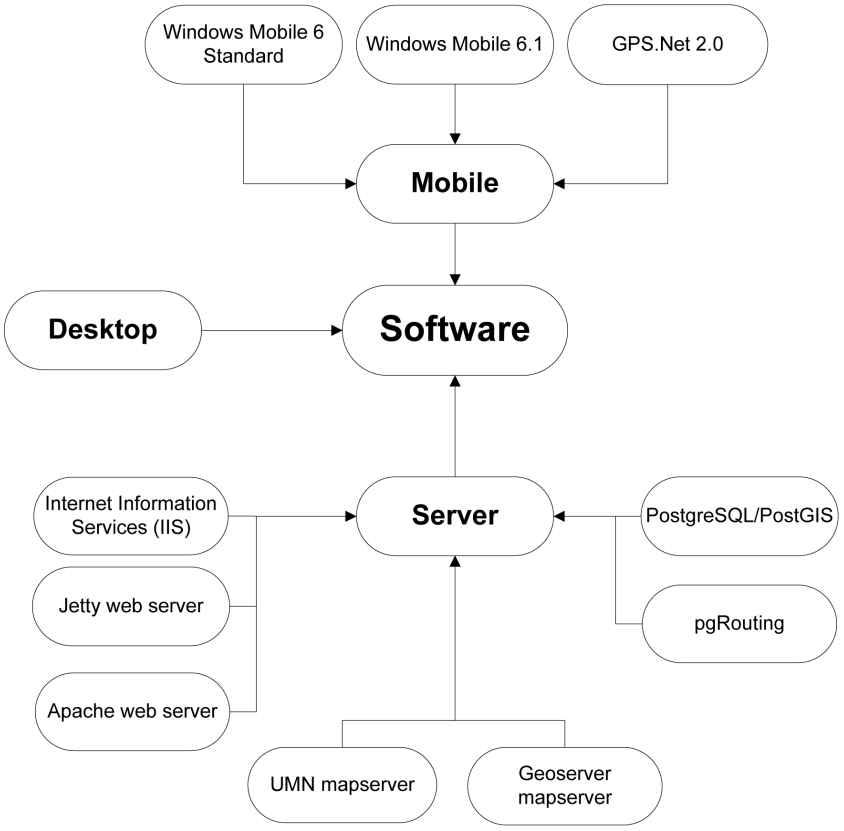


Figure 6.2: Software overview for WSC optimization.

The desktop software simply relies on a Windows® operating system. The software used on the mobile systems relies on Windows® Mobile – version 6 for smartphones and version 6.1 for PDAs. To connect with an external or internal GPS the third-party software GPS.Net 2.0 (Geoframeworks LLC, 2009) is used. As stated in chapter 5.6 this software has been released to the public domain in 2009. Due to the API that this software offers, it can be easily integrated in the mobile applications.

The server modules rely on a great number of external software. Basically Windows® operating systems have to be installed on each server. PostgreSQL 8.2 with PostGIS 1.2.1 serves as SDBMS. PostGIS implements the OpenGIS® Simple Features Interface Standard (SFS) (OGC, 2006a), which is beneficial for the standardized storage of spatial data. In order to perform shortest path calculations using various algorithms the add on pgRouting (Orkney, Inc., 2009) is appropriate, which is also discussed in chapter 5.5.2.

For web mapping and data handling issues web mapping servers are employed. As discussed in section 5.4, UMN mapserver (University of Minnesota, 2009) and Geoserver (OpenGEO, 2009) are used in parallel, due to the Geoserver's WFS-T capabilities. The UMN mapserver can be obtained in a bundled installer package (FWTools), with the Apache web server already included. In addition, the Geoserver brings its own web server and java servlet container – the Jetty web server (Eclipse Foundation, 2009). In order to support REST and SOAP web services in an easy way using the .Net framework the Microsoft Internet Information Services (IIS) web server is inevitable.

6.1.3 Spatial data and area of interest

A SDSS for WSC optimization certainly needs spatial data in order to compute accurate results. By using the data model (described in chapter 5.2), the mobile devices (chapter 5.6) and the desktop system (chapter 5.7) a set of spatial data is generated. In addition, the system needs "institutional" data that supports routing and mapping.

Besides the "ad-hoc" collected data – mentioned in the preceding paragraph – selected data sets of TeleAtlas Multinet for Austria are used, here in CRS UTM-33N (EPSG: 32633). The following layers can be found:

- Road network (polyline)
- Water bodies (polyline)
- Settled areas (polygon)
- Towns and villages (point)

Nevertheless, more data could be added like a hillshaded digital terrain model or railway lines. In order to reduce the amount of data sent – especially for mobile devices – the application is limited to the layers mentioned above. The road network consists of all roads that are collected by TeleAtlas

with according data and metadata. In the data section, valuable information on e.g. the average speed possible on a certain road segment is stored, which supports the optimization and the construction of a graph respectively. Due to the fact that these data are network models (see chapter 2), which are topological connected non-planar networks, they can be used for routing purposes. Because topology has to be built up in PostGIS using pgRouting, the modeling of turns is left out. Due to the lack of public available data on forest roads in Austria, in contrast to Germany where the company NavLog collects and sells navigable forest road data sets (NavLog GmbH, 2009), the data are restricted to the public road network. In order to show that the system and optimization itself is working properly, a forest road network is not essential.

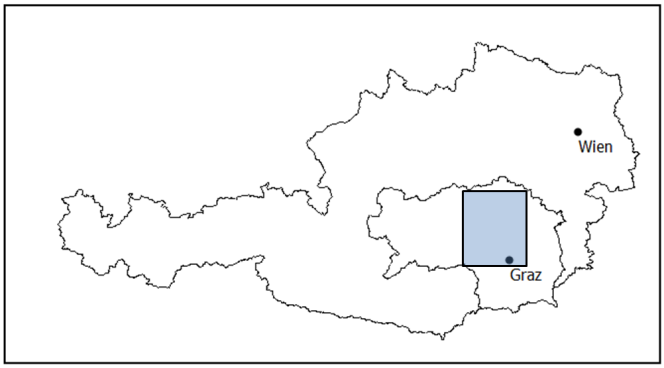
The area of interest of this project is in the center of the Austrian province of Styria (see figure 6.3). Within the given area of interest two problem instances are created that are discussed in chapter 6.1.4. The test area has a size of approximately 5018 km^2. The road network layer in this certain area contains 55308 road segments with a total length of 15090 km.

6.1.4 Problem instances

In order to test the WSC optimization implemented as described in chapter 5, two test instances are created that are analyzed using $ALNS_{WSC}$. The developed test instances are listed in part III – the appendix.

The test instances listed in part III have a number of abbreviations, that are explained in this paragraph. In the tables showing timber supply (e.g. table 3) the columns X and Y indicate the position of the timber pile (CRS EPSG:32633) with ID P_{ID}. The columns *start* and *due* show the start and due date of the pile, the columns *qual* and *quant* the quality index and the according quantity to be picked up. Column *start* indicates the start date of the timber quality - which may be different than the start date of the pile.

The saw mill positions are listed in e.g. table 5, where the positions are given in CRS EPSG:32633. The column *saw mill ID* indicates the primary key originating from the database table destination. In e.g. table 4 the corresponding timber demand of each saw mill is listed. The *saw mill ID* is equivalent to the column described above, and *task* indicates the primary key of the database table timberdelivery. Columns *start* and *due* show the start and end date of the timber demand, i.e. to which date the timber should be fully delivered. The columns *qual* and *quant* are equal to timber demand, and *price* denotes the price for a cubic meter of timber with specific quality.



(a) Overview map

(b) Detail map

Figure 6.3: The test area for WSC optimization marked with the blue box – the overview in 6.3(a) and the detailed map in 6.3(b).

The problem instance for vehicles and haulage companies is displayed e.g. in table 6. There, the column *haul_comp* is the primary key of the table haulagecompany with the according position of these depots in columns X and Y in EPSG:32633. The associated trucks are shown in this table too, where the *truck ID* is equivalent to the primary key of the table truck. The column *cap* indicates the truck's capacity in cubic meter, the columns *start* and *end* show the working time limitations of each truck. The column *w_days* represent the working days as described in chapter 5.2.

To test the application, two test instances have been created, that are listed in part III. Instance 1 consists of ten piles, 28 assortments (records in the table sortiment_pile), four saw mills, four delivery profiles (entries in the table timberdelivery), 38 records in the table sortiment_delivery and 35

trucks in six haulage companies. A map of the entities of test instance 1 is shown in figure 6.4. Thus, the spatial dimension and relation of the test instance can be determined. Instance 2 – see appendix – has 20 piles, 80 records in the table sortiment_pile, four saw mills, 43 entries in the table sortiment_delivery that are grouped in ten timberdeliveries and 35 vehicles in six haulagecompanies. The trucks, haulagecompanies as well as the saw mills are equal in both test instances. In figure 6.5 a map of the second test instance's entities are visualized. Both maps – test instances 1 and 2 – are using CRS UTM-33N (EPSG:32633).

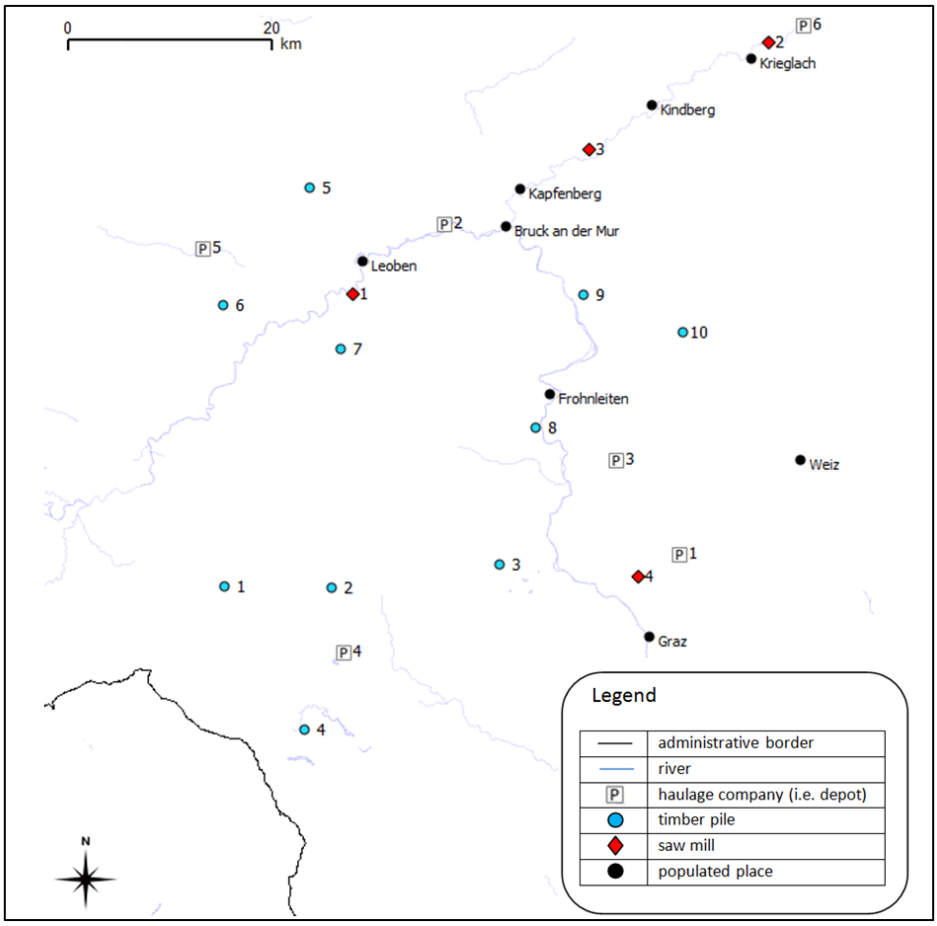


Figure 6.4: Map with the entities present in test instance 1 – piles, saw mills and haulage companies. The administrative borders, rivers and populated places are displayed for a better orientation.

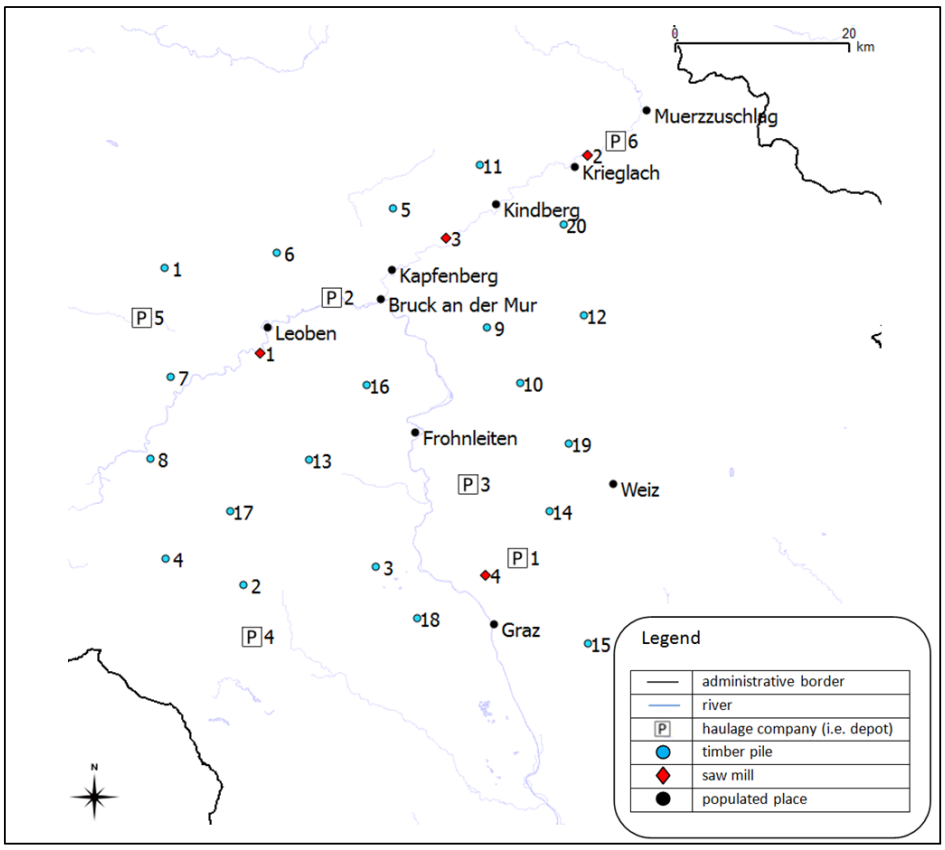


Figure 6.5: Map with the entities present in test instance 2 – piles, saw mills and haulage companies. The administrative borders, rivers and populated places are displayed for better orientation.

6.2 RESULTS

This section documents the results obtained with the system and the test data described in chapters 5 and 6.1. In addition, the work expenditure necessary to build up such a prototype is given. Subsequently, the results of ALNS_{WSC} applied to the two test instances, described in 6.1.4, are given.

6.2.1 Work Expenditure

This subchapter focuses on the work invested to develop the prototype described in chapter 5. The work was done following a project working plan, where the timeline was flexible. Nevertheless, the time necessary to create such a prototype can be seen as ”temporal guideline”. Thus, this issue is discussed here.

The overall time spent to create the prototype for WSC optimization was 1310 working hours, which were spent over a time of approximately two years. This amount of working hours represent solely the developing phase, without any documenting or scientific paper writing.

As it can be seen in figure 6.6 most of the time was spent developing the Mod_DE (see table 6.1). In figure 6.7 the sequence of working packages is displayed. It can be seen that the system architecture was developed first, followed by the DBMS. Subsequently, Mod_WMS and Mod_Tracking were implemented afterwards, in order to provide a solid basis of the SDSS. Mod_Mobile and Mod_DE make use of this structure and heavily rely on the prior developed modules. In the course of the implementation some alterations to Mod_Mobile were necessary, which were carried out in a late project phase. Also the Mod_Desktop was developed late in order to fully rely on the existing structures and services.

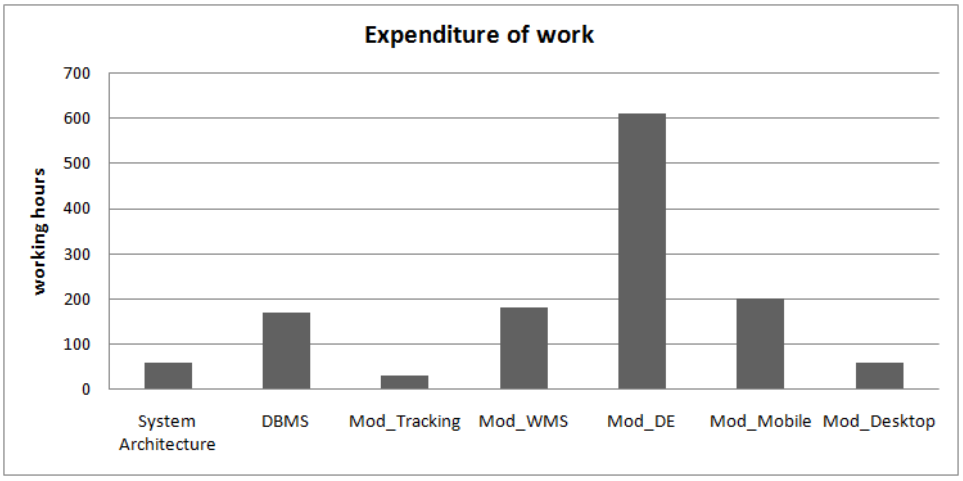


Figure 6.6: Amount of work in hours invested in the development of the WSC prototype, separated by modules. The position "System Architecture" stands for the overall conceptual planning of the system itself.

Module	total working hours	ratio
System Architecture	60	4.58%
DBMS	170	12.98%
Mod_Tracking	30	2.29%
Mod_WMS	180	13.74%
Mod_DE	610	46.56%
Mod_Mobile	200	15.27%
Mod_Desktop	60	4.58%
Sum	**1310**	**100%**

Table 6.1: Work expenditure in absolute and relative numbers by modules.

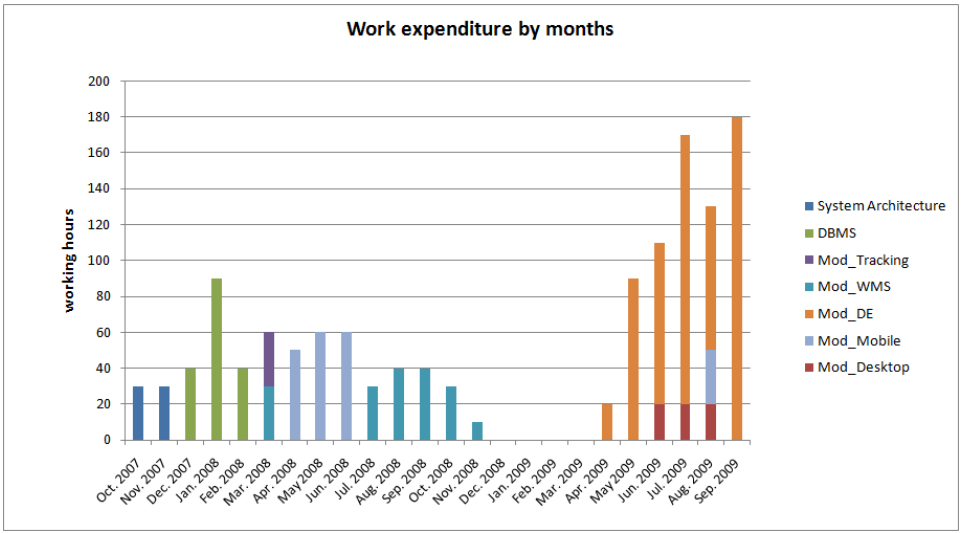


Figure 6.7: Amount of work in hours separated by modules by each month.

6.2.2 Results of ALNS applied to WSC

In the following section the results obtained with ALNS_{WSC}, described in chapter 5, are described and analyzed. There, the focus lies on the experiments with ALNS_{WSC} with the problem instances mentioned in section 6.1.4. In the following figures the objective value of the best (bestVRP), actual (actualVRP), new solution (newVRP) and initial solution (initial VRP) are visualized. The best solution is the objective value of the best solution found so far. The actual solution is the last solution, that was accepted by SA. The new solution is the one, that is generated in the according iteration – where a test of acceptance follows. The initial solution is the objective value of the generated initial solution.

Results of test instance 1 - test run 1

For test instance 1, as described in section 6.1.4, the first test run results with the outcome listed in the following paragraphs. In this test run the parameter q is set to 2, meaning that in each iteration two requests (pickup or delivery) are deleted from the VRP and two other ones are inserted. The results presented here reflect the solutions of a planning horizon that starts on September 23$^{\text{rd}}$, 2009, with the assumption that no timber has been hauled before that day. A solution for a planning horizon of three days is calculated, thus there are truck routes for the following days: 23$^{\text{rd}}$, 24$^{\text{th}}$ and 25$^{\text{th}}$ of September.

The initial solution, developed with construction methods, described in section 5.5.4, has an objective value of 17130.67. The best solution after 25000 iterations has an objective value of 19318.36. The mean value of all new generated solutions is 17202.27 with a standard deviation of 1708.66, and a median of 17794.70. With respect to the initial solution, the best found result reaches an increase of the objective function of 12.77%.

In order to elaborate on the detailed results, the first 1000 iterations are shown in figure 6.8. The figure shows that a good solution is found very early. In the course of the algorithm this solution is "destroyed" in order to generate other neighborhoods that are analyzed subsequently. Comparing the objective value for the actual and the new solution shows the acceptance of some solutions that are worse than the actual solution, according to Simulated Annealing (SA) described in chapter 3.4.3.

The situation after 2000 iterations is visualized in figure 6.9, where a good solution has been found in the first 1000 iterations. Nevertheless, a number of different configurations are analyzed, that finally lead to new best solutions in iterations 1637, 1639, 1640, 1645 and 1825. Between the "best" solutions there are very weak results, which are new neighborhoods that are developed – and, thus necessary. The results of the first 11300 iterations are displayed in 6.10, where the overall best solution is found.

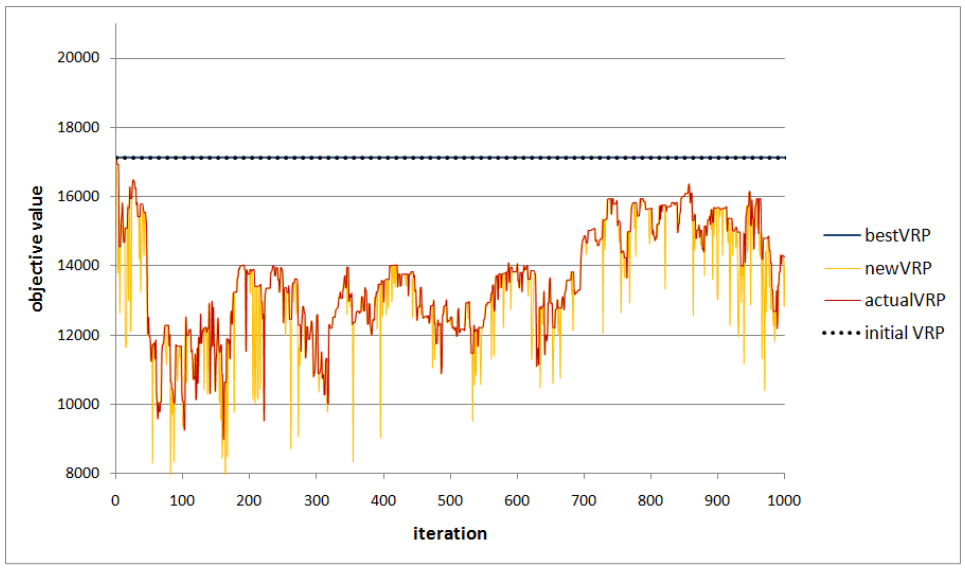


Figure 6.8: Result of $ALNS_{WSC}$ applied to test instance 1 (test run 1). The value of the objective function is displayed on the y-axis and the corresponding iteration number on the x-axis. The value of the best solution is in blue, the actual accepted (the one the $ALNS_{WSC}$ algorithm works with), which is denoted as the actual solution, is colored in red and the new solution in yellow. The value of the initial solution is marked with a dotted line.

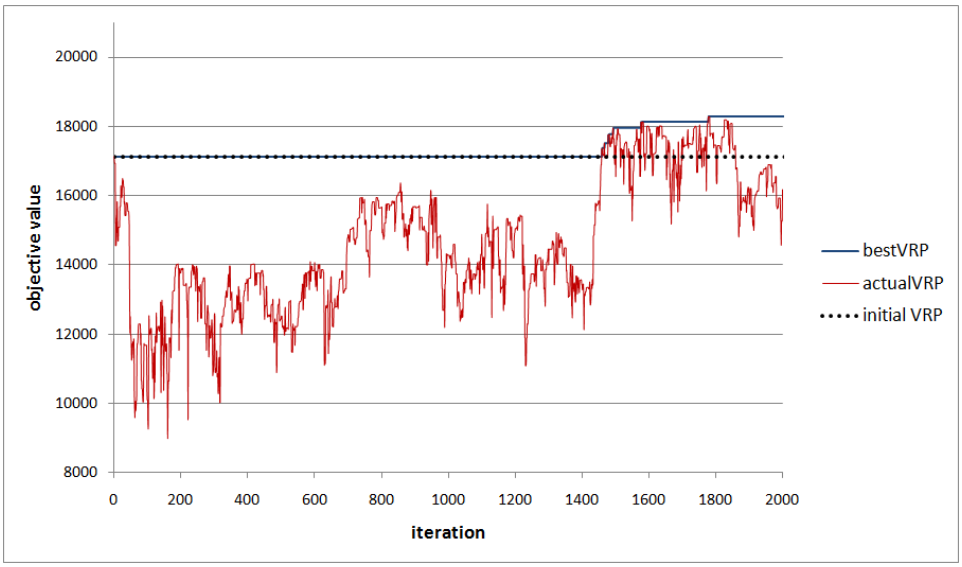


Figure 6.9: Result of ALNS_{WSC} applied to test instance 1 (test run 1) – here the first 2000 iterations of an algorithm run are displayed. The value of the objective function is displayed on the y-axis and the corresponding iteration number on the x-axis. The value of the best solution is in blue and the actual accepted (the one the ALNS_{WSC} algorithm works with), which is denoted as the actual solution, is colored in red. The value of the initial solution is marked with a dotted line.

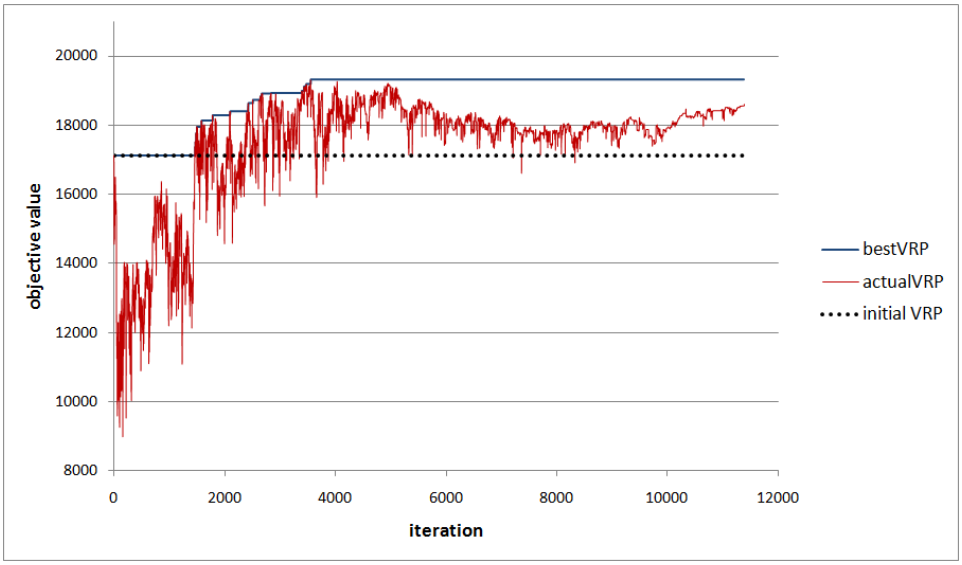


Figure 6.10: Result of ALNS_{WSC} applied to test instance 1 (test run 1) – here the first 11300 iterations of an algorithm run are displayed. The value of the objective function is displayed on the y-axis and the corresponding iteration number on the x-axis. The value of the best solution is in blue and the actual accepted (the one the ALNS_{WSC} algorithm works with), which is denoted as the actual solution, is colored in red. The value of the initial solution is marked with a dotted line.

Results of test instance 1 - test run 2

For test instance 1, as described in section 6.1.4 and the appendix, the second test run results with the following outcome. The results presented here (see figure 6.11) reflect the solutions of a planning horizon starting on October 21st, 2009, with the assumption that no timber has been hauled before that day. In this test run the parameter q is set to 2. A solution for a planning horizon of three days is calculated, thus we have truck routes for the following days: 21st, 22nd and 23rd of October.

The initial solution, developed with construction methods – described in section 5.5.4 – has an objective value of 897.64. The best solution after 25000 iterations has an objective value of 5090.90. The mean value of all new generated solutions is 4511.20 with a standard deviation of 491.40, and a median of 4655.40. With respect to the initial solution, the best found result reaches an increase of the objective function of 467.14%.

The overall result is displayed in figure 6.11. In this figure one can detect that the best solution is generated within the first 200 iterations – exactly in iteration number 164 (see figure 6.14). The situation after 500 and 1000 iterations, is visualized in figures 6.12 and 6.13, respectively.

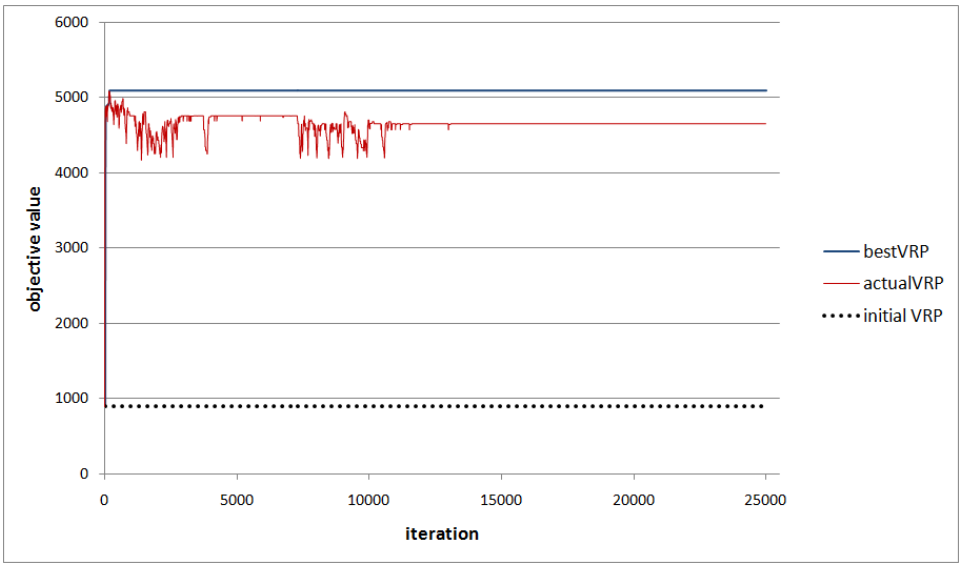


Figure 6.11: Result of ALNS$_{WSC}$ applied to test instance 1 (test run 2) – here all 25000 iterations are displayed. The value of the objective function is displayed on the y-axis and the corresponding iteration number on the x-axis. The value of the best solution is in blue and the actual accepted (the one the ALNS$_{WSC}$ algorithm works with), which is denoted as the actual solution, is colored in red. The value of the initial solution is marked with a dotted line.

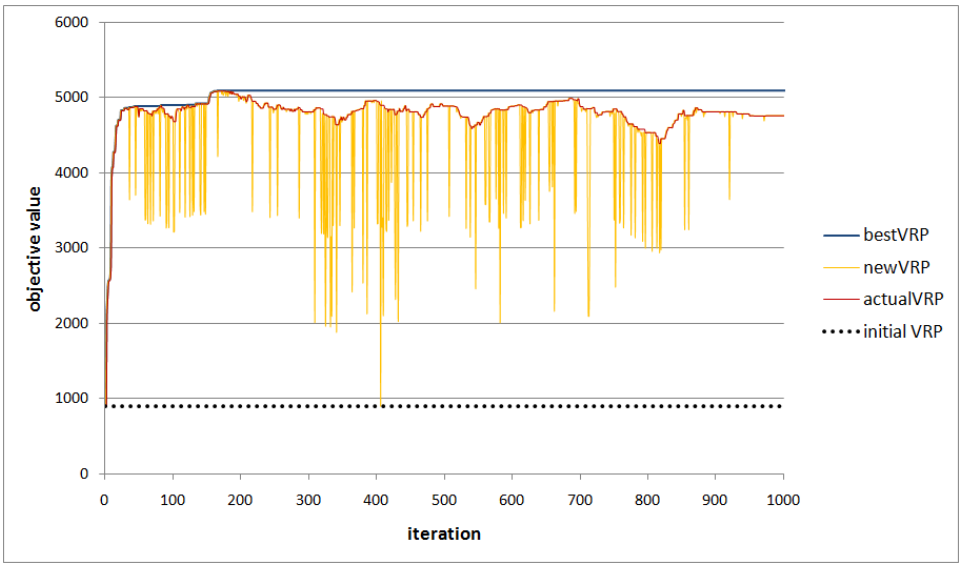


Figure 6.12: Result of ALNS$_{WSC}$ applied to test instance 1 (test run 2) – here the first 1000 iterations are displayed. The value of the objective function is displayed on the y-axis and the corresponding iteration number on the x-axis. The value of the best solution is in blue, the actual accepted (the one the ALNS$_{WSC}$ algorithm works with), which is denoted as the actual solution, is colored in red and the new solution in yellow. The value of the initial solution is marked with a dotted line.

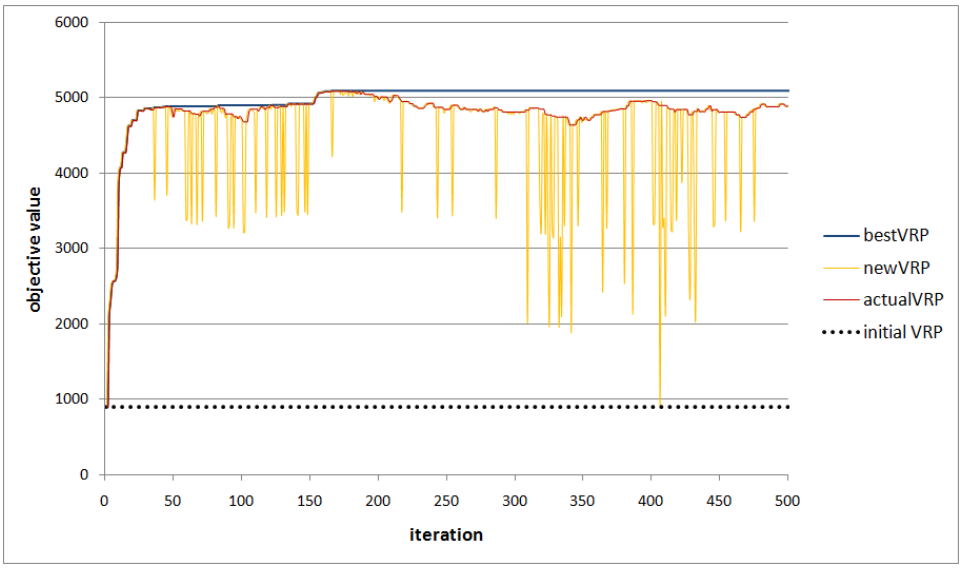


Figure 6.13: Result of ALNS$_{WSC}$ applied to test instance 1 (test run 2) – here the first 500 iterations are displayed. The value of the objective function is displayed on the y-axis and the corresponding iteration number on the x-axis. The value of the best solution is in blue, the actual accepted (the one the ALNS$_{WSC}$ algorithm works with), which is denoted as the actual solution, is colored in red and the new solution in yellow. The value of the initial solution is marked with a dotted line.

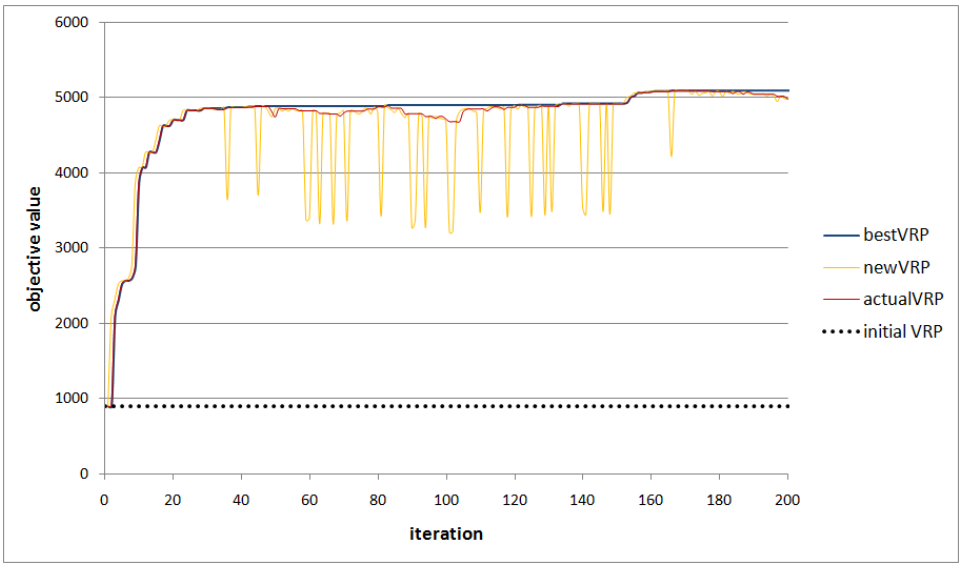


Figure 6.14: Result of ALNS$_{WSC}$ applied to test instance 1 (test run 2) – here the first 200 iterations are displayed. The value of the objective function is displayed on the y-axis and the corresponding iteration number on the x-axis. The value of the best solution is in blue, the actual accepted (the one the ALNS$_{WSC}$ algorithm works with), which is denoted as the actual solution, is colored in red and the new solution in yellow. The value of the initial solution is marked with a dotted line.

Results of test instance 1 - test run 3

For test instance 1, as described in section 6.1.4, the third test run results with the outcome listed in the following paragraphs. In this test run the parameter q is set to 4, meaning that in each iteration four requests (pickup or delivery) are deleted from the VRP and four (other) ones are inserted. The results presented here reflect the solutions of a planning horizon that starts on September 23$^{\text{rd}}$, 2009, with the assumption that no timber has been hauled before that day. A solution for a planning horizon of three days is calculated, thus there are truck routes for the following days: 23$^{\text{rd}}$, 24$^{\text{th}}$ and 25$^{\text{th}}$ of September.

The initial solution, developed with construction methods – described in section 5.5.4 – has an objective value of 17130.67. The best solution after 25000 iterations has an objective value of 19061.98. The mean value of all new generated solutions is 16081.22 with a standard deviation of 1780.89, and a median of 1661.45. With respect to the initial solution, the best found result reaches an increase of the objective function of 11.27%.

In order to elaborate on the detailed results, the first 5000 iterations are displayed in figure 6.15. There, the objective value of the best, actual and new solution is visualized. The figure shows that a solution, better than

the initial one is found in iteration 289. In the course of the algorithm the solution is "destroyed" in order to generate other neighborhoods that are analyzed subsequently. The situation after 5000 iterations is as visualized in figure 6.15, where one can detect that "good" solutions are found in the first 1000 iterations. Nevertheless, a number of different configurations are analyzed, that finally lead to the overall best solution in iteration 709 (see figure 6.17).

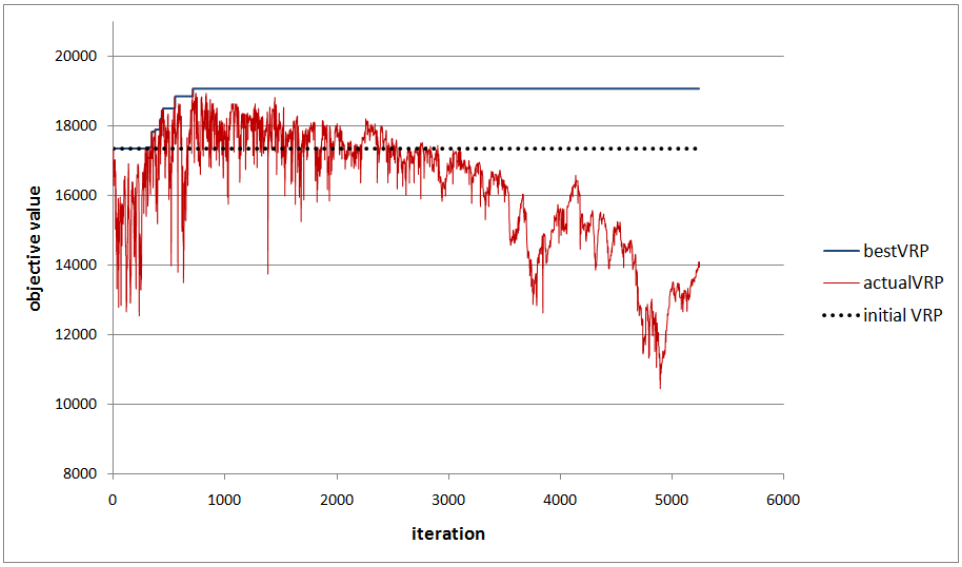


Figure 6.15: Result of ALNS$_{WSC}$ applied to test instance 1 (test run 3) after 5000 iterations. The value of the objective function is displayed on the y-axis and the corresponding iteration number on the x-axis. The value of the objective function is displayed on the y-axis and the corresponding iteration number on the x-axis. The value of the best solution is in blue and the actual accepted (the one the ALNS$_{WSC}$ algorithm works with), which is denoted as the actual solution, is colored in red. The value of the initial solution is marked with a dotted line.

Results of test instance 1 - test run 4

For test instance 1, as described in section 6.1.4, the fourth test run results with the outcome listed in the following paragraphs. In this test run the parameter q is set to 4, meaning that in each iteration four requests (pickup or delivery) are deleted from the VRP and four (other) ones are inserted. The results presented here reflect the solutions of a planning horizon that starts on October 21st, 2009, with the assumption that no timber has been hauled before that day. A solution for a planning horizon of three days is calculated, thus there are truck routes for the following days: 21st, 22nd and 23rd of October.

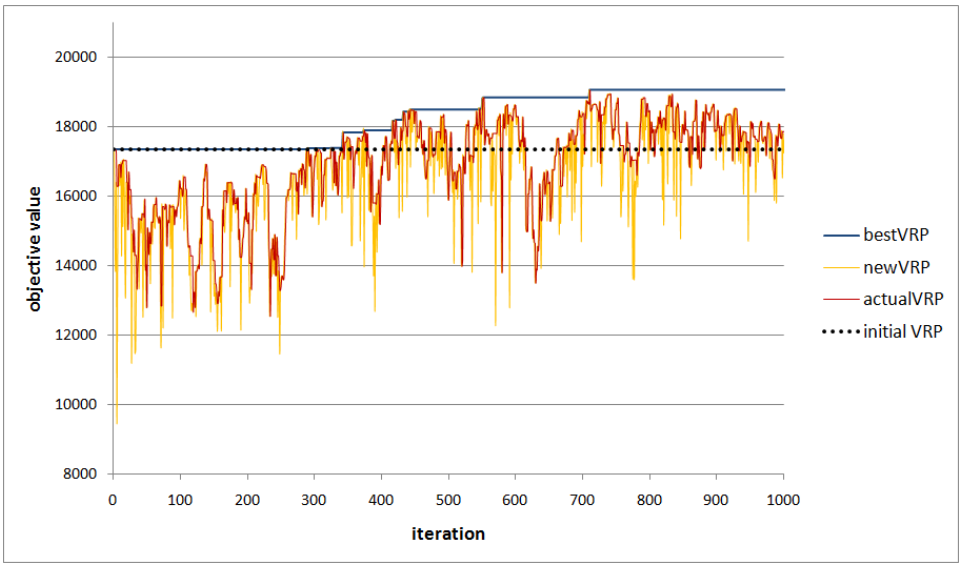


Figure 6.16: Result of ALNS$_{WSC}$ applied to test instance 1 (test run 3) after 1000 iterations. The value of the objective function is displayed on the y-axis and the corresponding iteration number on the x-axis. The value of the objective function is displayed on the y-axis and the corresponding iteration number on the x-axis. The value of the best solution is in blue, the actual accepted (the one the ALNS$_{WSC}$ algorithm works with), which is denoted as the actual solution, is colored in red and the new solution in yellow. The value of the initial solution is marked with a dotted line.

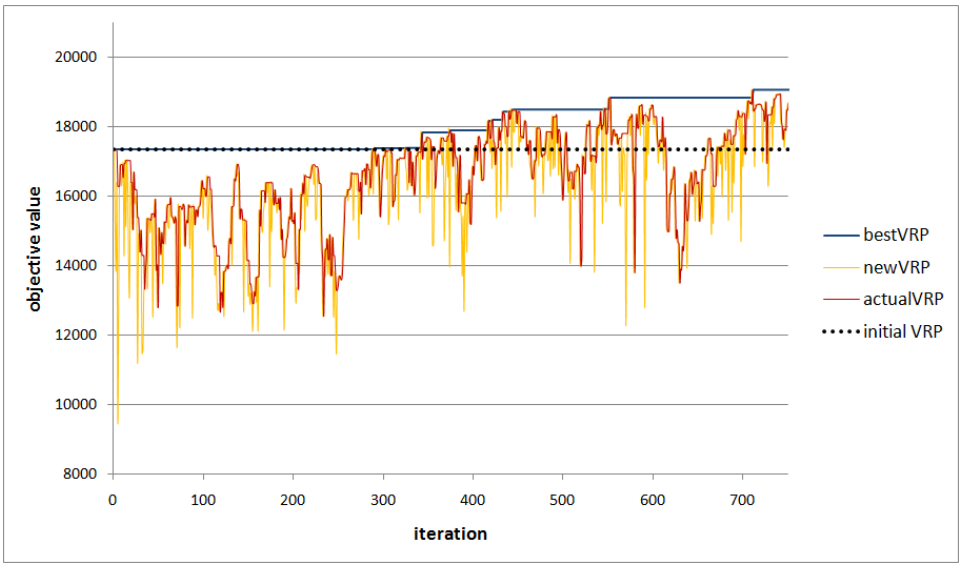


Figure 6.17: Result of ALNS$_{WSC}$ applied to test instance 1 (test run 3) after 750 iterations. The value of the objective function is displayed on the y-axis and the corresponding iteration number on the x-axis. The value of the objective function is displayed on the y-axis and the corresponding iteration number on the x-axis. The value of the best solution is in blue, the actual accepted (the one the ALNS$_{WSC}$ algorithm works with), which is denoted as the actual solution, is colored in red and the new solution in yellow. The value of the initial solution is marked with a dotted line.

The initial solution, developed with construction methods – described in section 5.5.4 – has an objective value of 897.64. The best solution after 25000 iterations has an objective value of 5060.06. The mean value of all new generated solutions is 3197.38 with a standard deviation of 924.24, and a median of 3004.29. With respect to the initial solution, the best found result reaches an increase of the objective function of 463.71%.

In order to elaborate on the detailed results, the first 5000 iterations are visualized in figure 6.18. There, the objective value of the best, actual and new solution is visualized. In the course of the algorithm the initial solution is "destroyed" in order to generate other neighborhoods that are analyzed subsequently. The situation after 5000 iterations is as visualized in figure 6.18, where one can detect that "good" solutions are found in the first 1000 iterations (see figure 6.19). Nevertheless, a number of different configurations are analyzed, that finally lead to the overall best solution in iteration 252 (see figure 6.20).

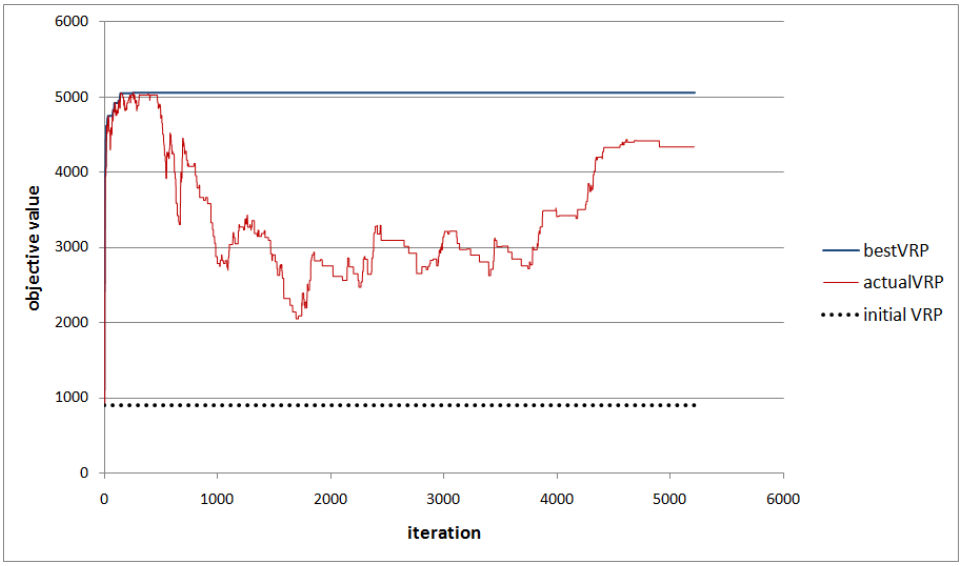


Figure 6.18: Result of ALNS$_{WSC}$ applied to test instance 1 (test run 4) after 5000 iterations. The value of the objective function is displayed on the y-axis and the corresponding iteration number on the x-axis. The value of the objective function is displayed on the y-axis and the corresponding iteration number on the x-axis. The value of the best solution is in blue and the actual accepted (the one the ALNS$_{WSC}$ algorithm works with), which is denoted as the actual solution, is colored in red. The value of the initial solution is marked with a dotted line.

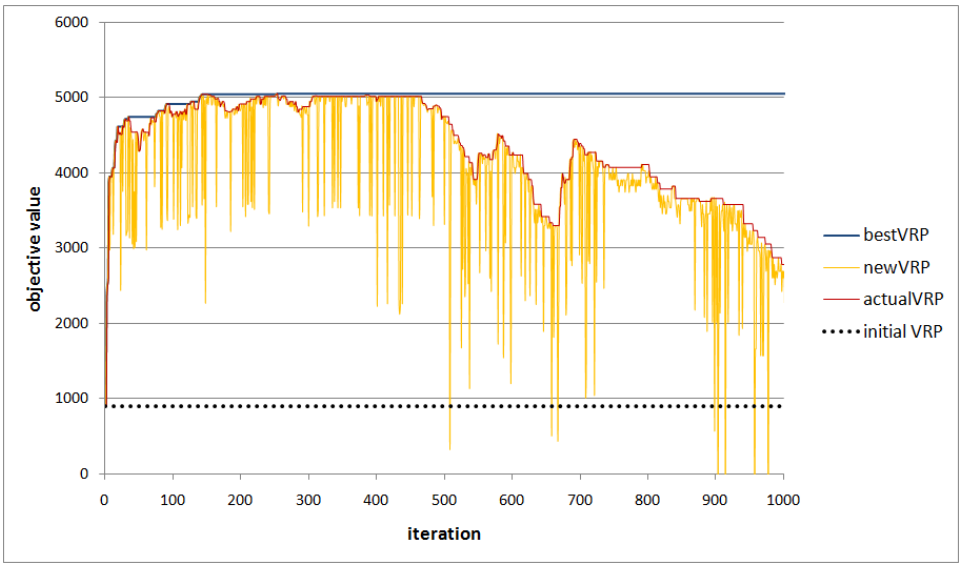


Figure 6.19: Result of ALNS$_{WSC}$ applied to test instance 1 (test run 4) after 1000 iterations. The value of the objective function is displayed on the y-axis and the corresponding iteration number on the x-axis. The value of the best solution is in blue, the actual accepted (the one the ALNS$_{WSC}$ algorithm works with), which is denoted as the actual solution, is colored in red and the new solution in yellow. The value of the initial solution is marked with a dotted line.

Results of test instance 2 - test run 1

For test instance 2, as described in 6.1.4 and in the appendix, the first test run results with the following outcome. In this test run the parameter q is set to 2, meaning that in each iteration two requests (pickup or delivery) are deleted from the VRP and two (other) ones are inserted. The results presented here reflect the solutions of a planning horizon that starts on September 23$^{\text{rd}}$, 2009, with the assumption that no timber has been hauled before that day. A solution for a planning horizon of three days is calculated, thus we may have truck routes for the following days: 23$^{\text{rd}}$, 24$^{\text{th}}$ and 25$^{\text{th}}$ of September.

The initial solution, developed with construction methods – described in section 5.5.4 – has an objective value of 223453.98. The best solution after 25000 iterations has an objective value of 223453.98 – the value of the initial solution. The mean value of all new generated solutions is 168410.50 with a standard deviation of 13124.47, and a median of 175132.91. The best solution found with the ALNS$_{WSC}$ algorithm has an objective value of 219196.25. With respect to the initial solution this reflects a decrease of the objective value of 1.905%.

In order to elaborate on the detailed results, the results of all 25000

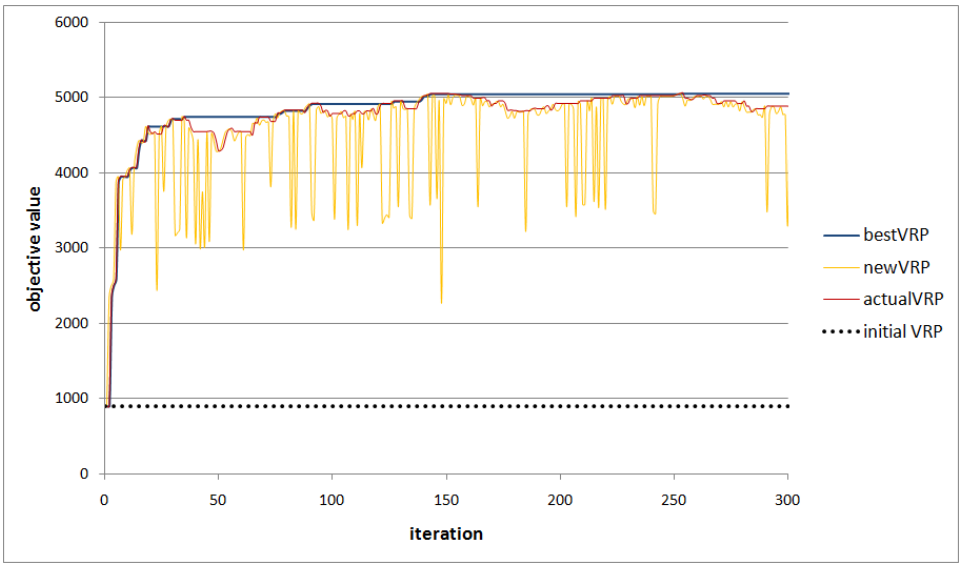


Figure 6.20: Result of ALNS$_{WSC}$ applied to test instance 1 (test run 4) after 300 iterations.

iterations in figure 6.21. There, the objective value of the best, actual and new solution is visualized. In the course of the algorithm the initial solution is "destroyed" in order to generate other neighborhoods that are analyzed subsequently. The situation in the last 2500 iterations is as visualized in figure 6.22, where one can detect that in the last iterations the objective value increases again. Nevertheless, the objective value of the initial solution can not be reached. Even in the first 3500 iterations a constant decrease of the objective value can be detected (see figure 6.23).

Results of test instance 2 - test run 2

For test instance 2, as described in 6.1.4 and in the appendix, the second test run results with the following outcome. The results presented here reflect the solutions of a planning horizon starting on October 21st, 2009, with the assumption that no timber has been hauled before that day. In this test run the parameter q is set to 2. A solution for a planning horizon of three days is calculated, thus there are truck routes for the following days: 21st, 22nd and 23rd of September.

The initial solution, developed with construction methods – described in section 5.5.4 – has an objective value of 160282.85. The best solution generated with ALNS$_{WSC}$ after 25000 iterations has an objective value of 158135.80. Hence, the ALNS$_{WSC}$ algorithm did not find a better solution. The mean value of all new generated solutions is 107413.23 with a standard

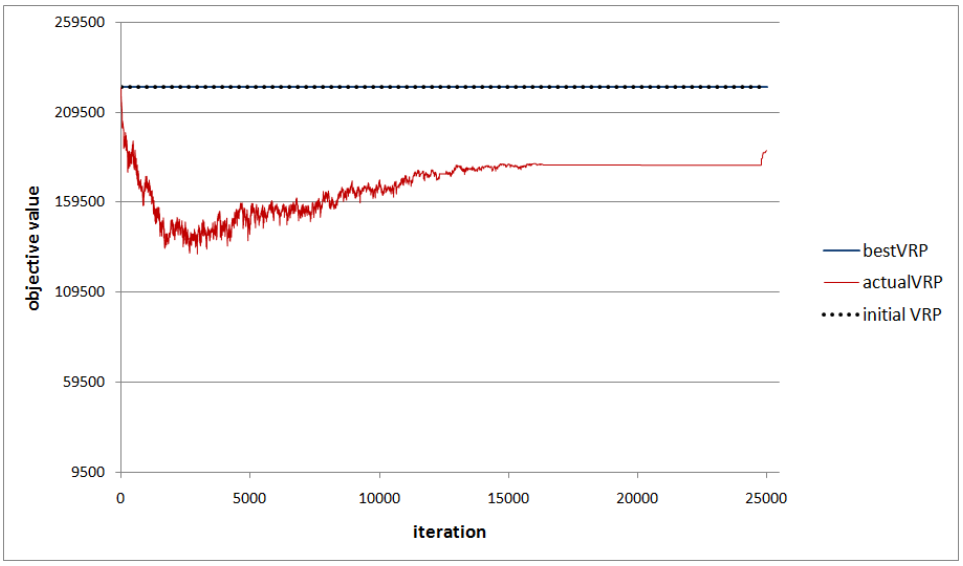


Figure 6.21: Result of ALNS$_{WSC}$ applied to test instance 2 (test run 1) after 25000 iterations. The value of the objective function is displayed on the y-axis and the corresponding iteration number on the x-axis. The value of the objective function is displayed on the y-axis and the corresponding iteration number on the x-axis. The value of the best solution is in blue and the actual accepted (the one the ALNS$_{WSC}$ algorithm works with), which is denoted as the actual solution, is colored in red. The value of the initial solution is marked with a dotted line.

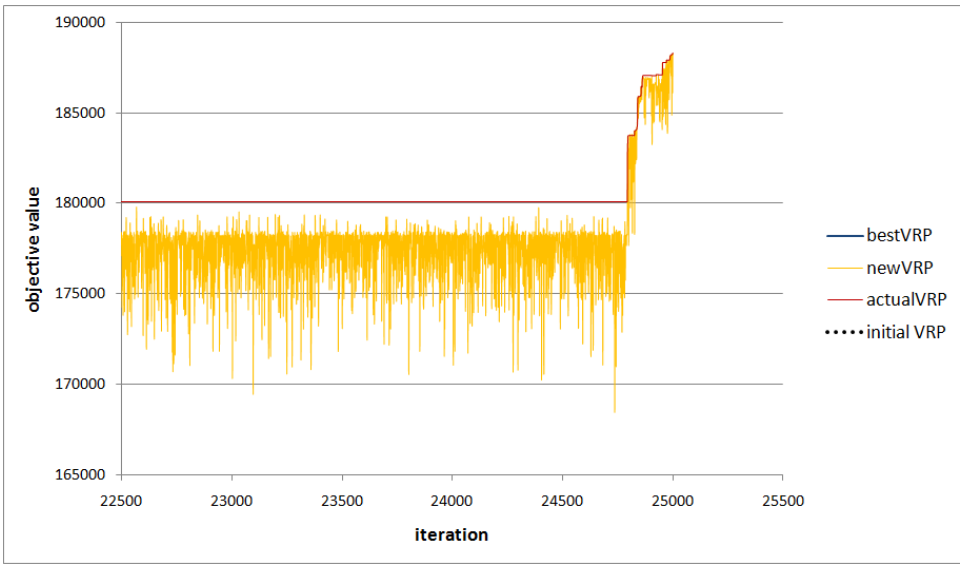


Figure 6.22: Result of ALNS$_{WSC}$ applied to test instance 2 (test run 1) – showing the last 2500 iterations. The value of the objective function is displayed on the y-axis and the corresponding iteration number on the x-axis. The value of the best solution is in blue, the actual accepted (the one the ALNS$_{WSC}$ algorithm works with), which is denoted as the actual solution, is colored in red and the new solution in yellow. The value of the initial solution is marked with a dotted line.

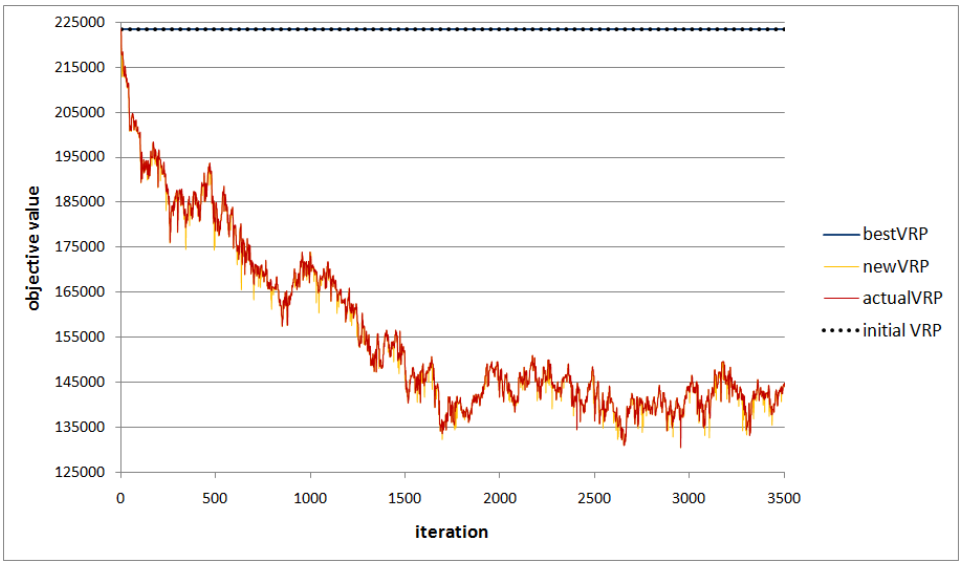


Figure 6.23: Result of ALNS$_{WSC}$ applied to test instance 2 (test run 1). Here the first 3500 iterations are visualized.

deviation of 9709.60, and a median of 111107.59. With respect to the initial solution, the best result found is 1.34% worse than the initial solution.

The situation after 20000 iterations is shown in figure 6.24. One can detect, that within the first iterations the initial solution is destroyed and new configurations are analyzed. This is clearly visualized in figure 6.26 – where the objective value of the generated solutions is decreasing. After 5000 iterations – see figure 6.25, the objective values are increasing again, indicating that better solutions are developed by the system.

6.3 CONCLUSION

This chapter elaborates on the experiment setup and the results of the real-time spatial optimization prototype for WSC optimization. In the beginning, the setup, which consists of the relevant hard- and software, the spatial data used and the developed problem instances, is described. For the evaluation of the ALNS$_{WSC}$ algorithm two test instances have been created, that are optimized by the prototype system at hand. The documented results show, that the spatial optimization of test instance one results in an increase of the objective function of 12.77% (test run 1), 467.14% (test run 2), 11.27% (test run 3), 463.71% (test run 4) compared to the initial solution. For test instance 2, no increase could be determined (test run 1: -1.905% and test run 2: -1.34% compared to the initial solution). The reasons for this behavior are discussed in chapter 7.2. Nevertheless, the system created is able to

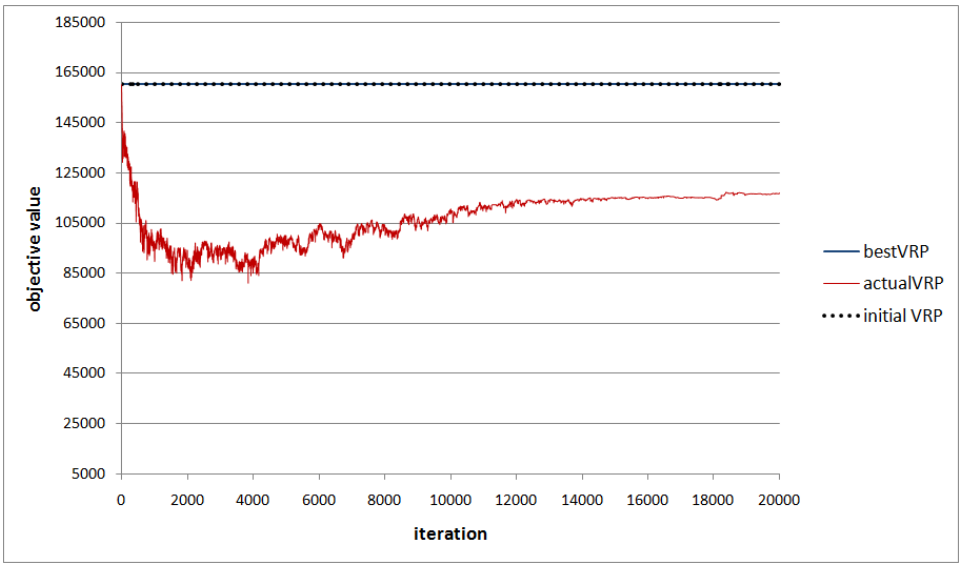


Figure 6.24: Result of ALNS$_{WSC}$ applied to test instance 2 (test run 2) after 20000 iterations. The value of the objective function is displayed on the y-axis and the corresponding iteration number on the x-axis. The value of the objective function is displayed on the y-axis and the corresponding iteration number on the x-axis. The value of the best solution is in blue and the actual accepted (the one the ALNS$_{WSC}$ algorithm works with), which is denoted as the actual solution, is colored in red. The value of the initial solution is marked with a dotted line.

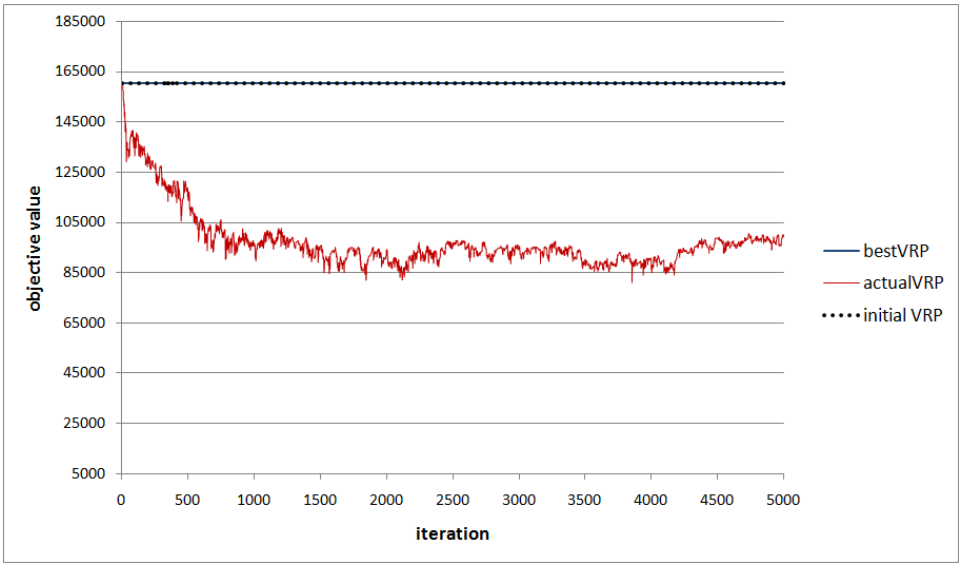


Figure 6.25: Result of ALNS$_{WSC}$ applied to test instance 2 (test run 2) – the first 5000 iterations are visualized. The value of the objective function is displayed on the y-axis and the corresponding iteration number on the x-axis. The value of the best solution is in blue and the actual accepted (the one the ALNS$_{WSC}$ algorithm works with), which is denoted as the actual solution, is colored in red. The value of the initial solution is marked with a dotted line.

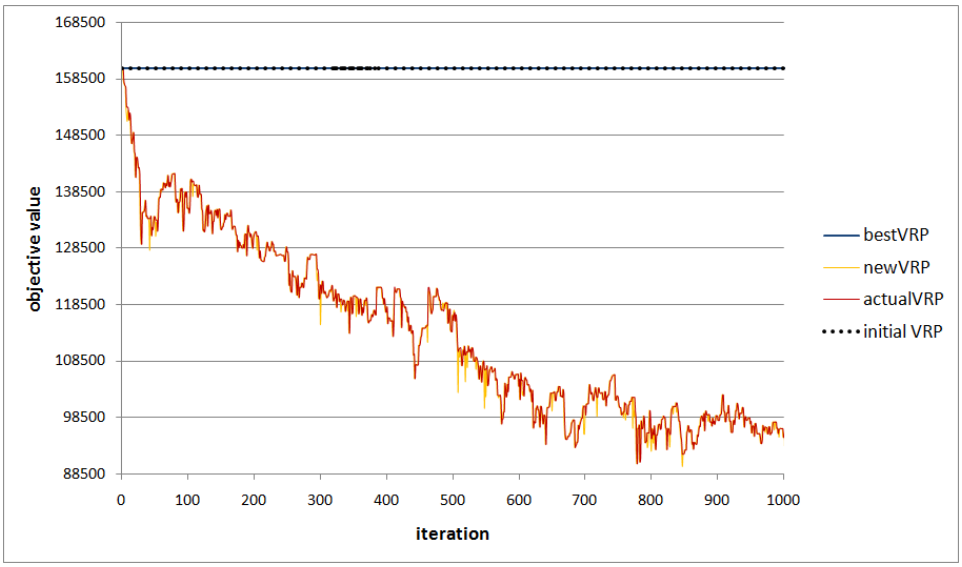


Figure 6.26: Result of ALNS$_{WSC}$ applied to test instance 2 (test run 2). Here the first 1000 iterations are visualized. The value of the objective function is displayed on the y-axis and the corresponding iteration number on the x-axis. The value of the best solution is in blue, the actual accepted (the one the ALNS$_{WSC}$ algorithm works with), which is denoted as the actual solution, is colored in red and the new solution in yellow. The value of the initial solution is marked with a dotted line.

optimize a given spatial problem at hand. If the ALNS$_{WSC}$ parameters are adjusted for large problem instances, the optimization process should optimize the given problem, as documented in Ropke (2005) for the general ALNS procedure. Chapter 7 addresses the optimal set of parameters for ALNS$_{WSC}$. In addition, a solution of such large instances is possible due to the fact that Ropke and Pisinger (2006) testet the ALNS algorithm on more than 350 benchmark instances with up to 500 requests. For more than 50% of the problems from literature, the procedure was able to improve the best known solutions.

Chapter 7

Discussion and Outlook

This chapter discusses the results documented in chapters 6.1 and 6.2 and justifies the chosen system architecture by comparing and evaluating alternative architectures. Furthermore, the implications on related fields are analyzed, as this optimization framework could be easily adapted to fit to other problems too. The future aspects of this thesis are discussed in the last section of this chapter.

7.1 Evaluation of Alternative Architectures

Before the actual implementation of the WSC prototype is discussed, the thesis elaborates on a number of different system architecture approaches and evaluate them according to their suitability for the optimization of the WSC and real-time spatial optimization. First, user needs are collected in a non-systematic way. The user needs are the result of numerous interviews with experts from Holzcluster Styria – a non-profit company that was founded as a consortium of the forest and timber industry in Styria that provides a rich expertise and develops innovative technology projects in these fields. Secondly, a procedure for quality assessment of system architectures is briefly described and applied to the alternative designs mentioned.

7.1.1 User requirements

The user needs for optimizing the WSC cannot be described in a few sentences. The requirements have to be collected separately for each stakeholder of the WSC (see chapter 1.2 and figure 1.3). The stakeholders are as follows:

- Saw mills
- Haulage companies
- Forest enterprises
- System administrator

Firstly, for saw mills, a continuous supply of timber is of highest priority. Secondly the timber origin is of high importance, due to accounting issues. Due to the fact that saw mills buy timber from piles at the forest road, they have to care and pay for the transport. Hence, a transparent charging of transport costs is a central claim of this stakeholder group. Thus, the transport costs should be calculated based on transport distance and/or transport duration.

Haulage companies and their truck drivers are the group that has to be treated with care, due to the drivers antipathy against additional computerized devices and tracking in general. On the contrary, the haulage companies would be interested in the actual positions and status of their trucks, without having to call them via e.g. mobile phones. Haulage companies do not want to have another set of expensive technology on their trucks, but do like rather cheap standard technology. In addition, haulage companies face the problem of finding drivers that are familiar with a certain forested region. Due to the fact that in most cases they have to find the timber piles on their own, the allocation of truck drivers to specific haulage tasks is another problem that has to be solved by the haulage companies. Thus, an "automatic" guidance to timber piles is of particular importance.

Forest enterprises require that all produced timber is transported to saw mills on time. Due to the fact that most foresters have a certain experience with computers and basic surveying and mapping techniques, they do not hesitate using additional computerized equipment. Due to the fact that a (forest) map is one of the most important working equipment of a forester, forest enterprises prefer having a map on the devices. A simple device and software to input, alter and to georeference timber piles accordingly is required with a self explanatory User Interface (UI). In addition, forest enterprises are interested in maximizing their profit in the supply chain, without loosing transparency. Furthermore, they do want to know which transports are planned – with relevant data and metadata – in advance.

Requirements of system administrators are different than the ones of other stakeholders, as they need to keep the system up and running. This requires regular maintenance operations and immediate handling of hard – and software failures. Nevertheless, they need a complete overview of the system and its performance in order to intervene where necessary. Thus, they need a UI for handling all these relevant issues.

In order to evaluate system architectures Mattsson *et al.* (2006) distinguish between two classes of requirements:

- functional requirements
- quality requirements

Functional requirements collect what the system should do in order to satisfy the users. Quality requirements describe what additional properties the system has to fulfill, e.g. easy portability or maintenance. In the user requirements listed above, functional and quality requirements are mixed. For the evaluation of alternative architecture designs both are of importance due to the fact that different systems allow different functionalities.

7.1.2 Quality Evaluation of System Architectures

In order to evaluate system architectures from a quality point of view a number of evaluation techniques have been developed that will be discussed later on. The focus lies on quality attributes of the architectures itself. Bosch (2000) divides quality attributes into two categories – development and operational quality parameters. A development quality is most important for the development and design of a system, whereas an operational quality reflects the user's point, e.g. usability. ISO (2001) – which was originally developed for general software quality – lists the following system quality parameters categories/views:

- Developer's view
- User's view
- Manager's view

The manager's view reflects the overall quality, efficiency and effectivity of a system. The user is more interested in the performance and how well it is usable. Developer's have to produce high quality software that fulfills the requirements defined by the users and managers. Before any quality attributes are listed the process steps of system architecture are highlighted as described in ISO (2001):

- Quality requirements definition: This part involves the definition of appropriate quality parameters.
- Evaluation preparation: This section requires the definition of a *metric* in order to measure the quality of software – which can be extended to system architectures (Côté *et al.*, 2004; Losavio *et al.*, 2004; Ortega *et al.*, 2003; Leung, 2001), as any software system needs some sort of architecture in behind. In addition, *rating levels* definition describe the ranges possible on the scales, including a rating distinction between a successful or unsuccessful score. Furthermore, an *assessment criteria definition* has the purpose of summarizing the results with respect to an individual weighting of quality criteria.

- Evaluation procedure: This stage is divided into three sub stages: measurement, rating and assessment. The first one evaluates the architecture and thus scores on the scale are obtained, followed by a rating level, which is the second stage. The following sub stage summarizes the results.

Alternative system architecture evaluation approaches are the Architecture Tradeoff Analysis Method (ATAM) (Kazman *et al.*, 1998) and its predecessor Software Architecture Analysis Method (SAAM) (Kazman *et al.*, 1996). These methods are not applied here, due to the fact they would require a set of experienced information technology experts and great temporal resources. Furthermore, these approaches are more suited to commercial software products, and thus, they are beyond the scope of this thesis. Nevertheless, if the prototype of this thesis would be realized as a commercial system the architecture described in section 5.1 and its alternatives would have to be analyzed using one of the two methods.

Quality parameters are the most important thing when it comes to evaluating a system. Thus, this thesis relies on two sources: a) IEEE (1990) that lists four attributes b) ISO (2001) that lists six attributes and attached sub attributes. The IEEE (1990) defines four development quality attributes, that are cited below:

- Maintainability:

  > "The ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment."

- Performance:

  > "The degree to which a system or component accomplishes its designated functions within given constraints, such as speed, accuracy, or memory usage."

- Testability:

  > "The degree to which a system or component facilitates the establishment of test criteria and the performance of tests to determine whether those criteria have been met."

- Portability:

  > "The ease with which a system or component can be transferred from one hardware or software environment to another."

Maintainability reflects the source codes understandability and readability. Thus, it is connected to the quality attribute testability, due to the fact that some – even isolated – parts of the system may be tested to fulfill requirements. For this purpose, parts of the source code have to be extracted or test routines have to be adapted in order to fit the system. A high degree of testability results in a quick validation of the system itself.

As described above, ISO (2001) lists six quality parameters that are partially mentioned in IEEE (1990). Nevertheless, a full list of the quality attributes and sub attributes is given here:

- Functionality
  - Suitability: This is the most important functionality characteristic and refers to the appropriateness of the software – meaning how well the specifications are met.
  - Accurateness: This is the correctness of functions implemented.
  - Interoperability: This concept is already described in chapter 2.1.1.
  - Compliance: If the software is developed with respect to appropriate standards or guidelines then it has a high rate of compliance.
  - Security: How well unauthorized access is prohibited in a system.
- Reliability
  - Maturity: This sub characteristic defines the error frequency in an architecture.
  - Fault tolerance: This describes the ability of a system to recover or withstand a certain amount of hardware or software failure.
  - Recoverability: This characteristic describes the system's capabilities to come back to full operation after a severe breakdown.
- Usability
  - Understandability: Understandability defines if the functions of a system are easy understandable and intuitively and manageable – thus, created with respect to Human Computer Interaction principles.
  - Learnability: Determines how fast and sustainable the system can be learned by different user groups.
  - Operability: This reflects the ability of the software to be operated by a user in a given environment.
- Efficiency

  - Time behavior
  - Resource behavior
- Maintainability
  - Analyzability: This parameter defines if it is easy to find the root cause of a system failure.
  - Changeability: This describes the effort and time necessary to change a system.
  - Stability: This parameter characterizes the ability of a system to work normally after a severe change in the system.
  - Testability: This defines the effort necessary to test a system.
- Portability
  - Adaptability: Defines the ability to adapt to new environments and/or specifications.
  - Installability: This parameter characterizes the effort necessary to install the software.
  - Conformance: The characteristic conformance relies to portability, as e.g. a conformance to the SFS enhances the portability of spatial data.
  - Replaceability: This sub characteristic reflects how easy it is to exchange a defined software component within a given environment.

Komarkova *et al.* (2007) mention – based on Ortega *et al.* (2003) – that every system evaluation does not necessarily have to work with the parameters mentioned in the prior listing. Thus, for evaluating different architectural designs the quality parameters usability and efficiency as well as the sub characteristic maturity are left out, as they have no direct relation to the evaluation of an architecture itself.

7.1.3 Possible system architecture designs

For the evaluation of different system architectures alternative designs are created. Due to the fact that numerous architectures exist, that differ only in minor details, the focus lies on three main design approaches: monolithic systems, SOA and ubiquitous computing approach. These designs will be discussed briefly and evaluated according to the quality parameters mentioned in section 7.1.2. The introduction into the system architectures relies on a number of resources (e.g. Taylor *et al.*, 2009; Vogel *et al.*, 2009).

Monolithic system architecture

Monolithic systems are system architectures that have ”everything” within ”one piece” (see figure 7.1). Thus, no modules or detached components are identifiable. Mostly monolithic systems are regarded as old mainframe systems present in computer centers in universities or companies. Nevertheless, such architectures are still alive, as e.g. word processing, development environments or spreadsheet programs prove.

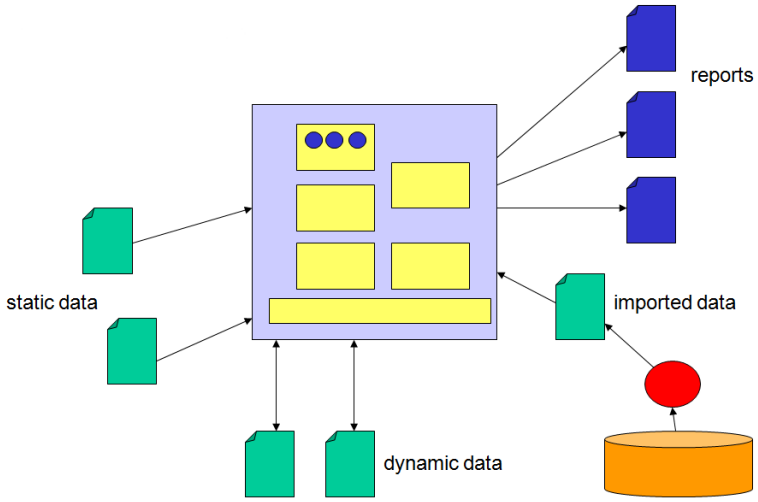


Figure 7.1: A typical monolithic architecture (Zalesky, 2003).

Monolithic system architectures are usually written in a single programming language – often only procedural – and compiled to a single application – thus, the name monolithic. These architectures may have a graphical UI or work in batch mode. Data are put in from various files or an underlying database, that is either fully integrated in the application or attached via some sort of ”translator” that enables data to be read into the application.

Problematic with monolithic systems is multi-user access, which is not possible or only with restrictions. Hence, simultaneous upload and data reading operations are only possible if done in a sequential order. Otherwise the system might show no accurate data.

Nevertheless, monolithic systems have advantages too, which will be explained now. First, performance is the main benefit of such systems, as data can be read in any desired and optimized format. Even data originating from a database can be migrated into a proprietary format that is specially tailored to the system. In addition, if most of the data are held in the memory data access is very fast. If the architecture does not have to deal

with multi-user access, then the design and coding is quite simple as the developer does not have to consider e.g. lockings, transactions or integrity issues.

Monolithic systems, may be "opened" by the use of middleware – first defined in a NATO report by Naur and Randell (1968, p. 14) – in order to communicate with other systems. Thus, it is possible to generate a monolithic architecture that may establish a connection to the "environment". For the evaluation of alternative architectural designs monolithic systems that are enhanced by middleware are used.

For a monolithic system architecture for WSC optimization there are (at least) two different options that will be discussed in the following paragraph and in figures 7.2 and 7.3. The architecture in figure 7.2 – denoted as option *A* – shows only a SDBMS that is directly connected to the main monolithic application. By the use of two middleware systems it is possible to exchange data with the mobile devices and the desktop system. Mobile devices should be located on the vehicles doing tracking and getting/sending data on timber to pickup or to deliver. In addition, the mobile devices present in the forest enterprises collect data on new timber piles. The desktop system does the central data visualization, and all optimization relevant parameters can be altered from this UI. In option *A* the middleware must serve a number of functionalities like data collection from mobile devices and (spatial) data providing. Additionally, the monolithic system, has to process data coming in from mobile and desktop systems via the middleware applications, sending it to the database, doing optimization tasks, serve maps and spatial data. Option *B* shows a slightly different approach than option *A*. Here, direct connections between mobile devices and the desktop systems send data directly to the database, and in turn receive data directly from the database without having to stress the monolithic system. Thus, the middleware and the monolithic system only need limited functionality, which in turn has to be shifted to the mobile and desktop applications. Hence, limited processing power on these devices is consumed by such functionalities.

These rather static architecture approaches are the reason that the software cannot be easily distributed over several machines. Thus, load balancing and a fail safe design is not realized. If a hardware or a piece of software would fail the whole system would fail. In addition, maintenance is also a difficult task, because there are no small components that can be exchanged which would be very easy. If errors happen, it is very difficult to analyze the possible root cause, due to a non existing partitioning of the system. If one wants to deploy that system – by an installation – this is a simple task because only one executable has to be copied to the target system. System security, with only one machine with limited connections to the environment, can be realized in an efficient way, because it is not necessary to secure a

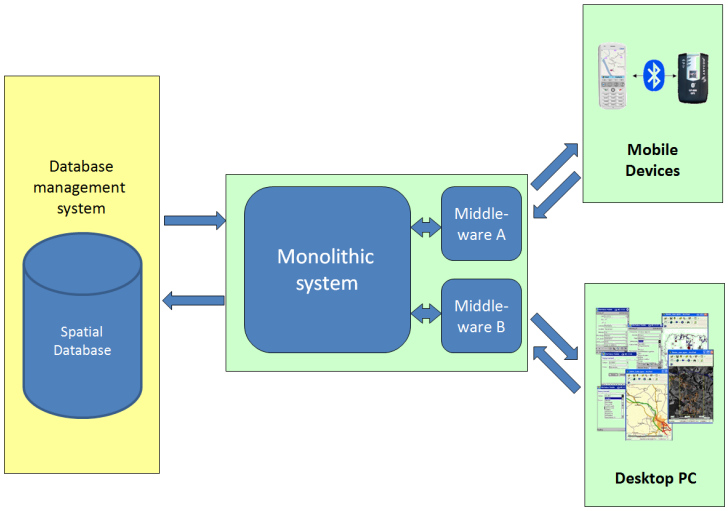


Figure 7.2: Monolithic architecture option *A* for WSC optimization.

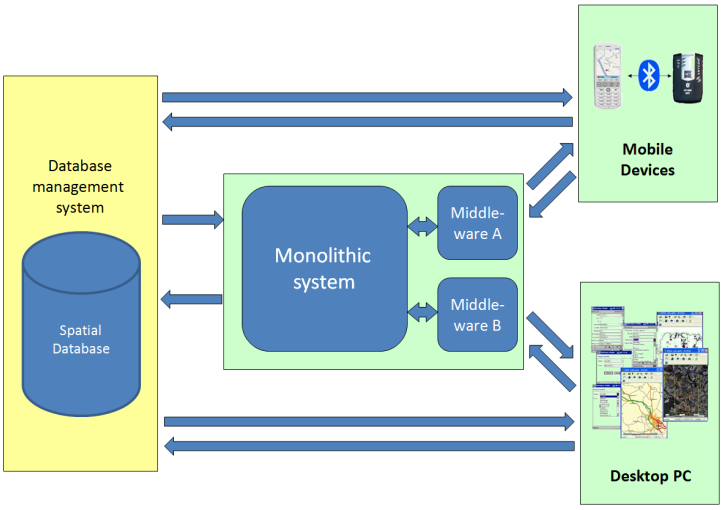


Figure 7.3: Monolithic architecture option *B* for WSC optimization.

number of communication channels on various machines.

Service Oriented Archtitecture

SOA, as described in chapter 2.2, will not be discussed in detail here. The implementation approach is shown in chapter 5.1, where figure 5.1 outlines the SOA used in this thesis. Here only the advantages and disadvantages of

this approach are mentioned, according to the quality parameters defined in section 7.1.2.

In a SOA the system is divided into a number of modules that fulfill a specified task and are connected over the Internet communicating with defined services and protocols. Thus, those modules can be distributed over several machines, resulting in a maintenance friendly, fail safe design. Especially for data storage it is possible to create database clusters that provide more safety for data stored in the system. In addition, errors can easily be traced due to the clearly distinguishable module functionality. Due to the flexibility that SOAs offer it is easy to change modules, or to have one module type running in parallel on more machines in order to provide a backup system. For security reasons the great number of connections, communication protocols and hardware environments may be a mess, resulting in a great effort to secure the modules properly. Nevertheless, setting up such a system can become complicated as a number of different applications have to be installed and the communication between them has to be established.

Ubiquitous computing architecture

The term Ubiquitous Computing (UbiCom) has been coined by Mark Weiser (Weiser, 1991) and involves a concept where "computing is everywhere" using small, cheap, network enabled devices (e.g. Poslad, 2009). Weiser (1991) proposed three main properties of ubiquitous systems:

- Computers need to be networked, distributed and transparently accessible
- Computer Interaction with humans needs to be more hidden
- Computers need to be aware of environment context

The properties have been extended:

- Computers can operate autonomously, without human intervention – thus, being self-governed
- Computers can handle a multiplicity of dynamic actions and interactions, governed by intelligent decision-making and intelligent organizational interaction; this entails some form of artificial intelligence;

In order to come up with an architecture, hardware components are defined that build up the system itself. In literature a number of devices are mentioned (e.g. Poslad, 2009; Weiser, 1991):

- Tabs: smallest components (centimeter sized), that are wearable
- Pads: devices that are decimeter sized, and may be handheld devices – comparable with a PDA
- Boards: display devices that have a size of one to several meters
- (Smart) Dust: tiny devices, without a graphical UI – e.g. micro electro-mechanical systems (MEMS); have a size from nanometers to millimeters.
- Skin: display devices on/directly in fabrics
- Clay: groups of MEMS even in three dimensions

With the help of this devices it is possible to create high dynamic and real-time architectures that immediately react to their environment. A number of today's devices have a number of the ubiquitous computing principles already "implemented" – examples are: mobile phones (especially smartphones), PDAs, GPS receivers or radio frequency identification tags (RFIDs). For WSC management ubiquitous computing could be realized by having smart dust or tabs applied to each single timber log, with according data and metadata. As a consequence, each single timber log can be tracked on its way from the forest to the saw mill. In addition, vehicles have a combination of smart dust and pads on them in order to automatically determine the state of the vehicle – e.g. the loading status or the action performed now (driving/loading/...). Having this data are collected in an automatic way and sent in real-time to the system which shows the modular architecture known from SOA. Clearly, everything is communicating utilizing services (Poslad, 2009). Additionally, sensors – e.g. tabs, dust, clay – can be grouped together as sensor nets, acting as one distinct entity in the system. Research and standardization work in this field is done by OGCs sensor web enablement working group (http://www.opengeospatial.org/projects/groups/sensorweb). A number of standards and discussion papers (e.g. OGC, 2007b,d,e,f,g, 2006b, 2003) have been published by this group, which are mentioned here, but not discussed in detail.

The advantages of such a design are that it automatically collects data and sends them to the central database, using services. In addition, through the SOA it is possible to create a very maintenance friendly and fail safe system. Through the modularized configuration there are similar advantages as a pure SOA, discussed in section 7.1.3. Disadvantages, as identified by the author might be hardware failures of sensors, due to rough conditions in the forest. In addition, energy supply might be critical for such devices located on timber logs – as they might have to withstand several weeks in

the nature. Another negative issue of ubiquitous computing is the fact that – similar to SOAs – communications channels and services have to be secured, which can result in a great effort due to numerous hardware and software environments.

7.1.4 Evaluation of System Architectures

In order to evaluate the the system architectures mentioned in section 7.1.3, only a subset of quality parameters listed in section 7.1.2 is used. Due to the fact that a number of parameters are dependent on the actual implementation – rather than the architectural concept itself – they are left out in this consideration – e.g. time and resource behavior is dependent on the programming style and usability issues can as well only be evaluated based on a concrete implementation. Hence, the following quality parameters are chosen:

- Functionality:
  - Suitability
  - Accurateness
  - Interoperability
  - Compliance
  - Security
- Reliability:
  - Fault tolerance
  - Recoverability
- Maintainability:
  - Analyzability
  - Changeability
  - Stability
  - Testability
- Portability:
  - Adaptability
  - Installability
  - Conformance
  - Replaceability

To rate the quality parameters a scale ranging from 1 - 10 is used, where 1 reflects a poor performance and 10 the best performance concerning to the mentioned quality parameter. The scores 1-5 are defined as "not successful" and the scores 6-10 are "successful" respectively.

The evaluation of the system architectures resulted in table 7.1. There the scores of each alternative are listed and can be compared. Both monolithic system architectures reach a result of 50 (Monolithic A) and 58 (Monolithic B) of 150 points. According to this evaluation, SOA and UbiCom are the best alternatives out of the discussed ones. Nevertheless, both have their weaknesses, which are described in section 7.1.3. A big drawback of SOA and UbiCom is security, as a number of connections have to be secured. SOAs are slightly better than UbiComs due to their lower number of "computers" involved. In addition, the installation of a number of applications that are communicating with each other can be time consuming. Due to the fact, that UbiComs have a great number of hardware components they have a lower rating than SOAs. Both monolithic architectures have the advantage, that only one executable has to be installed – plus the middleware –and everything should work properly. The small, well defined "programs" of SOA and UbiCom make testing and analyzing easy. Moreover, it is possible to increase the fault tolerance ans stability because with SOA and UbiCom not the whole system stops if a component has failed.

Given the discussion in the previous paragraph, the chosen architecture is SOAs. It is chosen because this architecture reaches the best scores in the evaluation. Additionally, a number of more practical arguments indicate the use of SOA. The number of necessary hardware components is high, which in turn increases the investments for such a system. Nevertheless, the applicability of e.g. RFID in forestry has to be proven, due to the rough conditions in forestry – although first studies show that RFID is applicable in forestry (e.g. Korten and Kaul, 2008). A research project dealing with RFID in timber logistics (Fraunhofer IFF, 2009) addresses these questions. In addition, a pure UbiCom might not be fully accepted by the community, as some data may be altered/put in by the truck drivers – especially as components (e.g. sensors on timber logs) might get broken. Generally speaking, SOA paves the way for UbiCom as this architecture relies on SOAs concepts. Thus, after successfully launching a system based on SOA this can be expanded by UbiCom parts.

7.2 Discussion of Results of ALNS applied to WSC

With respect to the results documented in section 6.2.2, a critical discussion is carried out in this chapter. Here, a detailed analysis of the obtained results

Parameter	Monolithic *A*	Monolithic *B*	SOA	UbiCom
Suitability	6	7	10	10
Accurateness	5	7	10	10
Interoperability	1	1	9	10
Compliance	1	1	10	10
Security	10	9	3	1
Fault tolerance	1	2	8	7
Recoverability	2	2	8	8
Analyzability	1	3	10	10
Changeability	3	4	9	10
Stability	1	1	8	8
Testability	3	3	10	10
Adaptability	2	2	9	9
Installability	10	10	3	2
Conformance	2	3	8	8
Replaceability	2	3	10	10
Overall score	**50**	**58**	**125**	**122**

Table 7.1: Evaluation of System Architectures.

is done, that tries to find reasons for the performance of ALNS_{WSC} as well as a comparison of the results as such.

Two test instances, that are described in section 6.1.4, are evaluated using ALNS_{WSC} that differ in their size. Thus, the application finds the best results after a different number of iterations. The test run 1 (starting at 23$^{\text{rd}}$ of September, 2009) of test instance 1 has a size of 43 nodes and 35 vehicles respectively. Test run 2 (starting at 21$^{\text{st}}$ of October, 2009) of the same instance has a size of 24 vertices and 35 vehicles. Test instance 2 – test run 1 (starting at 23$^{\text{rd}}$ of September, 2009) comprises 93 nodes and 35 vehicles. Test run 2 of this test instance (starting at 21$^{\text{st}}$ of October, 2009) has a size of 64 nodes and 35 trucks.

As stated in the previous paragraph, the size of the instances and parameter q – that controls the number of nodes to be removed and inserted in each iteration – has influence in the "speed" of the optimization. The "speed" is represented here not in terms of time, but means how "early" – in which iteration – the best solution is generated – described in the next paragraph. The iteration in which the best solution is found is displayed in table 7.2. In this table one can detect, that the bigger the problem size is, the "later" the optimal result is generated – for $q = 2$. Furthermore, the setting of q in combination with the problem size has a vital influence on the algorithm itself. Thus, two new test runs (number three and four) of test instance 1 have been conducted. These test runs have equal settings to test runs one and two, but the parameter q is set to 4.

The reason for this behavior can be easily explained. If the problem size is very big, the system is able to generate a number of new configurations – the sequence of nodes in each route – that might have a lesser objective value than the objective solution. This is possible, due to the "high" start temperature of SA at the beginning that accepts worse solutions. In the course of the ALNS algorithm the temperature is reduced, which lowers the probability of acceptance of new solutions that have a smaller objective value than the actual solution. Hence, the algorithm is not allowed to analyze different new configurations by a reordering of the nodes. Therefore, the ALNS algorithm has to "visit" different solutions before the temperature is too low.

For big problem instances a greater value of q could solve the problem. By setting $q > 2$ a greater number of nodes is deleted and inserted again. Thus, the initial solution is altered faster compared to $q = 2$, where different settings can be analyzed. Hence, it seems advisable to alter the parameter q depending on the problem size to be optimized. For small problems $q = 2$ is still a good setting (see table 7.2), as "optimal" solutions are reached early in the optimization process. In addition, if $q = 4$ is chosen for small problems – as test instance 1, test run 3 shows (see table 7.2) – it might take longer to come up with a "good" solution. The reason is that a fruitful solution is destroyed in the next iteration because four nodes have to be removed from this solution. Thus a low value of q is appropriate for small instances. Parameter tuning, which determines the optimal parameters for a given problem size is subject of future research activities – as stated in section 7.4.

test instance	**test run**	q	**# nodes**	**# vehicles**	**# iteration**
1	1	2	43	35	1825
1	2	2	24	35	164
1	3	4	43	35	709
1	4	4	24	35	252
2	1	2	93	35	n.a.
2	2	2	64	35	n.a.

Table 7.2: Test instances and according iteration of the best achieved optimization result – denoted as # iteration.

Despite the fact that in test instance 2 ALNS_{WSC} was not able to generate a better solution than the initial one, it is generally possible to optimize the WSC with real-time data from mobile devices. Nevertheless, with an adapted parameter set for ALNS_{WSC} – that has to be determined – the system is able to provide optimized routes for vehicles – that are delivered on demand, but are preprocessed beforehand and stored in the spatial database.

For test instance 1 the objective value of the initial solution can be increased by 12.77% (test run 1), 467.14% (test run 2), 11.27% (test run 3), 463.71% (test run 4).

Generally speaking, by the application of ALNS in the WSC the timber market could be altered. In the contemporary timber market timber is sold to saw mills often before timber is even harvested or after it is hauled to the next forest road. With the approach presented here, it is not possible to determine to which saw mill the timber is sold beforehand. In the course of ALNS_{WSC} timber is dynamically assigned – i.e. dynamically sold – to saw mills. Thus, the actual timber market is altered drastically to a *Dynamic Timber Market*. Only because of this drastic change of the timber market assumed here the optimization system presented in this thesis can be realized.

7.3 Implications of real-time spatial optimization on related fields

The obtained results of the research documented in this thesis is of interest for optimizing the WSC as such. Nevertheless, the reader might think of related "real world" problems that could be optimized using the findings of this thesis. Generally speaking, a combination of innovative concepts, like SOA and Operations Research (OR), makes it possible to exchange data in real time and optimize a given mathematical problem at hand.

In order to evaluate the implications of real-time spatial optimization it is necessary to look at the nature of the problems addressed in this thesis. In chapter 3 the relevant problems are listed. In general, the problems that are subject of the research carried out are VRPs. Especially such VRPs are of interest, where goods have to be transported – i.e. picked up and delivered – and/or customers have to be visited within a given time interval. Due to the fact that a system has been created that additionally allows a dynamic timber brokerage – based on timber prices and transport costs – it is possible to "solve" simpler problems too. A number of related fields can be targeted by such a system, that need routing and/or dispatching and/or scheduling. Hence, related fields are to be found in e.g. general vehicle based transportation (fleet management), garbage disposal, beverage industry, air/ship transportation, scheduling of service technicians, etc. Of importance for all mentioned fields are the following items:

- current *position* of vehicles/goods/service technicians – if possible in real-time
- *status* of the vehicles/goods/service technicians – if possible in real-

time

- *spatial optimization* and *scheduling*
- *routing* functionalities for vehicles/service technicians
- *mapping functionalities*

The application and results (see chapters 5 and 6) show that by the use of LBSs and ISO/OGC standardized services it is possible to ensure an efficient data flow from mobile devices, that may be located in e.g. vehicles. Thus, data can be obtained in real-time that indicate the current position as well as the actual status. Standardized services, originating from ISO and OGC – described in chapters 2.1 and 2.3 – are of immanent importance for displaying the collected data and routing functionalities. By using them, it is easy to create Spatial Data Infrastructures (SDIs) that allow a simplified exchange of spatial and attributive data overcoming institutional borders. Additionally, mapping and routing services can be built up, that can be used by others if desired, which in turn helps different institutions to cooperate in an efficient way. The SOA developed in this thesis facilitates this approach. Hence in SOAs it is practicable that more institutions share resources – in terms of e.g. a shared spatial database, or a shared routing service.

Optimization and scheduling are of particular interest in order to calculate the "best" solution for a given problem. As exact techniques are only efficient when optimizing comparable small problem instances, heuristics and metaheuristics are suitable for solving large problem instances (see chapter 3 for details). Nevertheless, by using approximative methods to solve such tasks it is possible to create "good" solutions for a problem – e.g. VRP, scheduling. The term "good" relates to a given objective function, which could be e.g. minimize CO_2 emissions, minimize travel distance, maximize profit. In addition, if the concept of SOA is applied in this sector too, it is possible to create services that could be shared among more institutions similar to OGCs Web Processing Service (OGC, 2007h).

In the following paragraphs some related application fields are mentioned that could be "optimized" using the results and methods of this thesis. First, a fleet of service technicians is of interest, as they are driving to customers with vehicles. Additionally, every customer wants to be serviced within a given time interval – e.g. from 13 o'clock until 15 o'clock. Taking into account a certain service time, a SDSS – with according mathematical models in behind – is capable of developing optimal solutions – meaning an optimal route and schedule for every service technician.

Waste disposal optimization can also be optimized by a similar system. Due to the fact, that waste has to be picked up within given time intervals

and delivered to a certain dumping ground or recycling facility if the truck is fully loaded, this problem is similar to the WSC. It is clear that the loaded amount of waste cannot be modeled with high accuracy, due to varying filling levels of garbage containers. Thus, the system would have to be able to cope with fuzzy data and fuzzy modeling and data series analysis, which is not part of this thesis.

7.4 Future aspects

Although this work attempts to show all aspects of real-time spatial optimization (based on the application in WSC management), some issues may be improved. In addition, a broader test of the application itself followed by an evaluation of the obtained results would be of interest. Nevertheless, this chapter lists and discusses the issues that could lead to a refinement and/or to emphasize other aspects of this work.

A comparison with "the real world" would give the obtained results more comparability. Thus, real world data of the WSC should be collected and evaluated. Therefore, it is necessary to monitor the vehicles with e.g. GPS, to gather data from forest enterprises and saw mills concerning demand and supply. This has to be followed by a thorough analysis of the real world data with regard to cost and profitability. Based on the supply and demand data VRP instances could be constructed which could be optimized. Comparing the real world data with the optimized results, could give detailed insight on how well such a system performs under real conditions.

As stated in chapter 7.3, similar problems from other fields of expertise could also be handled. Hence, it would be interesting to adapt the current approach to fit to other problems. Future scientific works could try to broaden the application areas of the system at hand.

As the Model Base and Model Base Management System in its actual configuration have only one mathematical model – ALNS_{WSC} – the system and the user are not capable of switching between different mathematical models. This could be enhanced by incorporating more mathematical models and optimization algorithms. Hence, exact algorithms could be tested concerning their applicability in this context, and the results of other heuristics could be compared with ALNS. These results and the system itself could be an interesting technology sandbox for researchers from mathematics and GIS&T.

Some more detailed improvements include the parameter tuning of the developed system – mostly the parameters of ALNS_{WSC}. By evaluating the results of different parameter sets – e.g. start temperature, cooling rate, q

the number of node removal and node input, p the degree of randomness – one could select the best parameters for a given problem. The results of this further research would be interesting especially for large problem instances, as they require more computing resources. By optimizing the model parameter the system could reach a "good" result within less iterations.

A more "global" improvement is the calculation of vehicle costs and the definition of the objective function itself. The vehicle costs could be determined based on the working hours of a vehicle – in contrast to the actual approach, that is directly related to the traveled distance. With a temporal cost approach the costs of different vehicle classes (e.g. semi-trailer truck, drawbar trailer combination) could be included too, which models the costs more accurate.

By a redefinition of the objective function the system could try to minimize the CO_2 emissions of the WSC. This would fit into current global efforts to save the climate. Therefore, the CO_2 emissions of a truck, with regard to road inclination, loading, engine specifications, etc. would have to be modeled and integrated into the system. Generally, this would be a novel approach for optimizing a Supply Chain Management System (SCM-system).

Another global improvement would be a change of the system architecture itself. As discussed in chapter 7.1, UbiCom was identified as one of the best architectures that could be applied to real-time spatial optimization. If security issues are resolved and the practicability of the hardware is proven this architectural approach is innovative, given the real-time data that could be gathered by the e.g. smart dust hardware. In order to build up a system based on UbiCom a SOA is the underlying basis. Hence, the results of this thesis pave the way for a future architecture that may be based on UbiCom.

7.5 Concluding Remarks

This thesis shows the potential of real-time spatial optimization that is a combination of Geographic Information Science and Technology (GIS&T) and Operations Research (OR) that are combined within a Spatial Decision Support System (SDSS). A prototype for the validation of this concept is developed that optimizes the Wood Supply Chain (WSC). The generated system addresses the drawbacks of contemporary WSC management. By the application of Service Oriented Architectures (SOAs) and standards (OGC and ISO) data from stakeholders can be collected and processed in real time, thus avoiding media breaks by a single, consistent database.

In this thesis a change of the the contemporary timber market is assumed.

In todays approach timber is sold to a defined saw mill – sometimes often before timber is harvested. In order to optimize the WSC with the results of this thesis, it has to be assumed that timber is sold dynamically. Hence, harvested timber is assigned dynamically – i.e. dynamically sold – to saw mills in order to optimize profitability of the supply chain.

The WSC can be mathematically modeled as Vehicle Routing Problem (VRP). In order to optimize the WSC– and the VRP – as such, ALNS is selected as appropriate optimization methodology. This method is adapted in order to fit to the WSC – and, thus the resulting mathematical model is named ALNS$_{WSC}$. Generally speaking, ALNS is a heuristic that destroys a solution to a given VRP by removing nodes and in turn "repairs" the VRP by inserting other nodes.

The analysis of the results of real-time spatial optimization show that this concept is able to optimize a given spatial problem with real-time data from mobile devices. In addition, this approach is able to deliver the results on demand. The results have to be preprocessed in the database, in order to be available for stakeholders of the problem at hand.

In addition, it is possible apply this concept to related spatial problems or fields of expertise. The key concepts GIS&T, SOA, OR combined in a SDSS offer new opportunities for solving complex spatial problems that could be analyzed in further research projects.

Bibliography

Adams, T., Koncz, N. and Vonderohe, J., 2000, Functional requirements for a comprehensive transportation location referencing system. In *Proceedings of the 2000 North American Travel Monitoring Exhibition and Conference* (Madison, Wisconsin: Transportation Research Board). [21]

Ahuja, R.K., Ergun, O., Orlin, J.B. and Punnen, A.P., 2002, A survey of very large-scale neighborhood search techniques. *Discrete Applied Mathematics*, **123**, 72–102. [72]

Ahuja, R.K., Magnati, T.L. and Orlin, J.B., 1993, *Network Flows: Theory, Algorithms, and Applications* (Upper Saddle River, NJ: Prentice Hall). [18, 47]

Alonso, G., Casati, F., Kuno, H. and Machiraju, V., 2004, *Web Services - Concepts, Architectures and Applications* (Heidelberg: Springer). [31]

Andrienko, G., Andrienko, P., Jankowski, P., Keim, D., Kraak, M.J., MacEachren, A. and Wrobel, S., 2007, Geovisual analytics for spatial decision support: Setting the research agenda. *International Journal of Geographical Information Science*, **21**, 839–587. [85, 86]

Anselin, L., Dodson, R. and Hudak, S., 1993, Linking GIS and Spatial Data Analysis in Practice. *Geographical Systems*, **1**, 3–23. [95]

Anthony, R.N., 1965, *Planning and Control Systems: A Framework for Analysis* (Cambridge, MA: Harvard University Graduate School of Business Administration). [9, 84]

Atkinson, J., 1994, A greedy look-ahead heuristic for combinatorial optimization: An application to vehicle scheduling with time windows. *Journal of the Operational Research Society*, **15**, 673–684. [63]

Azi, N., Gendreau, M. and Potvon, J.Y., 2007, An Exact Algorithm for a Vehicle Routing Problem with Time Windows and Multiple Use of Vehicles. *European Journal of Operational Research*, **178**, 755–766. [10]

Bailey, T.C. and Gatrell, A.C., 1995, *Interactive Spatial Data Analysis*, Harlow. [23]

Bartelme, N., 2005, *Geoinformatik - Modelle, Strukturen, Funktionen* (Berlin: Springer). [7, 9, 23, 91]

Beasley, J.E., 1996, *Advances in linear and integer programming* (Oxford: Clarendon Press). [59, 61]

Benatallah, B. and Motahari Nezhad, H.R., 2008, Service Oriented Architecture: Overview and Directions. In *Advances in Software Engineering* (Berlin: Springer), pp. "116–130". [31]

BERNERS-LEE, T., 2000, *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web* (New York, N.Y.: HarperCollins Publishers Inc.). [25]

BHARGAVA, H.K. and POWER, D.J., 2001, Decision Support Systems and Web Technologies: A Status Report. In *Proceedings of the AMCIS 2001.* [90]

BHARGAVA, H.K., POWER, D.J. and DAEWON, S., 2007, Progress in Web-based decision support technologies. *Decision Support Systems*, **43**, 1083–1095. [90]

BIANCHESINI, N. and RIGHINI, G., 2005, Heuristic algorithms for the vehicle routing problems with simultaneous pick-up and delivery. *Computers and Operations Research*, **32**, 578–594. [10]

BISHR, Y., 1998, Overcome the semantic and other barriers to GIS interoperability. *International Journal of Geographical Information Science*, **12**, 299–314. [24]

BOLLOBAS, B., 1998, *Modern Graph Theory* (Berlin: Springer Verlag). [47, 80]

BOSCH, J., 2000, *Design and Use of Software Architectures Ű Adopting and evolving a product-line approach* (Addison-Wesley Professional). [159]

BRAMEL, J. and SIMCHI-LEVI, D., 1998, *The Logic of Logistics: theory, algorithms, and applications for logistics management* (New York: Springer Verlag). [10]

BRIMICOMBE, A.J., 2002, GIS - Where are the frontiers now. In *Proceedings of the GIS 2002*, Bahrain, pp. 33–45. [xv, 37, 38]

BRITISH STANDARDS INSTITUTE, "What is a standard?", URL: `http://www.bsi-global.com/en/Standards-and-Publications/About-standards/What-is-a-standard/`, [accessed 28. March 2009]; 2009. [25]

BURROUGH, P.A. and MCDONNELL, R., 1998, *Principles of Geographical Information Systems* (New York: Oxford University Press). [7, 9, 23, 91]

CARDOSO, M.R., SALCEDO, R.L. and AZEVEDO, S.F., 1994, Non-equilibrium simulated annealing: a faster approach to combinatorial minimization. *Industrial Engineering Chemical Research*, **33**, 1908–1918. [71]

CERNY, V., 1985, A thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, **45**, 41–51. [69]

CHAKHAR, S. and MARTEL, J.M., 2003, Enhancing Geographical Information Systems Capabilities with Multi-Criteria Evaluation Functions. *Journal of Geographic Information and Decision Analysis*, **7**, 47–71. [91, 92]

CHARON, I. and HUDRY, O., 1993, The noising method: A new method for combinatorial optimization. *Operations Research Letters*, **14**, 133–137. [67]

CHARTRAND, G., 1985, *Introductory Graph Theory* (Dover: Dover Publications). [47]

CHEN, P., 1976, The Entity-Relationship Model-Toward a Unified View of Data. *ACM Transactions on Database Systems*, **1**, 9–36. [105]

CLARKE, G. and WRIGHT, J., 1964, Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, **12**, 568–581. [64]

CLOONAN, J., 1966, A heuristic approach to some sales territory problems. In *Proceedings of the fourth international conference on operations research* (MIT Press), pp. 81–84. [10]

CODD, E., 1982, Relational Database: A Practical Foundation for Productivity. *Communications of the ACM*, **25**, 109–117. [105]

COOKE, D.F., 1992, Spatial Decision Support System: not just another GIS. *Geo Info Systems*, **2**, 46–49. [91]

COPPOCK, J.T. and RHIND, D.W., 1993, The History of GIS. In *Geographical information systems: principles and applications*, D.J. Maguire, M.F. Goodchild and D.W. Rhind (Eds) (London: Longman), pp. 21–43. [91]

CORDEAU, J.F., LAPORTE, G. and ROPKE, S., 2008, Recent Models and Algorithms for One-to-One Pickup and Delivery Problems. In *Vehicle Routing: Latest Advances and Challenges*, B. Golden, S. Raghavan and E.A. Wasil (Eds) (Boston: Kluwer), pp. 327–357. [10]

CROSETTO, M. and TARANTOLA, S., 2001, Uncertainty and Sensitivity Analysis: Tools for GIS-Based Model Implementation. *International Journal of Geographical Information Science*, **15**, 415–437. [86]

CROSSLAND, M.D., WYNNE, B.E. and PERKINS, W.C., 1995, Spatial decision support systems: An overview of technology and a test of efficacy. *Decision Support Systems*, **14**, 219–235. [91]

CÔTÉ, M., SURYN, W., MARTIN, R. and LAPORTE, C., 2004, Evolving a Corporate Software Quality Assessment Exercise: A Migration Path to ISO/IEC 9126. *Software Quality Professional*, pp. 4–17. [159]

DAKIN, R.J., 1965, A tree-search algorithm for mixed integer programming problems. *The Computer Journal*, **8**, 250–255. [10, 59, 60]

DANTZIG, G.B., 1966, *Linear Programming and Extensions* (Princeton University Press). [61]

DANTZIG, G.B. and RAMSER, R.H., 1959, The Truck Dispatching Problem. *Management Science*, pp. 80–91. [10]

DEES, W.A. and KARGER, P.G., 1982, Automated rip-up and reroute techniques. In *Proceedings of the 19th conference on Design automation* (IEEE Press), pp. 432–439. [72]

DENKERS, S.D., 1992, Creating a GIS Data Schema for Information Resource Management. *URISA Journal*, **4**, 24–31. [24]

DENSHAM, P.J., 1993, Spatial decision support systems. In *Geographical information systems: principles and applications*, D.J. Maguire, M.F. Goodchild and D.W. Rhind (Eds) (London: Longman), pp. 403–412. [xvi, 83, 88]

DENZER, H., 2002, Generic Integration in Environmental Information and Decision Support Systems. In *Proceedings of the First Biennial Meeting of the International Environmental Modelling and Software Society: Integrated Assessment and Decision Support - iEMSs 2002*, A.E. Rizzoli and A.J. Jakeman (Eds), pp. 53–60. [92, 94]

DEPARTMENT OF DEFENSE, 2001, Global Positioning System standard positioning service performance standard. Technical report, U.S. Assistant for GPS, Positioning and Navigation, Defense Pentagon, Washington, D.C. [10, 40]

DEPARTMENT OF ENVIRONMENT, 1987, *Handling Geographic Information* (London: HMSO). [7]

DESAULNIERS, G., DESROSIERS, J., ERDMANN, A., SOLOMON, M.M. and SOUMIS, F., 2002, VRP with Pickup and Delivery. In *The Vehicle Routing Problem*, P. Toth and D. Vigo (Eds) (Society of Industrial and Applied Mathematics), pp. 225–242. [58]

DEWEY, J., 1910, *How do we think* (New York: D.C. Heath and Company). [84]

DI FLORA, C. and HERMERSDORF, M., 2008, A practical implementation of indoor location-based services using simple WiFi positioning. *Journal of Location Based Services*, **2**. [10, 39]

DIJKSTRA, E.W., 1959, A note on two problems in connexion with graphs. *Numerische Mathematik*, **1**, 269–271. [115]

DORRONSORO, B., "The VRP Web", URL: `http://neo.lcc.uma.es/radi-aeb/WebVRP/`, [accessed 12. October 2007]; 2007. [50]

DREW, C.H., 2003, Transparency - Considerations for PPGIS Research and Development. *URISA Journal*, **15**, 73–78. [93]

DUBLIN CORE METADATA INITIATIVE, "DCMI Glossary", URL: `http://dublincore.org/documents/usageguide/glossary.shtml`, [accessed 25. March 2009]; 2009. [24]

ECLIPSE FOUNDATION, "Jetty web server", URL: `http://www.mortbay.org/jetty`, [accessed 23. September 2009]; 2009. [136]

ERTZSCHEID, O. and GALLEZOT, G., "Formalizing the Concept of Serendipity in Web Searching", Presentation at the SearchEngine Meeting 2004, The Hague, Netherlands. Available at: URL: `http://www.infonortics.com/searchengines/sh04/slides/olivier.pdf`, [accessed 28. March 2009]; 2004. [25]

FEDRA, K., 1996, Distributed Models and Embedded GIS: Integration Strategies and Case Studies. In *GIS and Environmental Modeling: Progress and Research Issues*, M.F. Goodchild, L.T. Steyaert, B.O. Parks, C. Johnston, D. Maidment, M. Crane and S. Glendinning (Eds) (Wiley), pp. 413–417. [9, 92, 93]

FEO, T.A. and RESENDE, M.G.C., 1989, A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, **8**, 67–71. [66]

FIELDING, R.T., 2000, Architectural styles and the design of network-based software architectures. PhD thesis, University of California, Irvine, USA. [9, 32, 33]

FOGEL, L., OWENS, A. and WALSH, M., 1966, *Artificial Intelligence through Simulated Evolution* (Wiley). [62]

FOHL, P., CURTIN, K.M., GOODCHILD, M.F. and CHURCH, R.L., 1996, A non-planar, lane-based navigable data model for ITS. In *Proceedings of the Seventh International Symposium on Spatial Data Handling*, M. Kraak and M. Molenaar (Eds), Delft, pp. 7B.17–7B.29. [18]

Forrester Research, 2001, SCM schafft Wettbewerbsvorteile. In *Collaborative SCM in Branchen*, R. Scheckenbach and A. Zeier (Eds) (Galileo Press), p. 17ff. [2]

Fotheringham, A.S., Brunsdon, C. and Charlton, M., 2000, *Quantitative Geography - Perspectives on Spatial Data Analysis* (London: Sage Publications). [23]

Fourer, R., Gay, D.M. and Kernighan, B.W., 2003, *AMPL: A Modeling Language for Mathematical Programming* (Duxbury Press / Brooks/Cole Publishing Company). [103]

Fraunhofer IFF, "Intelligentes Holz – RFID in der Rundholzlogistik", URL: `http://www.intelligentes-holz.iff.fraunhofer.de`; 2009. [169]

Friedl, K., Kanzian, C. and Stampfer, K., 2004, Netzwerk Holz. Endbericht. Technical report, Institut für Forsttechnik, BOKU, Vienna. [120]

Friis-Christensen, A., Schade, S. and Peedell, S., "Approaches to solve schema heterogenity at the European level", URL: `http://www.ec-gis.org/Workshops/11ec-gis/papers/301peedell.pdf`, [accessed 27. March 2009]; 2005. [24]

Funke, B., Grünert, T. and Irnich, S., 2005, Local search for vehicle routing and scheduling problems: Review and conceptual integration. *Journal of Heuristics*, **11**, 267–306. [10, 65]

Garey, M.R. and Johnson, D.S., 1979, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (W.H. Freeman and Company). [46]

Geoframeworks LLC, "Geoframworks, GPS.Net 2.0", URL: `http://www.geoframeworks.com/`, [accessed 12. Februar 2009]; 2009. [127, 135]

Gillet, B. and Miller, L., 1974, A heuristic algorithm for the vehicle-dispatch problem. *Operations Research*, **22**, 340–349. [64]

Gisolfi, D., 2001, Web services architect. Technical report, IBM Developer Works. [xv, 32, 33]

Glover, F., 1986, Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, **13**, 533–549. [62]

Glover, F. and Laguna, M., 1997, *Tabu Search* (Berlin: Springer Verlag). [67]

Gomory, R.E., 1960, An algorithm for the mixed integer problem. Research Memorandum RM-2597-PR, RAND Corporation. [61]

Goodchild, M.F., 1992a, Geographic information science. *International Journal of Geographical Information Science*, **6**, 31–45. [15]

Goodchild, M.F., 1992b, Geographical data modelling. *Computers and Geosciences*, **18**, 401–408. [17]

Goodchild, M.F., 1998, Geographic information systems and disaggregate transportation planning. *Journal of Geographical Systems*, **5**, 19–44. [xv, 17, 18, 19, 23]

Goodchild, M.F., Egenhofer, M.J., Kemp, K.K., Mark, D.M. and Sheppard, E., 1999, Introduction to the Varenus Project. *International Journal of Geographical Information Science*, **13**, 731–745. [16]

GOODCHILD, M.F., KLINKENBERG, B. and JANELLE, D.G., 1993, A factorial model of aggregate spatio-temporal behavior: application to the diurnal cycle. *Geographical Analysis*, **25**, 277–294. [19]

GORRY, A. and SCOTT-MORTON, M., 1971, A Framework for Management Information Systems. *Sloan Management Review*, **13**, 55–70. [9, 84]

GRASS DEVELOPMENT TEAM, "GRASS GIS – Simulation Models", URL: `http://grass.itc.it/intro/modelintegration.html`, [accessed 9. December 2009]; 2009. [7]

GRONALT, M., CHLOUPEK, A., GREIGERITSCH, A. and HÄUSLMAYER, H., 2005, WoodLog - Perspektiven der Holzlogistik SupplyChain-Optimierungspotentiale durch ein Logistikleitzentrum Holz. Technical report, Bundesministerium für Verkehr, Innovation und Technologie, Wien. [5]

GRONALT, M., HARTL, R.F. and REIMANN, M., 2004, New savings based algorithms for time constrained pickup and delivery of full truckloads. *European Journal of Operational Research*, **151**, 520–535. [64]

GRUBER, T.R., 1993, A translation approach to portable ontology specification. *Knowledge Acquisition*, **5**, 199–220. [24]

GUERRA, G. and LEWIS, J., 2002, Spatial Optimization and GIS. *ArcUser*, **April-June**, 32–34. [8]

HART, P.E., NILSSON, N.J. and RAPHAEL, B., 1968, A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics SSC4*, **2**, 100–107. [115]

HARVEY, F., KUHN, W., PUNDT, H. and BISHR, Y., 1999, Semantic interoperability: A central issue for sharing geographic information. *The Annals of Regional Science*, **33**, 213–232. [24]

HERRY VERKEHRSPLANUNG, 2001, Transportpreise und Transportkosten der verschiedenen Verkehrsträger im Güterverkehr. Technical report, Kammer für Arbeiter und Angestellte für Wien. [120]

HERZIG, A., 2005, Entwicklung eines GIS-basierten Entscheidungsunterstützungssystems als Werkzeug nachhaltiger Landnutzungsplanung. PhD thesis, Christian-Albrechts-Universitaet, Kiel. [xvi, 88, 91, 92]

HERZIG, A., 2008, A GIS-based Module for the Multiobjective Optimization of Areal Resource Allocation. In *Proceedings of the 11th AGILE International Conference on Geographic Information Science 2008* (Girona, Spain: University of Girona). [8]

HÄGERSTRAND, T., 1970, What about people in regional science?. *Papers of the Regional Science Association*, **24**, 1–12. [xv, 20, 21]

HOCHBAUM, D.S. (Ed.) , 1996, *Approximation Algorithms for NP-Hard Problems* (Boston: Course Technology). [45]

HOFMANN-WELLENHOF, B., LEGAT, K. and WIESER, M., 2003, *Navigation - Principles of Positioning and Guidance* (Wien: Springer). [9, 10, 40]

HOLLEY, K., CHANNABASAVAIAH, K. and TUGGLE, E.M., J., 2003, Migrating to a Service-Oriented Architecture. Technical report, IBM Developer Works. [31]

Hornsby, K. and Egenhofer, M., 2000, Identity-based change: a foundation spatio-temporal knowledge representation. *International Journal of Geographic Information Systems*, **14**, 207–224. [19]

Howard, D., "Geographic Information Technologies and Community Planning: Spatial Empowerment and Public Participation", URL: `http://www.ncgia.ucsb.edu/varenius/ppgis/papers/howard.html`, [accessed 19. May 2009]; 1999. [93]

IEEE, 1990, *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries: 610* (New York, NY: IEEE). [24, 160, 161]

IEEE, 1998, *IEEE Standard for Conceptual Modeling Language Syntax and Semantics for IDEF1X* (New York, NY: IEEE). [105]

ISO, 2001, ISO 9126:Software engineering – Product quality – Part 1: Quality model. Technical report, International Organization for Standardization. [159, 160, 161]

ISO, 2003a, ISO 19112:Geographic information – Spatial referencing by geographic identifiers. Technical report, International Organization for Standardization. [9, 29, 41]

ISO, 2003b, ISO 19915:Geographic Information – Metadata. Technical report, International Organization for Standardization. [9, 29]

ISO, 2004, ISO 19125:Geographic Information – Simple Feature Access – Part2: SQL Option. Technical report, International Organization for Standardization. [9, 30]

ISO, 2005a, ISO 19118:Geographic Information – Encoding. Technical report, International Organization for Standardization. [29]

ISO, 2005b, ISO 19119:Geographic Information – Services. Technical report, International Organization for Standardization. [9, 30, 35]

ISO, 2005c, ISO 19128:Geographic Information – Web map server interface. Technical report, International Organization for Standardization. [9, 31]

ISO, 2005d, ISO 19133:Geographic Information – Location-based Services – Tracking and Navigation. Technical report, International Organization for Standardization. [31, 40, 41]

ISO, 2007a, ISO 19132:Geographic information – Location-based services – Reference Model. Technical report, International Organization for Standardization. [9, 31, 40, 41, 42]

ISO, 2007b, ISO 19134:Geographic information – Location-based services – Multimodal routing and navigation. Technical report, International Organization for Standardization. [31, 40]

ISO, 2007c, ISO 19136:Geographic information – Geography Markup Language (GML). Technical report, International Organization for Standardization. [29]

Jankowski, P., 1995, Integrating geographical information systems and multiple criteria decision making methods. *International Journal of Geographical Information Systems*, **9**, 251–273. [9, 92, 93]

JANKOWSKI, P. and NYERGES, T., 2003, Toward a Framework for Research on Geographic Information-Supported Participatory Decision-Making. *URISA Journal*, **15**, 9–17. [93]

JUN, C., 2000, Design of an intelligent geographic information system for multicriteria site analysis. *URISA Journal*, **12**, 5–17. [xvi, 9, 93, 95, 96]

JUNGNICKEL, D., 2005, *Graphs, Networks and Algorithms* (Berlin: Springer Verlag). [10, 47, 49, 80]

KAZMAN, R., ABOWD, G., BASS, L. and CLEMENTS, P., 1996, Scenario-Based Analysis of Software Architecture. *IEEE Software*, pp. 47–55. [160]

KAZMAN, R., KLEIN, M., BARBACCI, M., LONGSTAFF, T., LIPSON, H. and CARRIERE, J., 1998, The Architecture Tradeoff Analysis Method. In *Proceedings of the Fourth IEEE International Conference on Engineering of Complex Computer Systems (ICECCS98)* (Monterey, CA: IEEE Computer Society). [160]

KÜHMAIER, M., KANZIAN, C., HOLZLEITNER, F. and STAMPFER, K., 2007, Wertschöpfungskette Waldhackgut - Optimierung von Ernte, Transport und Logistik. Technical report, Institut für Forsttechnik, Department für Wald und Bodenwissenschaften, Universität für Bodenkultur, Wien - Projektstudie im Auftrag von BMLFUW, Land Niederösterreich, Stadt Wien und ÖBf AG. [11]

KIRKPATRICK, S., GELATT, C.D. and VECCHI, M.P., 1983, Optimization by simulated annealing. *Science*, **220**, 671–680. [xxiii, 69, 70]

KOMARKOVA, J., VISEK, O. and NOVAK, M., 2007, Heuristic Evaluation of Usability of GeoWeb Sites. In *Lecture Notes in Computer Science: Web and Wireless Geographical Information Systems, 7th International Symposium, W2GIS 2007*, J. Ware and G. Taylor (Eds) (Springer-Verlag Berlin Heidelberg), pp. 264–278. [162]

KONCZ, N.A. and ADAMS, T.M., 2002, A data model for multi-dimensional transportation applications. *International Journal of Geographical Information Science*, **16**, 551–569. [21]

KOOPERATIONSPLATTFORM FORST HOLZ PAPIER (Ed.) , 2006, *Österreichische Holzhandelsusancen 2006* (Wien: Kooperationsplattform Forst Holz Papier (FHP)). [109]

KORTEN, S. and KAUL, C., 2008, Application of RFID (Radio Frequency Identification) in the Timber Supply Chain Einsatz von RFID in der Holzerntekette. *Croatian Journal of Forest Engineering*, **29**, 551–569. [169]

KRISHNAMURTHY, P. and PAHLAVAN, K., 2004, Wireless Communications. In *Telegeoinformatics*, H.A. Karimi and A. Hammad (Eds) (CRC Press), pp. 111–142. [10, 38, 39]

KYTÖJOKI, J., NUORTIO, T., BRÄYSY, O. and GENDREAU, M., 2005, An efficient variable neighbourhood search heuristic for very large scale vehicle routing problems. *Computers and Operations Research*, **34**, 2743–2757. [10]

LAND, A.H. and DOIG, A.G., 1960, An automatic method of solving discrete programming problems. *Econometrica*, **28**, 497–520. [10, 59, 60]

LANDXML.ORG INDUSTRY CONSORTIUM, "LandXML.org", URL: `http://landxml.org/schema/LandXML-1.2/documentation/LandXML-1.2Doc.html`, [accessed 30. March 2009]; 2009. [28]

LAPORTE, G., 1992, The Vehicle Routing Problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, **59**, 345–158. [10]

LAPORTE, G. and SEMET, F., 2002, Classical heuristics for the capacitated VRP. In *The Vehicle Routing Problem*, P. Toth and D. Vigo (Eds) (Society of Industrial and Applied Mathematics), pp. 109–128. [10, 62]

LASSILA, O., 2007, Programming Semantic Web Applications: A Synthesis of Knowledge Representation and Semi-Structured Data. PhD thesis, Helsinki University of Technology. [25]

LEI, J.J. and LI, J., 2009, A Decision Support System for Supply Chain Management based on PSO and GIS. In *Proceedings of the 2009 IITA International Conference on Control, Automation and Systems Engineering* (Washington, DC., USA: IEEE Computer Society), pp. 58–51. [7]

LEITNER, M. and STAUFER-STEINOCHER, P., 2001, Kernel Density Estimation - An Innovative Exploratory Tool in Geomarketing. In *Proceedings of the Angewandte Geographische Informationsverarbeitung XIII*, J. Strobl, T. Blaschke and G. Griesebner (Eds) (Heidelberg: Herbert Wichmann Verlag), pp. 297–308. [7]

LENNTORP, B., 1976, Paths in space-time environments: A time geographic study of movement possibilities of individuals. *Lund Studies in Geography.* [21]

LEUNG, H.K.N., 2001, Quality Metrics for Intranet Applications. *Information & Management*, pp. 137–152. [159]

LINKO, F., 2008, Ermittlung von Transportkosten. *Österreichische Forstzeitung*, **4-2008**. [120]

LISTING, J.B., 1848, *Vorstudien zur Topologie* (Göttingen: Vandenhoeck und Ruprecht). [47]

LONGLEY, P., GOODCHILD, M., MAGUIRE, D. and RHIND, D. (Eds) , 1998, *Geographical Information Systems, Techniques, Management and Applications* (New York: John Wiley). [7, 9]

LOSAVIO, F., CHIRINOS, L., MATTEO, A., LÉVY, N. and RAMDANE-CHERIF, A., 2004, ISO Quality Standards for Measuring Architectures. *Journal of Systems and Software*, pp. 209–223. [159]

MACMILLAN, W.D., 1997, Computing and the science of geography: the postmodern turn and the geocomputational twist. In *Proceedings of the 2nd International Conference on GeoComputation*, R.T. Pascoe (Ed.), University of Otago, New Zealand, pp. 1–11. [7]

MAKOWSKI, M. and WIERZBICKI, A.P., 2000, Architecture of Decision Support Systems. In *Model-Based Decision Support Methodology with Environmental Applications*, A.P. Wierzbicki, M. Makowski and J. Wessels (Eds) (Springer), pp. 47–70. [xvi, 84, 88]

MALCZEWSKI, J., *NCGIA Core Curriculum in GIS. Unit 127-Spatial Decision Support Systems* 1997. [xvi, 83, 89, 92]

MALCZEWSKI, J., 1999, *GIS and Multicriteria Decision Analysis* (New York: John Wiley and Sons). [xvi, 9, 83, 85, 86, 87, 88, 89, 92]

MALCZEWSKI, J., 2006, GIS-based multicriteria decision analysis: a survey of the literature. *International Journal of Geographical Information Science*, **20**, 703–726. [9, 93, 94]

MARK, D.M., 2000, Geographic Information Science: Critical issues in an emerging cross-disciplinary research domain. *Journal of the Urban and Regional Information Systems Association*, **12**, 45–54. [15]

MARTELLO, S. and TOTH, P., 1990, *Knapsack Problems: Algorithms and Computer Implementations* (Chichester, UK: Wiley). [52]

MATTSSON, M., GRAHN, H. and MÅRTENSSON, F., 2006, Software Architecture Evaluation Methods for Performance, Maintainability, Testability, and Portability. In *Proceedings of the Second International Conference on the Quality of Software Architectures (QoSA 2006), June 27-29, 2006*, Västerås, Sweden. [158]

MAXIMAL SOFTWARE INC., "About Maximal Software", URL: `http://www.maximal-usa.com/maximal/`, [accessed 20. August 2009]; 2009. [103]

METROPOLIS, N., ROSENBLUTH, A., ROSENBLUTH, M., TELLER, A. and TELLER, E., 1953, Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, **21**, 1087–1092. [69]

MITTENTHAL, J. and NOON, C.E., 1992, An insert delete heuristic for the traveling salesman subset-tour problem with one additional constraint. *The Journal of the Operational Research Society*, **43**, 277–283. [10]

MLACENOVIC, N. and HANSEN, P., 1997, Variable neighborhood search. *Computers and Operations Research*, **24**, 1097–1100. [xxiii, 67, 68]

MOBILKOM AUSTRIA, "A1 Netzabdeckung", URL: `http://www.a1.net/business/coveragemap/private.php?utm_source=A1net&utm_medium=startseite&utm_content=breiband`, [accessed 24. February 2009]; 2009. [39]

NADDEF, D. and RINALDI, G., 2002, Branch-and-Cut Algorithms for the Capacitated VRP. In *The Vehicle Routing Problem*, P. Toth and D. Vigo (Eds) (Society of Industrial and Applied Mathematics), pp. 29–51. [51]

NAUR, P. and RANDELL, B. (Eds) , 1968, *SOFTWARE ENGINEERING: Report of a conference sponsored by the NATO Science Committee* (Scientific Affairs Division, NATO). [164]

NAVLOG GMBH, "NavLog - die vernünftige Navigation im Wald", URL: `http://www.navlog.de`, [accessed on 23. August 2009]; 2009. [137]

NYERGES, T.L., 1992, Coupling GIS and Spatial Analytical Models. In *Proceedings of the 5th International Symposium on Spatial Data Handling*, P. Breshanan, E. Corwin and D. Cowen (Eds), Charleston, SC. Humanities and Social Sciences Computing Laboratory, University of South Carolina, pp. 534–543. [9, 93]

OGC, "OpenGIS® Discussion Paper: Web Notification Service", URL: http://portal.opengeospatial.org/files/?artifact_id=1367; 2003. [167]

OGC, "OpenGIS® Web Feature Service Implementation Specification 1.1", URL: http://www.opengeospatial.org/standards/wfs; 2005. [9, 27]

OGC, "OpenGIS® Implementation Specification for Geographic information - Simple feature access - Part 2: SQL option 1.2.0", URL: http://www.opengeospatial.org/standards/sfs; 2006a. [29, 136]

OGC, "OpenGIS® Sensor Alert Service Candidate Implementation Specification (Best Practice Document)", URL: http://portal.opengeospatial.org/files/?artifact_id=15588; 2006b. [167]

OGC, "OpenGIS® Web Map Service Implementation Specification 1.3.0", URL: http://www.opengeospatial.org/standards/wms; 2006c. [9, 26, 41]

OGC, "OpenGIS® Geography Markup Language (GML) Encoding Standard 3.2.1", URL: http://www.opengeospatial.org/standards/gml; 2007a. [28]

OGC, "OpenGIS® Observations and Measurements - Part 1 - Observation schema", URL: http://www.opengeospatial.org/standards/om; 2007b. [167]

OGC, "OpenGIS® Sensor Model Language (SensorML) Implementation Specification", URL: http://www.opengeospatial.org/standards/sensorml; 2007c. [28]

OGC, "OpenGIS® Sensor Model Language (SensorML) Implementation Specification", URL: http://www.opengeospatial.org/standards/sensorml; 2007d. [167]

OGC, "OpenGIS® Sensor Observation Service", URL: http://www.opengeospatial.org/standards/sos; 2007e. [167]

OGC, "OpenGIS® Sensor Planning Service Implementation Specification", URL: http://www.opengeospatial.org/standards/sps; 2007f. [167]

OGC, "OpenGIS® Transducer Markup Language (TML) Implementation Specification", URL: http://www.opengeospatial.org/standards/tml; 2007g. [167]

OGC, "OpenGIS® Web Processing Service", URL: http://www.opengeospatial.org/standards/wps; 2007h. [173]

OGC, "OpenGIS® City Geography Markup Language (CityGML) Encoding Standard", URL: http://www.opengeospatial.org/standards/citygml; 2008a. [28]

OGC, "OpenGIS® Location Services (OpenLS): Core Services", URL: http://www.opengeospatial.org/standards/ols; 2008b. [9, 16, 27, 36, 41]

OGC, "OpenGIS® Location Services (OpenLS): part 6-Navigation Service", URL: http://www.opengeospatial.org/standards/ols; 2008c. [9, 27]

OGC, "OpenGIS® Location Services: Tracking Service Interface Standard", URL: http://www.opengeospatial.org/standards/ols; 2008d. [9, 27]

OGC, "OpenGIS® Web Services, Phase 4", URL: http://www.opengeospatial.org/projects/initiatives/ows-4; 2009a. [9, 35, 90]

OGC, "OpenGIS® Web Services, Phase 5", URL: http://www.opengeospatial.org/projects/initiatives/ows-5; 2009b. [35, 90]

OGC, "OpenGIS® Web Services, Phase 6", URL: http://www.opengeospatial.org/projects/initiatives/ows-6; 2009c. [90]

OPENGEO, "Geoserver Documentation", URL: http://docs.geoserver.org/, [accessed 6. March 2009]; 2009. [112, 136]

OPENSHAW, S. and ABRAHART, R.J., 1996, Geocomputation. In *Proceedings of the 1st International Conference on GeoComputation*, R. Abrahart (Ed.), University of Leeds, UK, pp. 665–666. [7]

ORGANIZATION FOR THE ADVANCEMENT OF STRUCTURED INFORMATION STANDARDS, 2006, "Reference Model for Service Oriented Architecture 1.0". Technical report, OASIS. [9, 31]

ORKNEY, INC., "The pgRouting Project", URL: http://pgrouting.postlbs.org/; 2009. [115, 136]

ORTEGA, M., PÉREZ, M. and ROJAS, T., 2003, Construction of a Systemic Quality Model for Evaluating a Software Product. *Software Quality Journal*, pp. 219–242. [159, 162]

OSTLÄNDER, N., 2004, Interoperable Services for Web-Based Spatial Decision Support. In *Proceedings of the 7th AGILE conference on Geographic Information Science*, Heraklion, Greece, pp. 807–811. [35, 90, 93]

OVUM, "Mobile Location Services: Market Strategies", 2000. [37]

PADBERG, M. and RINALDI, G., 1991, A branch-and-cut algorithm for the solution of large-scale traveling salesmen problems. *SIAM Review*, **33**, 1–41. [61]

PAPAZOGLOU, M. and VAN DEN HEUVEL, W.J., 2007, Service oriented architectures: approaches, technologies and research issues. *International Journal on very large data bases (VLDB)*, **16**, 389–415. [31]

PAS, E.I., 1985, State of the art and research opportunities in travel demand: another perspective. *Transportation Research*, **19**, 460–464. [22]

PEARL, J., 1984, *Heuristics: Intelligent Search Strategies for Computer Problem Solving* (Addison-Wesley Publishing). [62]

PENG, Z.R. and TSOU, M., 2003, *Internet GIS: Distributed Geographic Information Services for the Internet and Wireless Networks* (Hoboken, New Jersey: John Wiley and Sons). [35]

PIFF, M., 1991, *Discrete Mathematics: An Introduction for Software Engineers* (Cambridge: Cambridge University Press). [47, 49]

PISINGER, D. and ROPKE, S., 2005, A general heuristic for vehicle routing problems. *Computers and Operations Research*, **34**, 2403–2435. [xvi, xxiii, 10, 45, 72, 73, 74, 75, 76, 118, 123]

POSLAD, S., 2009, *Ubiquitous Computing Smart Devices, Environments and Interactions* (Wiley & Sons). [166, 167]

RECHENBERG, I., 1965, *Cybernetic Solution Path of an Experimental Problem* (Royal Aircraft Establishment Library Translation). [62]

REICHENBACHER, T., 2004, Mobile Cartography - Adaptive Visualisation of Geographic Information on Mobile Devices. PhD thesis, TU Munich. [3, 9, 37, 41, 42]

RENOLEN, A., 2000, Modelling the real world: conceptual modelling in spatiotemporal information system design. *Transactions in GIS*, **4**, 23–42. [xv, 19, 20, 21]

RINNER, C., 2003, Web-based Spatial Decision Support: Status and Research Directions. *Journal of Geographic Information and Decision Analysis*, **7**, 14–31. [xvi, 35, 92, 93]

ROPKE, S., 2005, Heuristic and exact algorithms for vehicle routing problems. PhD thesis, University of Copenhagen. [xvi, xxiii, 10, 45, 51, 56, 57, 58, 62, 64, 66, 77, 156]

ROPKE, S. and CORDEAU, J.F., iRev, Branch-and-Cut-and-Price for the Pickup and Delivery Problem with Time Windows. *Transportation Science*. [10, 46, 62]

ROPKE, S., CORDEAU, J.F. and LAPORTE, G., 2007, Models and a Branch-and-Cut Algorithm for Pickup and Delivery Problems with Time Windows. *Networks*, **9**, 258–272. [10, 46, 62]

ROPKE, S. and PISINGER, D., 2006, An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transportation Science*, **40**, 455–472. [xxiii, 72, 74, 75, 78, 118, 124, 126, 156]

SAVELSBERGH, M., 1997, A branch-and-price algorithm for the generalized assignment problem. *Operations Research*, **45**, 831–841. [59]

SCHILLER, J.H., 2003, *Mobile Communications* (London: Addison Wesley). [10, 38, 39]

SCHRIMPF, G., SCHNEIDER, J., STAMM-WILBRANDT, H. and DUECK, G., 2000, Record braking optimization results using the ruin and recreate principle. *Journal of Computational Physics*, **159**, 139–171. [72]

SHAW, P., 1998, Using constraint programming and local search methods to solve vehicle routing problems. In *Proceedings of the Fourth International Conference on Principles and Practice of Constraint Programming (Volume 1520 of Lecture Notes in Computer Science)*, pp. 417–431. [46, 72, 74]

SHIM, J.P., WARKENTIN, M., COUTNEY, J.F., POWER, D.J., SHARDA, R. and CARLSSON, C., 2002, Past, present, and future of decision support technology. *Decision Support Systems*, **33**, 111–126. [83, 84]

SIMON, H.A., 1960, *The New Science of Management Decision* (New York: Harper Brothers). [9, 84]

SPITTER, J.M., 2005, Rolling Schedule Approaches for Supply Chain Operations Planning. PhD thesis, Technische Universiteit Eindhoven. [xv, 2, 10, 79]

SPRAGUE, R.H., 1980, A framework for the development of decision support systems. *Management Information Sciences Quarterly*, **4**, 1–26. [xvi, 89, 90, 91, 92]

STUCKENSCHMIDT, H., 2003, Ontology-Based Information Sharing in Weakly Structured Environments. PhD thesis, Amsterdam: Vrije Universiteit. [24]

TALBI, E.G., 2009, *Metaheuristics: from design to implementation* (Hoboken, NJ: John Wiley & Sons). [xvi, xxiii, 10, 45, 46, 47, 62, 63, 66, 67, 68, 69]

TAYLOR, R.N., MEDVIDOVIC, N. and DASHOFY, E., 2009, *Software Architecture: Foundations, Theory, and Practice* (Hoboken, NJ: John Wiley & Sons). [162]

TENG, S.Y., ONG, H.L. and HUANG, H.C., 2006, Heuristic algorithms for visiting the customers in a rolling schedule environment. *OR Spectrum*, **28**, 241–266. [10, 79]

THE R FOUNDATION FOR STATISTICAL COMPUTING, "The R Project for Statistical Computing", URL: `http://www.r-project.org/`, [accessed 21. September 2009]; 2009. [89]

THILL, J.C., 2000, Geographic information systems for transportation in perspective. *Transportation Research Part C*, **8**, 3–12. [17]

THOMAS, J.J. and COOK, K.A. (Eds) , 2005, *Illuminating the Path. The Research and development Agenda for Visual Analytics* (Los Alamitos, CA: IEEE Computer Society). [85]

TOMLIN, C.D., 1990, *Geographic Information Systems and Cartographic Modeling* (Englewood Cliffs, NJ: Prentice-Hall). [93]

TOTH, P. and VIGO, D., 2002a, Branch-and-Bound Algorithms for the Capacitated VRP. In *The Vehicle Routing Problem*, P. Toth and D. Vigo (Eds) (Society of Industrial and Applied Mathematics), pp. 29–51. [51]

TOTH, P. and VIGO, D., 2002b, An overview of Vehicle Routing Problems. In *The Vehicle Routing Problem*, P. Toth and D. Vigo (Eds) (Society of Industrial and Applied Mathematics), pp. 1–26. [xvi, 10, 50, 51, 52, 55, 56]

TOTH, P. and VIGO, D. (Eds) , 2002c, *The Vehicle Routing Problem* (Society of Industrial and Applied Mathematics). [50]

TRYFONIA, N. and JENSEN, C., 1999, Conceptual data modeling for spatiotemporal applications. *Geoinformatica*, **3**, 245–268. [19]

TSOU, M.H., 2001, A dynamic architecture for distributing geographic information services on the internet. PhD thesis, University of Colorado, Boulder. [35]

UMAR, A., 2004, *Mobile Computing and Wireless Communications: Applications, Networks, Platforms, Architectures and Security* (nge Solutions). [10, 38, 39]

UNGERER, M.J. and GOODCHILD, M.F., 2002, Integrating Spatial Data Analysis and GIS: A New Implementation Using the Component Object Model (COM). *International Journal of Geographical Information Science*, **16**, 41–53. [9, 93, 94]

UNIVERSITY CONSORTIUM FOR GEOGRAPHIC INFORMATION SCIENCE, 1996, Research priorities for geographic information science. *Cartography and Geographic Information Systems*, **23**, 115–127. [16, 17]

UNIVERSITY OF MINNESOTA, "Welcome to Mapserver – Mapserver 5.4.2 Documentation", URL: `www.mapserver.org`, [accessed 24. February 2009]; 2009. [112, 136]

VAN DIJK, S., THIERENS, D. and DE BERG, M., 2002, Using genetic algorithms for solving hard problems ins GIS. *GeoInformatica*, **6**, 381–413. [7]

VAN LAARHOVEN, P.J.M. and AARTS, E.H.L., 1987, *Simulated Annealing: Theory and Applications (Mathematics and Its Applications)* (Dordrecht, Holland: D. Reidel Publishing Company). [70]

VAZIRANI, V.V., 2001, *Approximation Algorithms* (Berlin: Springer-Verlag). [46]

VIRRANTAUS, K., MARKKULA, J., GARMASH, A. and TERZIYAN, Y.V., 2001, Developing GIS-Supported Location Based Services. In *Proceedings of the WGIS 2001 - First International Workshop on Web Geographical Information Systems* (Kyoto, Japan: MIT Press), pp. 423–432. [36, 41]

VOGEL, O., ARNOLD, I., CHUGTAI, A., IHLER, E., KEHRER, T., MEHLIG, U. and ZDUN, U., 2009, *Software-Architektur: Grundlagen - Konzepte - Praxis* (Spektrum Akademischer Verlag). [162]

VON BODELSCHWINGH, E., BAUER, J. and LONGO, M., 2003, Informationsflüsse in der modernen Holzerntekette. Praxiseinsatz der Logistiksoftware GeoMail. *AFZ-DerWald*, **17**, 2–4. [xv, 6, 11]

VOSS, S., 2001, *Metaheuristics: The state of the art. Lecture Nodes in Computer Science, 2148.* [46]

VOSS, S., BORN, J. and SANTIBANEZ-KOREF, I., 2005, Looking ahead with the pilot method. *Annals of Operations Research*, **285-302**, 225–242. [63]

VOUDRIS, C. and TSANG, E., 1999, Guided local learch. *European Journal of Operational Research*, **113**, 469–499. [67]

W3C, "OWL Web Ontology Language", URL: `http://www.w3.org/TR/owl-ref/`, [accessed 27. March 2009]; 2009a. [24, 35]

W3C, "Simple Object Access Protocol (SOAP) 1.2", URL: `http://www.w3.org/TR/soap/`, [accessed 07. April 2009]; 2009b. [9, 32, 34]

WAGNER, H.M. and WHITHIN, T.M., 1958, Dynamic Version of the Economic Lot Size Model. *Management Science*, **5**, 89–96. [10, 79]

WANG, D. and CHENG, T., 2001, A spatio-temporal data model for activity-based transport demand modelling. *International Journal of Geographical Information Science*, **15**, 561–585. [xv, 20, 21]

WANG, L. and CHENG, Q., 2006, Web-based collaborative decision support services: concept, challenges and application. *International Archives of Photogrammetry, Remote Sensing, and Spatial Information Sciences Vol. XXXVI - Part 2*, pp. 79–84. [34, 35]

WANNENWETSCH, H., 2005, *Vernetztes Supply Chain Management* (Berlin: Springer). [xv, 1, 3, 4]

WARD, D.P., MURRAY, A.T. and PHINN, S.R., 2003, Integrating spatial optimization and cellular automata for evaluating urban change. *Annals of Regional Science*, **37**, 131–148. [8]

WEIMANN, F.X., 2008, Development of a Pedestrian Navigation System. PhD thesis, Graz University of Technology. [10, 40]

WEISE, T., 2009, *Global Optimization Algorithms: Theory and Application*, URL: `http://www.it-weise.de/projects/book.pdf`;. [xvi, 48, 62]

WEISER, M., 1991, The computer for the 21st century. *Scientific American*, **265**, 94–104. [166]

WESELS, J. and WIERZBICKI, A.P., 2000, Model-Based Decision Support. In *Model-Based Decision Support Methodology with Environmental Applications*, A.P. Wierzbicki, M. Makowski and J. Wessels (Eds) (Springer), pp. 9–28. [83]

WOLSEY, L.A., 1998, *Integer Programming* (New York: Wiley). [59, 61]

WORBOYS, M.F., 1995, *GIS – a Computing Perspective* (London: Taylor Francis Ltd.). [7, 23]

YU, H., 2006, Spatio-temporal GIS Design for Exploring Interactions of Human Activities. *Cartography and Geographic Information Science*, **33**, 3–19. [xv, 21, 22]

YUE, P., DI, L., YANG, W., YU, G., ZHAO, P. and GONG, J., "Semantic Web Services-based process planning for earth science applications", International Journal of Geographical Information Science, iFirst Article, URL: `http://dx.doi.org/10.1080/13658810802032680`; 2008. [24, 35]

ZALESKY, M.J., "Lecture script: Systems Architecture – Monolithic systems", URL: `http://www.cs.toronto.edu/~matz/instruct/csc407/lectures/10c_monolithic-4.pdf`, [accessed 30. October 2009]; 2003. [xxi, 163]

ZIMMERMANN, H.J., 2005, *Operations Research* (Wiesbaden: Vieweg Verlag). [xvi, 10, 61]

ZIPF, A., 2004, Mobile Anwendungen auf Basis von Geodateninfrastrukturen - von LBS zu UbiGIS. In *Geodateninfrastruktur, Grundlagen und Anwendungen*, L. Bernard, J. Fitzke and R. Wagner (Eds) (Heidelberg: Herbert Wichmann Verlag). [9]

Part III

Appendix

Test Instance 1

Timber supply							
P_{ID}	**X**	**Y**	**start**	**due**	**qual**	**quant**	**start**
1	494943	5218044	01/09/2009	15/10/2009	13	200	01/09/2009
					14	300	01/09/2009
					15	150	01/09/2009
					93	30	15/09/2009
2	504643	5217943	01/09/2009	15/10/2009	14	550	01/09/2009
					15	150	01/09/2009
3	519799	5220065	30/09/2009	15/11/2009	92	200	10/10/2009
					93	50	30/09/2009
4	502117	5205111	01/10/2009	15/10/2009	13	150	01/10/2009
5	502622	5254115	01/10/2009	03/10/2009	14	30	01/10/2009
6	494741	5243506	17/10/2009	23/10/2009	14	100	23/10/2009
7	505451	5239565	01/11/2009	08/11/2009	93	300	11/01/2009
					92	120	01/11/2009
					91	30	01/11/2009
8	523032	5232392	15/09/2009	15/10/2009	123	250	15/09/2009
					122	100	15/09/2009
					121	60	15/09/2009
					93	70	15/09/2009
9	527376	5244415	15/09/2009	30/09/2009	14	35	15/09/2009
					13	13	15/09/2009
					15	15	15/09/2009
					93	10	15/09/2009
10	536369	5241081	01/10/2009	10/10/2009	91	30	01/10/2009
					92	49	01/10/2009
					93	80	01/10/2009
					123	20	01/10/2009
					15	25	01/10/2009

Table 3: Test Instance 1: timber supply.

Timber demand						
saw mill ID	**task**	**start**	**due**	**qual**	**quant**	**price**
1	1	01/09/2009	30/09/2009	13	10	75
				14	35	55
				15	15	35
				93	10	35
1	2	01/10/2009	15/10/2009	13	100	70
				14	450	45
				15	150	30
				91	10	100
				92	20	75
				93	50	45
				121	20	120
				122	30	90
				123	100	55
2	6	15/10/2009	20/11/2009	92	245	80
				13	100	72
				14	110	52
				15	55	39
				93	230	106
3	4	01/09/2009	20/11/2009	14	100	59
				15	20	36
				93	90	41
				92	60	72
				13	70	73
4	5	15/09/2009	20/11/2009	14	190	58
				15	75	42
				91	30	105
				92	30	80
				93	45	49
				13	30	74
4	3	01/09/2009	15/10/2009	13	50	60
				14	130	45
				15	25	25
				91	20	90
				93	100	40
				92	29	70
				121	40	110
				122	70	69
				123	170	40

Table 4: Test Instance 1: timber demand.

saw mill ID	X	Y
1	506484	5244553
2	544079	5267324
3	527851	5257730
4	532326	5219031

Table 5: Test instance 1: saw mills.

Vehicles and haulage companies							
haul_comp	**X**	**Y**	**Truck ID**	**cap**	**start**	**end**	**w_days**
1	536114	5221004	6	20	07:30	17:00	12345
			11	19	07:30	17:00	12345
			12	14	07:30	17:00	12345
			15	22	07:30	17:00	12345
			27	16	07:30	17:00	12345
			28	13	07:30	17:00	12345
			33	19	07:30	17:00	12345
2	514759	5250904	2	14	07:30	17:00	12345
			18	21	07:30	17:00	12345
			19	24	07:30	17:00	12345
			25	22	07:30	17:00	12345
			31	25	07:30	17:00	12345
3	530361	5229488	3	14	07:30	17:00	12345
			7	20	07:30	17:00	12345
			8	13	07:30	17:00	12345
			17	18	07:30	17:00	12345
			22	23	07:30	17:00	12345
			23	23	07:30	17:00	12345
			26	23	07:30	17:00	12345
			35	20	07:30	17:00	12345
4	505627	5212063	5	13	07:30	17:00	12345
			9	20	07:30	17:00	12345
			21	16	07:30	17:00	12345
			24	23	07:30	17:00	12345
			32	22	07:30	17:00	12345
			34	14	07:30	17:00	12345
5	492977	5248649	4	21	07:30	17:00	12345
			10	23	07:30	17:00	12345
			29	19	07:30	17:00	12345
6	547326	5268859	1	15	07:30	17:00	12345
			14	20	07:30	17:00	12345
			20	15	07:30	17:00	12345
			30	16	07:30	17:00	12345

Table 6: Test Instance 1: vehicles and haulage companies.

Test Instance 2

Timber supply							
P_{ID}	**X**	**Y**	**start**	**due**	**qual**	**quant**	**start**
1	495639	5254318	01/09/2009	30/10/2009	13	1000	01/09/2009
					14	2500	01/09/2009
					15	1600	01/09/2009
2	504643	5217943	01/09/2009	15/10/2009	14	750	01/09/2009
					15	470	01/09/2009
					13	520	01/09/2009
3	519799	5220065	30/09/2009	15/11/2009	92	380	30/09/2009
					93	570	30/09/2009
					13	600	30/09/2009
					14	890	30/09/2009
					15	1400	30/09/2009
4	495767	5220891	01/10/2009	08/11/2009	13	600	01/10/2009
					14	724	01/10/2009
					15	599	01/10/2009
					121	365	01/10/2009
					122	637	01/10/2009
					123	465	01/10/2009
					93	380	01/10/2009
5	521760	5261169	15/09/2009	20/10/2009	13	690	15/09/2009
					14	1200	15/09/2009
					15	810	15/09/2009
					92	470	15/09/2009
					93	630	15/09/2009
					121	1400	15/09/2009
					122	1800	15/09/2009
6	508485	5256030	23/09/2009	10/10/2009	91	680	23/10/2009
					92	945	23/10/2009
					93	1200	23/10/2009
					123	260	23/10/2009
7	496289	5241758	10/10/2009	20/11/2009	13	700	10/10/2009
					14	654	10/10/2009
					15	890	10/10/2009
					93	230	10/10/2009
					123	192	10/10/2009
to be continued on next page ...							

Timber supply							
P_{ID}	**X**	**Y**	**start**	**due**	**qual**	**quant**	**start**
8	493941	5232368	23/09/2009	10/10/2009	13	400	23/09/2009
					14	670	23/09/2009
					15	920	23/09/2009
					122	480	23/09/2009
					123	490	23/09/2009
9	532544	5247496	15/09/2009	15/10/2009	121	780	15/09/2009
					122	930	15/09/2009
					123	745	15/09/2009
10	536369	5241081	15/09/2009	30/09/2009	91	390	15/09/2009
					92	530	15/09/2009
					93	420	15/09/2009
					123	230	15/09/2009
11	531766	5266104	15/09/2009	10/11/2009	121	400	10/10/2009
					122	700	10/10/2009
					123	380	10/10/2009
					92	260	15/09/2009
					93	360	15/09/2009
12	543758	5248917	20/09/2009	30/10/2009	13	690	20/09/2009
					14	582	20/09/2009
					15	792	20/09/2009
13	512181	5232329	01/09/2009	30/10/2009	13	460	01/09/2009
					14	530	01/09/2009
					15	390	01/09/2009
14	539761	5226334	14/09/2009	15/10/2009	91	520	14/09/2009
					92	370	14/09/2009
					93	690	14/09/2009
					121	270	14/09/2009
					122	380	14/09/2009
					123	590	14/09/2009
					15	150	14/09/2009
15	544157	5211145	15/09/2009	30/09/2009	13	580	15/09/2009
					14	320	15/09/2009
16	518776	5240923	15/09/2009	20/10/2009	91	790	15/09/2009
					92	480	15/09/2009
17	503187	5226334	01/10/2009	08/11/2009	122	300	01/10/2009
18	524572	5214142	01/09/2009	30/10/2009	121	430	01/09/2009
					122	550	01/09/2009
					123	380	01/09/2009
					93	270	01/09/2009
19	541959	5234128	30/09/2009	15/11/2009	13	210	30/09/2009
					14	400	30/09/2009
					15	600	30/09/2009
					123	380	30/09/2009
20	541359	5259309	15/09/2009	20/10/2009	14	550	15/09/2009
					15	150	15/09/2009
					15	150	15/09/2009

Table 7: Test Instance 2: timber supply.

Timber demand						
saw mill ID	**task**	**start**	**due**	**qual**	**quant**	**price**
1	1	01/09/2009	30/09/2009	13	580	87
				14	200	60
				91	390	95
				92	530	60
				93	420	45
1	2	15/09/2009	15/10/2009	15	980	39
				13	400	85
				14	570	56
				92	720	67
2	6	15/09/2009	20/11/2009	122	2700	74
				13	2500	72
				14	3950	43
				5	4900	32
2	8	15/09/2009	20/11/2009	93	1890	43
				121	2700	119
				122	1500	80
				123	1600	46
				92	1000	62
3	7	01/10/2009	20/11/2009	13	2450	75
				14	3530	51
				15	1500	37
3	4	15/09/2009	15/10/2009	14	970	50
				13	520	83
				15	610	35
				91	700	79
				92	670	58
				93	930	58
				123	2085	36
4	5	01/09/2009	15/10/2009	122	970	71
				121	380	114
4	3	15/09/2009	15/10/2009	121	670	113
				91	500	72
				92	390	52
				93	960	39
				122	850	106
4	10	01/10/2009	11/20/2009	122	177	86
				15	631	41
				91	790	101
				92	125	67
				93	560	46
				121	325	125
				123	407	60
4	9	15/09/2009	30/09/2009	123	230	52

Table 8: Test Instance 2: timber demand.

saw mill ID	X	Y
1	506484	5244553
2	544079	5267324
3	527851	5257730
4	532326	5219031

Table 9: Test instance 2: saw mills.

Vehicles and haulage companies							
haul_comp	**X**	**Y**	**Truck ID**	**cap**	**start**	**end**	**w_days**
1	536114	5221004	6	20	07:30	17:00	12345
			11	19	07:30	17:00	12345
			12	14	07:30	17:00	12345
			15	22	07:30	17:00	12345
			27	16	07:30	17:00	12345
			28	13	07:30	17:00	12345
			33	19	07:30	17:00	12345
2	514759	5250904	2	14	07:30	17:00	12345
			18	21	07:30	17:00	12345
			19	24	07:30	17:00	12345
			25	22	07:30	17:00	12345
			31	25	07:30	17:00	12345
3	530361	5229488	3	14	07:30	17:00	12345
			7	20	07:30	17:00	12345
			8	13	07:30	17:00	12345
			17	18	07:30	17:00	12345
			22	23	07:30	17:00	12345
			23	23	07:30	17:00	12345
			26	23	07:30	17:00	12345
			35	20	07:30	17:00	12345
4	505627	5212063	5	13	07:30	17:00	12345
			9	20	07:30	17:00	12345
			21	16	07:30	17:00	12345
			24	23	07:30	17:00	12345
			32	22	07:30	17:00	12345
			34	14	07:30	17:00	12345
5	492977	5248649	4	21	07:30	17:00	12345
			10	23	07:30	17:00	12345
			29	19	07:30	17:00	12345
6	547326	5268859	1	15	07:30	17:00	12345
			14	20	07:30	17:00	12345
			20	15	07:30	17:00	12345
			30	16	07:30	17:00	12345

Table 10: Test Instance 2: vehicles and haulage companies.