Produktion und Logistik

Steffen Schorpp

Dynamic Fleet Management for International Truck Transportation

Focusing on Occasional Transportation Tasks

GABLER

RESEARCH

Steffen Schorpp

Dynamic Fleet Management for International Truck Transportation

GABLER RESEARCH

Produktion und Logistik

Kontakt: Professor Dr. Hans-Otto Günther, Technische Universität Berlin,
H 95, Straße des 17. Juni 135, 10623 Berlin

Diese Reihe dient der Veröffentlichung neuer Forschungsergebnisse auf den Gebieten der Produktion und Logistik. Aufgenommen werden vor allem herausragende quantitativ orientierte Dissertationen und Habilitationsschriften. Die Publikationen vermitteln innovative Beiträge zur Lösung praktischer Anwendungsprobleme der Produktion und Logistik unter Einsatz quantitativer Methoden und moderner Informationstechnologie.

Steffen Schorpp

Dynamic Fleet Management for International Truck Transportation

Focusing on Occasional Transportation Tasks

With a foreword by Prof. Dr. Bernhard Fleischmann

GABLER

RESEARCH

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Dissertation University of Augsburg, 2010

1st Edition 2011

Editorial Office: Stefanie Brich | Nicole Schweitzer

Gabler Verlag ist eine Marke von Springer Fachmedien.
Springer Fachmedien ist Teil der Fachverlagsgruppe Springer Science+Business Media.
www.gabler.de

Cover design: KünkelLopka Medienentwicklung, Heidelberg
Printed on acid-free paper
Printed in Germany

ISBN 978-3-8349-2879-5

Foreword

Vehicle routing has been the subject of intensive research for 50 years, both in Operations Research (OR), where the development of many kinds of algorithms has been advanced, and in Logistics, where various fields of application have been investigated. A quite new direction of transport planning has occurred due to recent developments in information and communication technology, “Telematics”, which enable a central planning department to control a large vehicle fleet in real time. The necessary technical equipment is standard in any truck nowadays. Thus, OR-oriented research has increasingly turned towards dynamic vehicle routing since about 2000. Most studies in this field attempt to “dynamize” known algorithmic concepts and to investigate their appropriateness for dynamic routing. The most frequently used test data for this purpose are dynamic versions of the classical Solomon data (1987) for the vehicle routing problem, which unfortunately has little importance as a dynamic problem, except maybe for collecting goods. Much more important in this context is the Pickup and Delivery Problem (PDP) with depot free routing. Most of the real applications so far concern a PDP in a local urban area.

The present thesis starts from earlier work in this field. The author extends the approach of Fleischmann and Sandvoß (2004) for the Single Load PDP with time windows, based on the optimal assignment of orders to vehicles, for the Multi Load case with capacity constraints. Moreover, he develops a new Local Search algorithm, based on Multiple Neighborhood Search (MNS), and compares various algorithms in a comprehensive computational test.

However, the cooperation with a large German carrier led to an entirely new field of application: the dynamic control of a huge fleet of more than 1000 trucks which perform occasional transportation orders, mostly full truck loads, across the whole of Europe. This network free transportation concept, also known as “Tramp Transportation”, is of increasing importance. Particular requisites of this case are the consideration of the EC regulations on driving and working hours, and different types of vehicles and orders with restricted compatibility.

The author modifies the MNS algorithm for this problem and uses it in a large case study with real data from a five-week period with 950 vehicles and 14,000 orders. He simulates the use of the MNS algorithm and compares the results with the actual routes as a benchmark. Objectives are the empty driven kilometers and the delays against the time windows, which can be influenced by different settings of the penalty costs, resulting in a trade-off curve. He shows that computation time for the local search is critical: the best results are obtained if the simulation clock advances in real time. The author succeeds in creating solutions with a trade-off curve significantly below the benchmark. The results are validated in detail by the experts of the carrier.

This work impresses with innovative algorithms, carefully designed computational tests and a thorough analysis of the results. Its main achievement is the solution of a practical case of dynamic routing with an extremely complex planning situation, which had not been investigated prior to now. I hope that this outstanding contribution to the field of transport planning attracts widespread attention.

Augsburg, January 2011 Bernhard Fleischmann

Acknowledgements

Writing this thesis at the University of Augsburg has been a great, but also very challenging, experience for me. During this process, I received a lot of support from several people. I thank all of them. The ones below in particular.

First, I would like to express my sincere gratitude to Professor Dr. Bernhard Fleischmann, who was my supervisor over the last four years. I want to thank him for allowing me to research and write this dissertation at the Chair of Production and Logistics, for the professional advice he offered in each phase of this work, and also for his continual confidence in my efforts. Moreover, I like to thank Professor Dr. Robert Klein for reviewing the thesis as co-supervisor and also Professor Dr. Axel Tuma for serving as chair on the dissertation committee.

I would also like to thank my colleagues Helmut Kick, Ramin Sahamie, Andreas Tegel, Karl Mauch, Sebastian Geier, Bernd Löhle, and our secretary Doris Rochelle for their support, fruitful discussions and the good working atmosphere. I am also thankful to our student assistents Herbert Mayer and Laura Wieczorek for the creation of graphical illustrations and for the inspection of various planning results.

Special thanks goes to Willi Betz Logistik in Reutlingen, particularly to Jörg Schneider, Daniel Krivec and Birgit Drewitz, for their cooperation, the provision of real-life test data and for their patience in answering questions.

Furthermore, I highly appreciate the help of Rebekka Kehlenbrink, Zachary White and Sue Tolson, who proof-read the dissertation's English version.

And, last but not least, I like to thank my family and my friends, the source of love, energy, inspiration, and joy that fuel any endeavor in my life.

Augsburg, January 2011 Steffen Schorpp

Contents

Abbreviations

ADAC	Allgemeiner Deutscher Automobil-Club e.V.
AETR	European Agreement concerning the Work of Crews of Vehicles engaged in International Road Transport
AGV	Automated Guided Vehicle
anticip.	anticipation
avg.	average
BAG	Bundesamt für Güterverkehr
BGL	Bundesverband Güterkraftverkehr Logistik und Entsorgung e.V.
cap.	capacitated
constr.	constraint
cp.	compare
CPLEX	Optimization software package (solver) for linear, integer and quadratic programming problems
CPU	Central Processing Unit
D	Delivery
DARP	Dial-A-Ride Problem
diff.	different
DL	Delivery Location
DOD	Degree of Dynamism
DT	Delivery Time
dyn.	dynamic
e.g.	for example
EBIT	Earnings before Interest and Taxes
EC	European Commission
EC15	Directive 15/2002 on working hours of persons performing mobile road transport activities
EC561	Regulation 561/2006 on driving and rest period restrictions in the European Union
EDOD	Effective Degree of Dynamism
EDOD-TW	Effective Degree of Dynamism, considering Time Windows
EDT	Earliest Delivery Time
EPT	Earliest Pickup Time
et al.	and others
EU	European Union
FCFS	First Come First Served
GA	Genetic Algorithm
GHz	Gigahertz
GPS	Global Positioning System
GSM	Global System for Mobile Communication

h .... hour(s)
i.e. .... that means
ILOG .... An IBM company for enterprise software products; often synonymously used for the application of OPL and CPLEX solver
LDT .... Latest Delivery Time
LPT .... Latest Pickup Time
min .... minute(s)
MIT .... Massachusetts Institute of Technology
MLPDP ..... Multi Load - Pickup and Delivery Problem
MLPDPTW .. Multi Load - Pickup and Delivery Problem with Time Windows
MNS .... Multiple Neighborhood Search
netw. .... network
no. .... number
OPL .... Optimization Programming Language (by ILOG)
P .... Pickup
PC .... Personal Computer
PDP .... Pickup and Delivery Problem
PDPTW ..... Pickup and Delivery Problem with Time Windows
PL .... Pickup Location
PT .... Pickup Time
RAM .... Random Access Memory
s .... second(s)
SCM .... Supply Chain Management
sim speed .... simulation speed
SLPDP ...... Single Load - Pickup and Delivery Problem
SLPDPTW .. Single Load - Pickup and Delivery Problem with Time Windows
tkm .... tonne kilometers
TRP .... Traveling Repairman Problem
TSP .... Traveling Salesman Problem
TSPTW ..... Traveling Salesman Problem with Time Windows
TW .... Time Window
uncap. .... uncapacitated
US .... United States (of America)
VMI .... Vendor Managed Inventory
VND .... Variable Neighborhood Descent
VNS .... Variable Neighborhood Search
VRP .... Vehicle Routing Problem
VRPTW ..... Vehicle Routing Problem with Time Windows

List of Figures

List of Tables

Chapter 1

Introduction

This chapter starts with a motivation of the investigated topic "Dynamic Fleet Management for International Truck Transportation with occasional transportation tasks", considering practical relevance and previous attention in the existing literature (Section 1.1). In Section 1.2 the goal of the study and the main research questions are outlined. In Section 1.3 a review of developments towards the application of Dynamic Fleet Management Systems is given, followed by an analysis of the freight forwarding market in Germany and Europe (Section 1.4). Finally, this work's course of action is explained (Section 1.5).

1.1 Motivation

Tour planning has been a popular field in Operations Research since the 1960s (Dantzig and Ramser, 1959). A large number of researchers have dealt with all kinds of static problems, developing a huge variety of procedures. In recent years, the *center of attention has* also *moved to the field of dynamic tour planning problems*, where new information evolves concurrently to the plan execution and has to be handled efficiently by the dynamic planning approach (see Chapter 2 for a detailed definition of the term *dynamic*).

Many authors have developed dynamic procedures, basically extensions to already existing static ideas. In the majority of cases, the dynamic publications were focused on *local area* problems with the goal of minimizing traveled distance (see Chapter 3 for a Literature Review). There are only a few works available so far that have dealt with dynamic *wide area* planning problems.

This finding may be due the *predominance of line transportation* in wide area environments: A recurring medium-term plan – effective for several weeks or months – is generated and constitutes each vehicle's circulation between the nodes of a *fixed transportation network* (e.g., operation of a driving route from location A to B every Monday morning). New requests are fed into the driving routes of the existing line operation schedule, which, for example, induces advantages in consolidation of less-than-truckload requests. Short-term dynamic planning is not necessarily required.

However, it can be observed that some internationally operating freight forwarding companies – Willi Betz International, LKW Walter, Hindelang, etc. – have also successfully specialized in another type of *wide area* freight transportation: in *occasional transporta-*

Steffen Schorpp, *Dynamic Fleet Management for International Truck Transportation,*
DOI 10.1007/978-3-8349-6675-9_1,

tion, independent of predefined networks – also referred to as "ad hoc" or "tramp transportation" (Falk, 1995). Since there is no line schedule on a fixed network, there is the need for dynamic replanning to react in the short term on newly occurring requests and other changing information. Primary objectives are the minimization of empty traveled distance, the minimization of delay and high vehicle utilization.

Today, those companies do not apply any dynamic algorithms for their tour planning. Planning tasks, like order-to-vehicle assignment, vehicle routing and scheduling are performed completely manually by human dispatchers. This raises the question of whether there has been sufficient consideration of the planning problem *dynamic International*[1] *Truck Transportation with occasional transportation tasks* in the existing literature.

There is a small group of available publications that cover the specified planning problem: e.g. Powell (1996), Powell et al. (2002) and Yang et al. (2004). However, one drawback is that most of the specific *European real-life requirements for long-haul transportation* have been neglected in these publications. Even in the available static literature, requirements for International Truck Transportation are only *partially* considered, in some of the latest publications (e.g. Kok et al., 2009). A *complete* consideration of such requirements, however, is needed for a freight forwarding company's[2] planning system, since compliance with requirements (like EC social regulations) is statutory.

In addition to the incomplete real-life requirements, the available (static) wide area publications have only been tested with self-generated artificial test data. The applied test data sets all possess an unrealistically small quantity of vehicles and orders, and contain an inadequate duration of maximum six days. This is not sufficient to test a *wide area* planning procedure, since some restrictions take effect only over a horizon of several weeks. Due to their static character, the available publications do not consider any implementation aspects, which, however, are a crucial component in getting a dynamic planning system up and running at a freight forwarding company.

As far as we know, there is no work available that includes all the subsequent aspects that are important in practice: (i) development of a Dynamic Fleet Management System for International Truck Transportation focusing on occasional transportation tasks, (ii) comprehensive consideration of all important real-life restrictions for Europe-wide Truck Transportation, e.g. EC social regulations, working time restrictions, traffic bans, etc., (iii) test of the procedure with a sufficiently large real-life data set (in terms of duration, number of orders and number of vehicles), and (iv) benchmark with actual planning performed at a freight forwarding company.

Due to the obvious negligence of this field of dynamic tour planning, the untreated aspects are selected for investigation in this work.

---

[1] We use the term *international* with a focus on Europe; synonymously with *wide area* or *long-haul*.

[2] The term *freight forwarding company* (German: Spedition) is primarily directed at the *agency function* of organizing shipments for individuals or other companies. A *freight forwarding company* is often not active as a *carrier* and outsources the actual execution to *road haulage companies* (German: Transportunternehmer/Frachtführer) (Bundesverband Güterkraftverkehr Logistik und Entsorgung (BGL) e.V., 2010). In this work, however, we use the term *freight forwarding company* for both types of company.

1.2 Goal of the Study and Problem Outline

The *goal of the study* is:

> *To design a Dynamic Fleet Management System for International Truck Transportation focusing on occasional transportation tasks that is capable of improving the planning process at a freight forwarding company in terms of empty traveled distance and service quality, hereby taking into account all important European real-life requirements (EC social regulations, working time and traffic bans).*

In the following, a number of *research questions* are posed. These questions outline the research problem in more detail and shall guide us in reaching the goal of the study.

What are the specific characteristics of dynamic planning problems?

Dynamic planning problems differ in many aspects from static ones. Before a new dynamic planning procedure is developed, the specific characteristics of dynamic planning problems have to be elaborated. They can be helpful for adjusting a new planning procedure in order to meet the specific needs of a dynamic planning problem.

Where do dynamic planning situations occur in real-life?

Before a new dynamic planning procedure is developed, it is also interesting to evaluate where dynamic planning situations actually occur in real-life and to what theoretical planning problems they can be connected. This analysis legitimates the treatment of the dynamic planning problem that was chosen in this work. It also helps to assess the real-life value of other dynamic publications from the literature.

What is the state of the art in the literature on Dynamic Fleet Management?

Before designing a solution method for the selected real-life planning problem, we need to familiarize ourselves with the state of the art in the literature on this topic. Since the literature on dynamic wide area applications is scarce, we discuss the literature on dynamic routing problems in general. The investigation of algorithm orientated papers gives us an idea, with what procedures best performance could be achieved.

What dynamic solution approaches are suitable for a Dynamic Fleet Management System?

We develop a choice of two planning approaches with two completely different planning ideas. First stage, the approaches are not designed for the final planning problem with all its real-life restrictions, but instead, for a simplified local area problem. The available test instances for this local area problem are used to perform extensive tests and to evaluate the strengths and weaknesses of both procedures. One procedure is finally chosen for adaptation to the real-life planning problem.

What general requirements come along with International Truck Transportation?

In order to achieve an operable Fleet Management System, we need to elaborate the real-life restrictions that have to be actually considered for Europe-wide Truck Transportation. EC social regulations, working time restrictions and traffic bans are analyzed in detail. The main restrictions are chosen for inclusion in the real-life planning procedure.

What specific requirements are necessary to cover the planning situation at the cooperating freight forwarding company?

Incorporating the general planning requirements does not necessarily result in actual real-life applicability. The specific planning situation at a freight forwarding company, which comes along with additional restrictions, has to be considered as well. To this end, we perform a detailed analysis of the planning process and of the planning data of our cooperating freight forwarding company and adjust our planning procedure to the specific situation.

What simulation speed should be used to evaluate the real-life planning procedure's performance?

In the selection process of an appropriate procedure and at the final calculations with the real-life test data set, we face the question of how to run simulations: with high or slow speeds. A high simulation speed produces results faster and therefore allows for more simulation runs. This can be an advantage, since it allows for a higher number of different parameter variations to be tested. A slow simulation speed (e.g., real time simulation), however, allows for more improvement calculations to be executed during each simulation run. Therefore, this type of simulation is supposed to produce a better overall solution quality (at least, if the planning procedure is capable of using the available time).

How much potential savings can be generated with the application of a computer-based dynamic planning system for International Freight Transportation? Is it reasonable to implement such a Decision Support System?

We compare the results that can be achieved with the newly developed planning procedure (for a five-week real-life data set) with the manual planning performed at our cooperating freight forwarding company (benchmark). From this, we derive the potential savings in empty traveled distance and delay that can be generated with the application of our computer-based planning system. This is followed by a discussion of the pros and cons of an actual implementation of such a Decision Support System.

1.3 Developments towards Dynamic Tour Planning

There are several developments that promote the popularity of Dynamic Tour Planning. Significant *advances in information and communication technologies* contribute the necessary technical requirements. *Increased competition* and *rising environmental awareness* call for efficient planning and the economical use of resources.

In addition, there is also a *consolidation* and *growth* effect (more vehicles and more orders per company) that makes it more difficult to achieve good planning results by manual planning, thus triggering the implementation of a computer-based Decision Support System for Dynamic Fleet Management.

Subsequently, these aspects are explained in more detail.

- **Advances in information and communication technologies:** Today, the *real time determination of a vehicle's position via GPS* is state of the art. *Mobile communication* can be used to exchange information between planning center and driver.

  Figure 1.1 shows an exemplary information flow in a GPS-based dynamic Fleet Management System (cp. Larsen, 2000; Goel, 2007). The vehicle in motion receives at least three positioning signals from GPS satellites and calculates its current position. This position is transferred via GSM cellular phone network to the planning center. The planning center knows all the vehicles' status, all the open orders etc. and produces a preferably good vehicle dispatching (order to vehicle assignment, vehicle routing, vehicle scheduling). Afterwards, planning decisions and updates are transferred via GSM back to the vehicle. Communication between planning center and driver may be text message based as well as by phone.

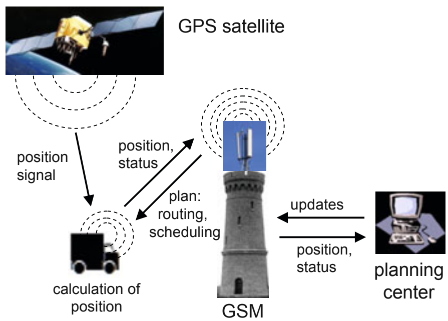


Figure 1.1: Information flow in a dynamic routing system (Larsen, 2000)

  In addition, the use of the latest *digital road maps* and the *inclusion of actual traffic information* to calculate shortest paths between Pickup and Delivery locations has become standard. Route planners (e.g. PTV map&guide, Tele Atlas) are employed at the planning center as well as directly at the vehicles.

  A further aspect concerning information technology is the *improved capacity of computing systems*. At the planning center today, a single personal computer is able to quickly handle large amounts of data, e.g. processing the complete planning of a truck fleet operating Europe-wide.

- **Increased competition:** The introduction of a *single European transportation market* has reduced market barriers and allows freight forwarders from each European country to offer international transportation services over the entire European territory.

  Cabotage (the execution of a transportation service within a country by a foreign freight forwarder), however, is still restricted: according to regulation EC 1072 of 21.10.2009 (European Union, 2009) cabotage is only allowed subsequent to an incoming international transport. In such a case, a vehicle may only execute a maximum of three cabotage transports, with the last Delivery having to be finished within seven days of completion of the incoming international transport (Freight forwarders from Bulgaria and Romania are excluded from this partial authorization of cabotage until 31.12.2011).

Since 01.01.1994, *prices* for freight transportation are no longer subject to regulation and can be *freely negotiated with regard to supply and demand* (Staub et al., 2004). Internet market places (freight exchanges) allow a shipper to find the cheapest freight forwarder, while freight forwarders may offer free transportation capacity to the highest bidder. In addition to reduced geographical market barriers, such freight exchanges provide order mediation as well as market transparency concerning supply, demand and prices. This is of special interest for small and medium-sized freight forwarding companies that do not have their own distribution channels.

Furthermore, a *harmonization and simplification of the general conditions in the European countries* has been initiated. Since 2006, driver based social regulations (see Section 5.1 for further details) are the same over the whole of Europe. However, there are still major differences in taxation (e.g., company and fuel taxation) as well as labor costs. While average yearly labor costs per employee in the transportation sector in 2005 was at a level of € 45,000 in Belgium and € 32,700 in Germany, an employee in Bulgaria only earned € 3,900 (Eurostat, 2009).

Profit margins in the freight forwarding sector are *quite low*. According to Commerzbank Research (2010), the average EBIT (Earnings Before Interest and Taxes) in the German freight forwarding sector was 4.2% (reference year: 2006), with a spread between -3.0% and 13.6%. An effective real time planning system may help to gain competitive advantages and to ensure a company's survival.

- **Environmental awareness:** In the recent years, *environmental awareness* has attracted a great deal of attention in politics and the public. Freight forwarders are in specific focus, since 26% of the EU-27's total energy consumption can be attributed to road transport (reference year: 2006; Eurostat, 2009). Efficient planning, e.g. by minimizing empty traveled distance, therefore, is not only a necessity for direct cost reduction, but also a socially desired objective.

- **Growth:** In a work by Powell (1996), it is reported that a higher number of available vehicles spread over the area of execution produces a smaller percentage of empty traveled distance (economies of density). This is an intuitive finding. The drawback, however, is the *increasing planning complexity*. A human dispatcher who is simultaneously managing a large number of vehicles will be barely able to "optimally" react to a large amount of very frequently changing data (realistic dimension for a big freight forwarder: 1000 vehicles in motion, and 2000 open transportion orders). In the case of several human dispatchers, who manage distinctive subproblems, the generation of a "globally optimal solution" is becoming even more unlikely.

  A computer based dynamic Fleet Management System, however, can create "good solutions" in quick response to a large, varying information base, simply because there is enough computation power to consider the planning problem as a whole and to evaluate many more possible planning options in the short period of available reaction time.

1.4 Market Analysis

A look at the *European, and specifically the German, freight forwarding market* shows a strong growth in truck transportation volumes over the recents years. In the following, some statistics are presented which confirm this statement and which also try to explain this development. In addition, trends in prices and in the number of freight forwarding companies are considered.

The freight forwarding sector is an important economic factor, accounting for 7% of the European Union's value added and employing 8.7 million people (reference year: 2005). In the years from 1995 until 2008, the European transportation market grew by 33.6%, from 3,060 billion tkm to 4,090 billion tkm. Figure 1.2 shows the transportation performance of the modes road, sea, rail, inland waterway, pipeline, and air. Over the whole considered horizon, it can be observed that road transportation accounts for the highest transport performance with a successively increasing gap to the other modes. While in 1995 only 42.1 %tkm of the total goods transportation were executed by truck, in 2008 the share of truck transportation increased to 45.9 %tkm (Eurostat, 2009).

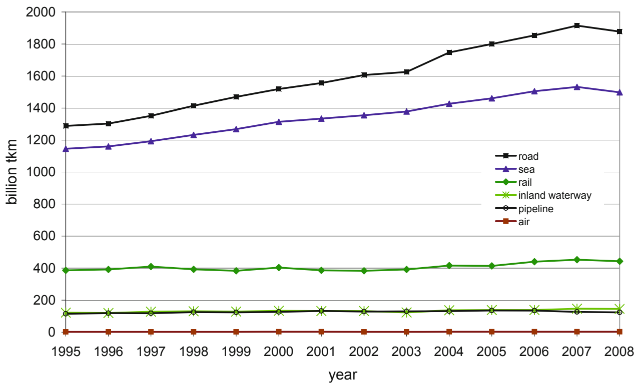


Figure 1.2: Modal split EU27: 1995 - 2008 in billion tkm

From many shippers' perspectives freight transportation by truck seems to remain the preferred choice. In a survey, Pfohl and Schäfer (1998) asked 134 companies that order logistical services how they would rate the performance of different traffic carriers (truck, airplane, rail, waterway sea, waterway inland, and intermodal transport) for seven key indicators on a scale from 1 (worst qualification) to 5 (best qualification). The results are presented in Figure 1.3.

In terms of *transportation time*, *network connections*, *flexibility*, and *reliability*, truck transportation is rated best. Interestingly, the *transportation costs* per truck also outperform the other traffic carriers. Only in *adherence to schedule* and *tracking and tracing* the airplane is rated slightly better.

In the following, the specific development of the German freight forwarding sector is analyzed (Statistisches Bundesamt Deutschland, 2010; Commerzbank Research, 2010):

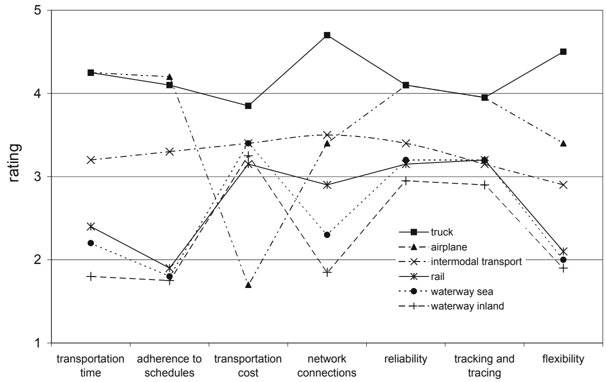


Figure 1.3: Assessment of different transportation modes from a shipper's perspective

In Figure 1.4 the sales volume is processed for the years 2001 until 2009. A substantial growth in sales volume (+68%) can be observed between 2001 and 2008, reaching a maximum sales volume of € 74.74 billion in 2008. Afterwards, the world financial crisis caused a reduction in sales volume to € 62.63 billion in 2009.

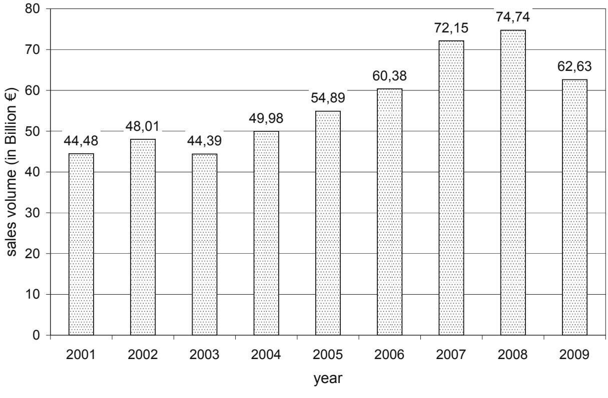


Figure 1.4: Sales volumes in the freight forwarding sector (Germany): 2001 - 2009

The previously mentioned growth of the European transportation performance (in tkm) of +33.6% over a 13-year horizon, and the development of the sales volume in the German freight forwarding sector (in €) of +68% over a 8-year horizon cannot be directly compared. Nevertheless, these numbers suggest that an overproportionately high share of the European growth was generated by German freight forwarders.

A look at the prices that had to be paid by the shippers is given in Figure 1.5. Especially in 2005 and 2006, freight forwarders were able to achieve higher freight rates (+7.2% and +8.1%, respectively). In 2009, however, the reduced transportation volume led to overcapacities, resulting in decreasing market prices (-1.7%).

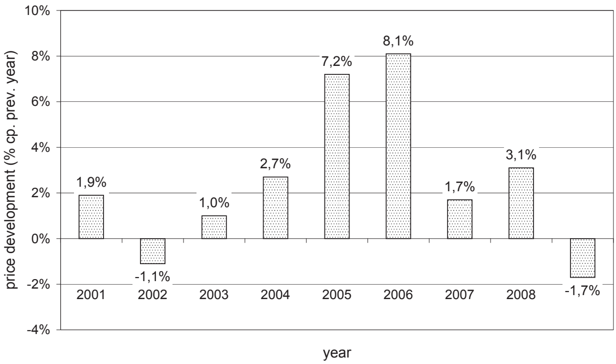


Figure 1.5: Development of prices for freight forwarding services (Germany): 2001 - 2009

The last statistic deals with the development of the total number of freight forwarding companies in Germany (Figure 1.6). Similar to the increases in sales volume, the number of freight forwarders increased by 41% between 2001 and 2008. In 2009, however, a substantial decrease in the number of freight forwarding companies is reported.[3]

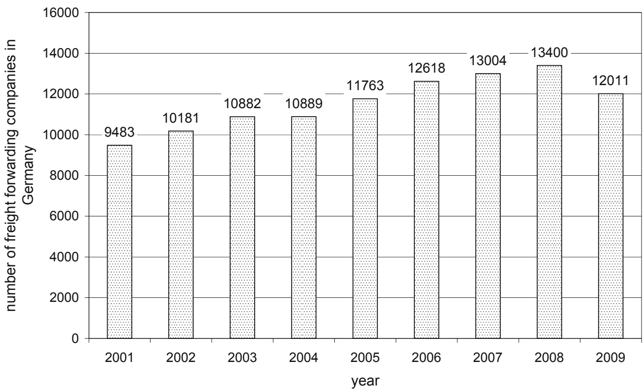


Figure 1.6: Number of freight forwarding companies (Germany): 2001 - 2009

1.5 Course of Action

This section describes the main contents of the following chapters. The sequence of investigated topics roughly complies with the sequence of research questions that have been proposed in Section 1.2.

---

[3] It should be mentioned that this statistic also includes very small freight forwarding companies. Here, the minimum requirement to be counted as a freight forwarding company was a sales volume of at least 17,500 Euro

Chapter 2 includes a general **characterization of dynamic vehicle routing problems** and proposes a configuration framework for such kind of problems (Section 2.1). Furthermore, the most relevant dynamic real-life applications are elaborated and connected with the associated theoretical problem definitions (Section 2.2). Finally, two relevant dynamic problem specifications (a local and a wide area one) are selected and characterized for further consideration (Section 2.3).

Chapter 3 includes a detailed **literature review** of publications on *Dynamic Fleet Management*. At the beginning, some general statistics on the surveyed literature are given (Section 3.1), followed by some exemplary publications showing the variety of dynamic applications in real-life (Section 3.2). Afterwards, algorithm orientated papers are classified into three groups, depending on the knowledge of the future (Section 3.3). The first two groups do not have any knowledge of the future and therefore only perform *myopic* planning. In contrast to the first group, the second group, however, tries to anticipate the future. Stochastic information about the future is available only in the third group of publications, where the algorithms make explicit use of such information.
The remaining sections review the most popular dynamic test instances in the literature (Section 3.4) and present the results of some papers that do not primarily focus on the algorithmic performance but on the acceptance of dynamic planning applications in real-life (Section 3.5).

In **Chapter 4**, **two dynamic planning procedures are developed and evaluated:** Multiple Neighborhood Search (Section 4.1) and an Assignment based procedure (Section 4.2). For reasons of simplicity, the procedures' basic versions are directed to the local area capacitated MLPDPTW (covering the dynamic real-life application of Dial-A-Ride Services) and not to the final real-life planning application.
Both procedures' specific characteristics are compared in Section 4.3, elaborating the main differences. Afterwards, some test data sets - self-generated as well as taken from the literature - are introduced (Section 4.4). These data sets are used for a comparison of the procedures' performance and also to gain some general insights in dynamic problems (Section 4.5). Finally, one procedure is chosen for adaptation to the real-life scenario (Section 4.6).

In **Chapter 5** the selected basic solution approach is adapted to perform actual **real-life application at an Internationally Operating Freight Forwarding Company**.
Firstly, the requirements that have to be considered for long-haul transportation in Europe are elaborated (Section 5.1). Afterwards, the actual planning process at the leading European freight forwarding company is drafted (Section 5.2). Section 5.3 describes the adaptation of the Multiple Neighborhood Search procedure to the real-life planning situation. In Section 5.4 the preprocessing of a five-week real-life test data set and the derivation of benchmark objective function values are explained. Finally, the computational results that are achieved with the adapted Multiple Neighborhood Search for this real-life data set are reported (Section 5.5).

In **Chapter 6** methodology, achievements and main findings of the study are summarized (Section 6.1). Afterwards, some recommendations for further research in the field of *Dynamic Fleet Management* are proposed (Section 6.2).

Chapter 2

Dynamic Transportation Problems

This chapter begins with a general characterization of dynamic problems, specifically pointing out the differences between dynamic and static problems and proposing a configuration framework for dynamic algorithms (Section 2.1). Subsequently, the most relevant dynamic real-life applications are elaborated and connected with the associated theoretical problem definitions (Section 2.2). Finally, two dynamic problem definitions that have been chosen for further investigation in Chapters 4 and 5 are introduced (Section 2.3).

2.1 Characteristics of Dynamic Problems

As a preliminary matter, the **term "dynamic"** (also referred to as "real time" or "on-line") needs to be defined. One of the earliest references related to Vehicle Routing is Psaraftis (1988):

> *"In a dynamic vehicle routing problem, inputs may (and generally will) change (or be updated) during the execution of the algorithm and the eventual execution of the route. Algorithm execution and route execution are processes that evolve concurrently in a dynamic situation, in contrast to a static situation in which the former process clearly precedes (and has no overlap with) the latter."*

In a definition by Pankratz (2005), the need for "irreversible" decisions under incomplete information is emphasized:

> *"planning and execution are overlapping processes, and planning decisions, which may be irreversible, have to be taken before all problem data become known."*

A dispatcher is therefore forced to plan in a sequential or **rolling horizon** manner. He solves a part of the overall problem on the basis of the information available at the present moment. This partial problem is denoted as **static subproblem**. When new information arrives the static subproblem changes and the dispatcher has to resolve the problem.

The static subproblem is not only affected by newly occurring information, but also by **fixation**. Fixation is defined as the successive alteration of provisional planning decisions into permanent ones. It is basically triggered by proceeding time, i.e., if an event's scheduled execution time is reached, the event is permanently fixed. A fixation reduces

Steffen Schorpp, *Dynamic Fleet Management for International Truck Transportation,*
DOI 10.1007/978-3-8349-6675-9_2,

the available planning options and therefore is a second source of changes to the static subproblem.

The static subproblem's **planning horizon** depends on the applied solution approach: it ranges from the present moment up to the last scheduled activity. All decisions are of provisional type and can be changed by an improvement procedure, unless they are finally fixed.

There are several **sources** inducing dynamism of a vehicle routing problem: New customer orders, cancelation or modification of already known customer orders, revealment or changes of actual demand level and customer service time, changes in vehicle travel time (due to unforeseen events, such as traffic jams or unexpected delays), up to complete vehicle breakdown.

Differences: Dynamic vs. Static

To identify the specific characteristics of a dynamic problem, a comparison with a classic static problem is useful. A whole catalog of such differences was defined by Psaraftis (1988) and is subsequently quoted in a shortened form:

- **Time dimension is essential:** In static vehicle routing, time dimension may or may not be an important factor in the problem. If there is a scheduling component alongside the routing component, time dimension is essential. Actually, most classic generic routing problems, such as Traveling Salesman Problem (TSP) or Vehicle Routing Problem (VRP), do not have a scheduling component. In contrast, time dimension is essential in every dynamic vehicle routing situation, whether it is time constrained or not. It is necessary to keep track of how vehicle schedules and scheduling options evolve dynamically over time.

- **Problem may be open-ended:** In a static situation, the duration of the routing process is more or less bounded or known in advance. The duration of such a process in a dynamic situation may neither be bounded nor known.

- **Future information may be imprecise or unknown:** In a static context, information about all problem inputs is assumed to be of the same quality, irrespective of where within the schedule this input happens to be (beginning, middle, or end). This is not the case in a dynamic problem, in which information on any input is usually precise for events that happen in real time, but more tentative for events that may occur in the future. Probabilistic information about the future may be available.

- **Near-term events are more important:** Because of uniformity of information quality and lack of input updates, all events (whether in the beginning, in the middle, or at the end of a vehicle's route) carry the same "weight" in a static context. In dynamic routing, it would be unwise to immediately commit vehicle resources to requirements that will have to be met in the distant future. This is because other intermediate events may render such decisions suboptimal, and because such future information may change anyway. Focusing on near term events is therefore an essential aspect of dynamic vehicle routing.

- **Information update mechanisms are essential:** Virtually all inputs into a dynamic routing problem are subject to revision at any time during the execution of the route. Therefore, update mechanisms are an integral part of the algorithmic structure. Data structure and database management techniques that help revise problem inputs efficiently, as well as adeptly figure out the consequences of such revisions, are central to a dynamic routing scheme. In contrast, in a static scenario, such mechanisms are not necessary.

- **Resequencing and reassignment decisions may be warranted:** In a dynamic vehicle routing situation, the appearance of a new input may render decisions that have already been made prior to that input's appearance suboptimal. This fact concerns both sequencing and assignment decisions. Thus, the appearance of new input may necessitate either the resequencing of the stops of one (or more) vehicle(s), or the reassignment of those vehicles to demands requesting service (or both).

- **Fast computation times are necessary:** The need to reoptimize routes and/or vehicle assignments on a continual basis in real time necessitates computation times faster than those necessary in a static situation. In a static situation, computation runs may take several hours or overnight. In a dynamic routing situation, if new information is available, the dispatcher wishes to know the solution to a particular problem as soon as possible (within minutes or seconds).

- **Indefinite postponement mechanisms are essential:** Indefinite deferment means that the service of a particular demand can be postponed indefinitely due to that demand's unfavourable geographical characteristics relative to other demands. The problem can be handled by introducing time constraints.

- **Time constraints may be different:** Dynamic routing inputs, such as Earliest Pickup Time (EPT) and Latest Pickup Time (LPT), tend to be softer than in a static situation. If a "hard" deadline makes a routing problem infeasible, it is far better to renegotiate that deadline so as to make it feasible than it is to declare infeasibility and quit.

- **Flexibility to vary vehicle fleet size is lower:** In theory, another alternative to denying service to a customer, if a time constraint cannot be met, is to add an additional vehicle, at a cost, to serve that customer. However, this proposition may not necessarily be viable in a dynamic vehicle routing because it may not be possible to have access to backup vehicle resources in real time. In a static situation, the time gap between execution of the algorithm and execution of the route is usually long enough to allow for such a decision to be made.

These aspects are complemented by a comparison of the *open order characteristic* in a static and in a dynamic environment (cp. Sandvoß, 2002). In the static context, all orders are initially known, being successively completed as time proceeds. The number of unfulfilled orders decreases monotonously. Figure 2.1 shows an exemplary curve.

In a dynamic problem, however, the initial order level is lower (or non-existent), since dynamic orders are revealed later (up to time T). This may result in time intervals of an increasing number of open orders when the number of new occurring orders exceeds the number of completed orders. After the latest Call-In time T, the number of unfulfilled orders monotonously decreases like in the static case (cp. Figure 2.1).

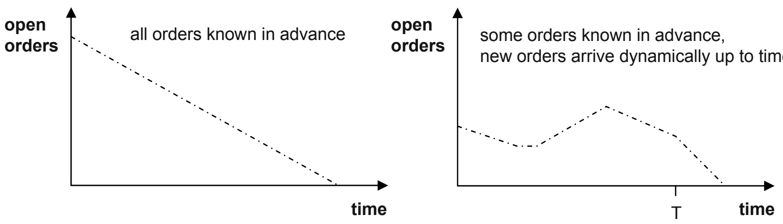


Figure 2.1: Open order characteristics for static and dynamic problems

Degrees of Information Availability and Possible Reactivity

In the following, four problem categories are differentiated, based on *initial information availability and information certainty* as well as *possible reactivity*:

- **static**,
- **static stochastic**,
- **dynamic stochastic**, and
- **dynamic**.

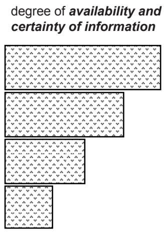
Initial information availability is related to the "quantity of information" that is known to the decision maker before the planning horizon starts. We use the word "complete" to denote the condition of having information about all relevant facts available (e.g. about all customers and the associated locations, all specific demand levels, etc.). In contrast, the word "incomplete" is used for the situation of initially not having all basic information ready. Thus, some customers, demands, etc. may be revealed during the planning horizon.

However, the availability of the initial information predicates nothing about initially and subsequently revealed *information's certainty*: We call information "certain" if it is given as an exact deterministic value. Otherwise, the information is subject to "uncertainty", e.g. when a probability distribution is given instead of an exact value, or if the exact value may be subject to further changes.

Figure 2.2 shows the associated characteristics of *initial information and information certainty* for the regarded problem categories (column one). The second column describes the possible availability of stochastic information about the future. To the right, some bars are plotted in order to symbolize the relative "degree of availability and certainty of information". The size of the bars is related to column one and two, and drops from the first to the fourth problem category, indicating lowest information availability in category four.

Afterwards a third column is given that states whether the associated problem category allows for "replanning" during plan execution or not. The bars to the right visualize the relative extent of *possible reactivity* and show increased reactivity for categories three and

	initial info, certainty?	stochastic information	replanning capability
static	complete, with certainty	no	no
static stochastic	complete, with uncertainty	yes	no
dynamic stochastic	incomplete, with uncertainty	yes	yes
dynamic		no	yes



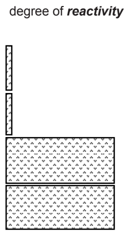


Figure 2.2: Information availability and replanning options for different planning categories

four, compared to categories one and two, which do not allow for replanning at all.

In the first category, **static**, all information is "completely" available before the planning horizon starts and is not subject to any changes ("certainty"). Stochastic information of the future and a re-planning option are neither available nor required. This case corresponds to the classical static planning situation, hence several traditional procedures may be applied in order to find the best solution.

In the second category, **static stochastic**, initial information is "complete" but subject to "uncertainty". Parts of information are given as discrete or continuous random variables, for example, approximated through the use of historical data. The static character requires an *initial planning* run to construct an "a-priori" solution that optimizes the expected value of a given objective function (first stage). All possible decisions are fixed, including some strategies on "how to react to unexpected developments". Later changes of these recourse strategies are not possible ("no re-planning").

When the routes are actually executed in the second stage, these initially specified strategies (*recourse actions*) are applied to the first stage solution in order to address the current realization (e.g., "skip a customer who does not show up", or "send the vehicle back to depot before it resumes its tour because of capacity shortage"). Typical representatives are the "Probabilistic TSP", "VRP with stochastic customers", "VRP with stochastic demands", and "VRP with stochastic customers and demands". (cp. Ichoua et al., 2006)

The third category is denoted as **dynamic stochastic**. Initial information is "no longer complete". Instead, some information is revealed during the planning horizon. In addition, there is "no certainty" about the given information, which may be subject to repeated changes. As in the second category, some stochastic information on the future is available, partially compensating for the lack of basic information quantity. There is no longer the need to decide on reaction strategies "a-priori", instead, an appropriate online replanning may be performed in a rolling horizon manner.

In order to deal with the stochastic information, several approaches can be chosen. In a *sampling approach*, for example, the algorithm generates a sufficient number of future scenarios (by drawing from the given probability distributions) and uses the scenarios to find a good and robust solution at each rolling horizon step. In contrast thereto, a *stochastic algorithm* explicitly incorporates the current information and probabilities of future events into its objective function: The given probabilities are used for calculation of "recourse functions", which include the costs that the assumed solution scenario does not occur. Other approaches *directly derive some measures*, like "vehicle re-allocation" or "scheduling of extra waiting times in promising regions" when certain probabilities exceed some threshold.

In the fourth category, **dynamic**, initial information (if available at all) is "incomplete". All initial and later revealed information is subject to changes ("uncertainty"). Information about the future is not available. However, there is a *replanning capability* available for appropriate reaction to dynamic information. In comparison to the previous categories, the degree of initial information availability and certainty is the lowest. However, especially in comparison with the static stochastic situation, this lack of information may be compensated by the dynamic replanning capability. A dynamic planning situation is usually handled with a rolling horizon planning approach, which includes the new obtained information step-by-step.

Measuring Dynamism

In the previous explanations, several sources that may cause dynamism were mentioned, especially dynamic requests and dynamic travel times. According to Larsen (2000), a measure for dynamism shall quantify the extent of new information emerging during the operational phase of the system, thus being helpful for evaluating and comparing the "difficulty" of various problem instances.

A first "request-related" measure was proposed by Lund et al. (1996) and Larsen (2000). They define the *basic degree of dynamism (dod)* as the number of dynamic requests n_{dyn} relative to the total number of requests n.

$$dod = \frac{n_{dyn}}{n}$$

Larsen et al. (2002) distinguish three levels of dynamic systems, based on the degree of dynamism. First, *weakly dynamic systems*, with up to 20% dynamic orders, where reaction time is considerably longer compared to other dynamic problems (e.g. distribution of heating oil to private households, residential cable and telephone repair services, or the transportation of the elderly and physically disabled). The second group consists of *moderately dynamic systems*, having a substantial part of dynamic requests (20% up to 80%). As typical applications here, overnight mail services and appliance repair are mentioned, where scheduled customers are interspersed with dynamic ones that need immediate attention due to the gravity of their request. Finally, there are *strongly dynamic systems* with more than 80% of dynamically occurring customers, where frequent changes in data have to be handled within minimal response times (e.g. taxi and emergency services).

The same authors have also formulated extended versions of the "degree of dynamism". The first extension additionally incorporates the relative position of the Call-In time $t_i \in (0 < t_i \leq T)$ of the dynamic order i, in relation to the latest possible Call-In time T.

The resulting measure is called *effective degree of dynamism (edod)*:

$$edod = \frac{1}{n}\sum_{i=1}^{n_{dyn}} \frac{t_i}{T}$$

Further refinement is achieved by considering the temporal gap between an order's Call-In time and its time window. The time window of request i is specified by earliest service time e_i and latest service time l_i. The time gap between t_i and l_i is called *reaction time* and is used to calculate the measure *edod-tw*:

$$edod - tw = \frac{1}{n}\sum_{i=1}^{n} \frac{T-(l_i - t_i)}{T}$$

Figure 2.3 visualizes the concept of *edod-tw*: t_1 and t_2 symbolize different Call-In times with regard to the same time window. Since $t_2 > t_1$, there is a shorter reaction time in the second case.

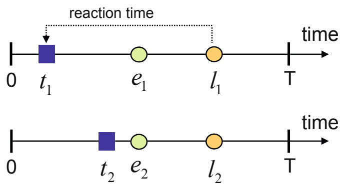


Figure 2.3: Effective degree of dynamism - with time windows

The work of Tjokroamidjojo et al. (2006) investigates the impact of different *reaction times* for a Single Load Pickup and Delivery Problem with Time Windows (SLPDPTW). It is shown for a theoretical test environment that the use of advance load information (with longer reaction time) produces significantly better results compared to a situation with Call-In at the latest possible time. A mandatory assumption for such a statement is an adequate replanning option in order to use the available "reaction time". The authors show that the absence of such an option may completely invalidate the advantage of "early information".

Due to the fact that dynamic requests can be characterized as the most important source of dynamism, these specific measures have their warranty. On the other hand, further possible sources of uncertainty are completely neglected. An approach which tries to cover other sources of dynamism is proposed by Schumann et al. (2009). Unfortunately, it is rather a theoretical type of consideration.

The authors extend Larsen's *edod-tw* formula. Instead of all orders n, they just consider dynamic orders n_{dyn} and propose the inclusion of additional dynamic events. The authors select solely dynamic events $\gamma \in \Gamma$ that may invalidate the current plan or may decrease solution quality. The set Γ includes (i) decrease of l_i, (ii) increase of e_i, (iii) increase of travel time between two nodes, (iv) increase of a customer's demand and (v) occurrence of new requests. The proposed dynamic measure ϕ is calculated with the formula:

$$\phi = \frac{1}{|\Gamma|}\sum_{\gamma}^{\Gamma} \frac{T-(h_\gamma - t_\gamma)}{T}$$

The value of h_γ depends on the specific event: (i) if l_i is decreased, h_γ is chosen to be the new value of l_i; (ii) if e_i is increased, and (iii) if a link's travel time increases, the first customer in the tour, whose time window cannot be met any longer, is identified. Then h_γ is chosen to be this customer's latest service time; (iv) if a customer's demand increases, the first subsequent customer in the tour is identified, who can no longer be served because of limited capacity. Then h_γ is chosen to be the departure time from this node's predecessor; (v) if a new order occurs, h_γ is chosen to be the latest service time associated with the new order.

If in cases (ii), (iii) and (iv), no customer satisfying the query can be found, the formula's numerator is set to zero for the considered γ-event.

A drawback of the proposed measure is its dependency on the current plan. An identical dynamic data set may result in different levels of dynamism ϕ, depending on the performance of the underlying planning approach. In addition, "positive events", e.g. reduction in travel times, are completely neglected, even though such "positive events" also have to be handled, causing "dynamic stress" to the planning system.

In general, a meaningful measure for dynamism should incorporate three main aspects:

- the type of new information and the magnitude of changes,
- the available reaction time for the planning system between disclosure date and possibly resulting negative impact, and
- the number of changes made known to the planning system per time unit ("stress").

In addition, it should be certainly independent of the applied solution procedure.

Due to the complexity of this task, to our knowledge no such measure has been developed to date.

Configuration Framework for Dynamic Algorithms

The most important and distinctive feature of a dynamic algorithm is the capability to **quickly** produce a new **good plan** after new information has arrived. The way to perform this task can be described with the aspects of the following configuration framework (cp. Bock, 2004).

1. Technique of Adjustment

This aspect answers the question of what kind of **methodological approach** to use **to incorporate new information**. There are easy *predefined decision rules* like "Nearest Neighbor", or more advanced *heuristic approaches*, which construct a feasible plan and try to improve it by applying exchange operators (possibly guided by a metaheuristic). In addition, *exact approaches* (e.g. Column Generation based) may also be applied, either in their original "exact" setting (for small instances) or in a heuristical way (for greater instances).

Intuitively (from a static perspective), the use of a more advanced approach applied to the static subproblems should result in a better overall performance. Interestingly, this point is subject to discussion. While Yang et al. (2004) report "that fully re-optimizing each time (seeking the optimal solution) leads to overall better performance under various testing situations", Hvattum et al. (2006) experience that "a better solution to the static subproblem does not necessarily lead to a better overall solution".

In fact, in a few cases we had the same "non intuitive" experience in our tests (see Section 4.5.1). This behavior can be explained as follows. As we saw in the definition of a dynamic problem, it requires planning decisions under incomplete information, which are to some extent irreversible. Even if a "more advanced" procedure A produces a better solution than an inferior procedure B under information of time t, new information at time t+1 can completely render the situation. Perhaps procedure A has fixed some decisions (optimal at time t), which emerge sub-optimal under new information of time t+1. So, in total, it is possible that procedure A, despite its structural dominance, is no more capable of catching up with procedure B in the remaining time.

However, in the most cases we made the "intuitive" experience that more advanced procedures applied to the static subproblems also produced better overall results.

2. Reaction of Adjustment

If former planning runs have been performed, there will be an obsolete solution available (not including the newly arrived information). This second categorization aspect decides **how to use** such **old planning solutions**. One approach is to completely neglect former planning results and to build up a new feasible solution *from scratch*, incorporating all new information, treating it as equal to old information. In contrast thereto, a *constructive* approach takes the planning results from the last planning run as given and just updates it with all new information now available.

3. Frequency of Adjustment

This categorization answers the question of **how often the plan in execution is updated**, for example because of new information or improvements. Four different options can be distinguished. The first is that new information becoming available immediately triggers a new planning run and an update of the plan in execution (*event-based*). In the second option, new information is gathered over a specified period and if time is elapsed, a new planning run is started (*time driven*).

In the third option, new information is gathered up to a point where a specified number of new events has occurred, then a new planning run is started (*size driven*). The fourth option allows updates of the plan in execution, not only at every new event, but also when an improvement procedure, running simultaneously to execution, finds a better specification for the plan in execution (*continuous*).

4. Duration of Adjustment

Duration of Adjustment describes the **time available to the algorithm** for incorporating new information and for performing improvement. In a *time limit based* environment, a known finite *anticipation horizon* is available to incorporate new information. An improvement strategy can be developed so as to use the complete time available for a good search strategy (diversification and intensification; see Rochat and Taillard, 1995). Finding a value that achieves a good trade-off between execution time and solution quality is a challenging task: If the *anticipation horizon* is too small, there is not enough time available for the optimization procedure. Situations may be incorrectly assessed, resulting in inferior decisions. Otherwise, if the anticipation horizon is too long, planning opportunities may be lost due to the delayed possibility for reaction (see Ichoua et al., 2000).

In an *event-based* environment, duration is stochastic and not known to the algorithm. A feasible solution needs to be produced very fast, so that it is ready in case of the occurrence of the next event. Here, intensification strategies are especially appropriate for improvement. A further duration category is called *zero-time*. In this case, the algorithm just gets a very small amount of time (near zero) to incorporate new information into the plan in execution. Improvement is not performed.

5. Synchronization of Adjustment

In a real time planning environment, calculations of a control algorithm have to be done parallel to the progress of time in the real world. Several options are available to achieve **synchronization between algorithmic calculations and plan in execution**. The first (*prioritization of computation*) implies a stop of the execution of the current plan while calculations are performed. Afterwards, the plan in execution is updated with the new results and execution is continued. For practical reasons, this option seems to be hardly applicable.

In the second option (*extensive simultaneity*), execution and computation run simultaneously. The algorithm is allowed to change all decisions which are not due within a short *anticipation horizon*. A third approach specifies a larger part of the "plan in execution", which is not allowed to be changed (*prioritization of execution*), avoiding stress of frequent re-scheduling, but also reducing the potential solution space.

6. Scope of Adjustment

The scope describes how the algorithm is allowed to change decisions which become effective in the future. The scope is called *restricted* if some adjustments, which were technically possible, are not allowed. Examples are the restriction of diversion (not to allow a change of the destination when a vehicle is already traveling to a specified location) or of transshipment (not to allow the exchange of an already picked up load from one truck to another). The scope of adjustment is called *complete* if no future decision is restricted. However, in the most cases, the scope will be somehow *restricted*.

Simulation Techniques for Dynamic Algorithms

For evaluation of a new dynamic procedure's performance, simulation runs with various test data sets are usually applied. The best simulation mode depends on the individual algorithmic concept. For subsequent explanation, the simulation speed s is defined, meaning that 1 *hour* of real-life operations is simulated in $1/s$ *hours* of computer time.

If the algorithm possesses an improvement component, it may be beneficial to run the simulation in "real time" ($s = 1$), allowing for the same number of potential improvement operations as in reality. Here, higher simulation speeds ($s > 1$, with shorter simulation run time) result on average in an overall worse solution quality. Examples of real time simulation can be found in Shieh and May (1998), Gendreau et al. (2006) and Chen and Xu (2006). For a detailed investigation of simulation speed effects, see Section 5.5.

If there is no improvement component available, it is not necessary to choose the time consuming "real time" simulation (the time between two subsequent events would be wasted). Instead, it is sufficient to run an "event driven" simulation. In this case, it is not possible to give an exact value of simulation speed. It is only possible to estimate the number of events, multiply it with the average calculation time per event and set it into relation to the total simulated real time. Examples of event driven simulation can be found in Bent and van Hentenryck (2004), Fleischmann et al. (2004) and Tang and Hu (2005).

Performance Analysis for dynamic algorithms

The performance of a dynamic heuristic on a given dynamic test instance can be evaluated in five possible ways:

- by analytical derivation of the heuristic's worst case performance, compared to the optimal solution, obtained for the corresponding *static* instance,
- by comparison with the *optimal* solution obtained for the corresponding *static* instance,
- by comparison with a *heuristic* solution obtained for the corresponding *static* instance,
- by comparison with solutions achieved by other well-established dynamic heuristics for the original *dynamic* instance, and
- by comparison with *manually* achieved solutions of human planners, in case of a *dynamic* real-life instance.

The value of the **first case** is more of a theoretical nature; many researchers with a mathematical orientation focus on it under the name "competitive analysis". The notation "dynamic vs. static" is substituted here by "online vs. offline".

Comparing an online algorithm to an optimal offline algorithm was first suggested by Sleator et al. (1985) and the term "competive analysis" was coined by Karlin et al. (1988). An online algorithm is called *c-competitive* if the objective function value of the solution produced on any input sequence is **at most** c times that of an optimal offline algorithm on the same input. With the optimal offline algorithm having complete knowledge of the whole input sequence.

"Competitive analysis of online algorithms can be imagined as a game between an online player and a malicious offline adversary. The online player uses an online algorithm to process an input which is generated by the adversary. If the adversary knows the strategy of the online player, he can construct a request sequence which maximizes the ratio between the player's cost and the optimal off-line cost." (cp. Krumke, 2001)

Under very specific assumptions, competitive ratios for the dynamic TSP and the dynamic Traveling Repairman Problem (TRP) are e.g. derived by Jaillet and Wagner (2006). However, for many algorithms directed at solving practical applications such a *worst case* estimation is neither realizable nor very useful. Due to the NP-hardness of all problems, being extensions to the TSP, optimal solutions can usually not be calculated. In addition, the "worst case" scenario may not happen in practice. Information about an algorithm's performance should rather include average performance plus variability measures.

The **second option** can be seen as a simplified modification of the first, without conducting an analytical "worst case" analysis. The performance of some dynamic results is assessed by direct comparison with the optimal static solution. Again, the calculation of such a static optimal solution is usually not possible.

The **third option** for evaluating a dynamic algorithm, which is more practicable than its predecessors, is explained by Mitrovic-Minic et al. (2004). The authors define a "value of information $V(H)$ under heuristic H", which measures the possible gain in solving a dynamic problem ex post heuristically, if all information is known:

$$V(H) = \frac{\dot{x}^H - x^H}{\dot{x}^H}$$

with $\dot{x}^H$ being the best solution of the dynamic instance under heuristic H and x^H being the best solution of the static instance I under heuristic H.

Here, the same heuristic H is used explicitly for calculations on dynamic and static test data, which is probably not the most reasonable decision. According to Psaraftis (1988) and to former explanations, the configuration of dynamic and static algorithms necessitates differences. Here, the calculated value of $V(H)$ will always depend on "how well the corresponding static instance can be solved with a dynamic algorithm which is not supposed to solve such a problem." Therefore, an additional option would be the use of two heuristics: a dynamic one for the dynamic instance and a static one for the corresponding static instance.

The **fourth option** is based on the comparison with another dynamic heuristic that has been proved to produce competitive results for various dynamic data sets. Such an approach is realizable at justifiable expenses and is supposed to yield meaningful results.

The **fifth option** compares the results of a dynamic algorithm with the manually performed planning by human dispatchers. Even if overall manual solution quality can hardly be evaluated, it allows for assessment of a dynamic algorithm's relative solution quality. Therefore, practical usability of this performance measure is also quite high.

2.2 Relevance and Classification of Dynamic Standard Problems

Dynamic transportation problems can be found in several real-life applications where planning decisions have to be made subject to an environment of changing information. This section characterizes the most important real-life planning scenarios and connects these to the underlying theoretical problem definitions.

To classify real-life applications it is reasonable to differentiate between **depot bound** and **depot free** dynamic transportation problems. In the depot bound case, some good has to be transported to or away from a depot. In the depot free case, a depot is not involved. As a second distinctive feature, the application's geographical extension is used. If transportation tasks primarily cover limited operational areas, e.g. "urban areas" with an approximate maximum radius of 50km, we denote the application as **local area**; other applications without geographical limitation are described as **wide area**.

Classification: Depot bound/Depot free and Local Area/Wide Area

A typical representative of the **depot bound** group is the *VRP*, where each vehicle starts and finishes its tour at a prespecified depot, executing a Delivery tour (transportation of goods from the depot to the customers) or a Pickup tour (transportation of goods from the customers to the depot). Since maximum vehicle traveling distance per planning interval is limited, these problems usually occur in **local area** environments. Especially VRPs including Pickup tasks tend to have a dynamic component. This is due to the fact that additional Pickup tasks can be added to a vehicle's tour in the short term just by checking some feasibility constraints, like vehicle capacity or maximum tour duration, once the vehicle is already on its way.

VRP Delivery tours, however, require all Delivery objects to be loaded to the vehicle before the tour starts. If it is decided to add additional Delivery objects to the tour anyway, the vehicle would have to re-visit the depot in order to pick up the additionally needed objects. In practice, Courier services (like UPS or DHL) use the first half of the day to perform static Delivery tasks from the depot to the customers, afterwards (when the vehicle is empty), new (to some extent dynamic) objects are picked up and brought to the depot for further processing. The Vehicle Routing Problem for dynamic Delivery tasks is only conceivable in some special cases, like homogeneous good distribution (e.g. "oil and liquid gases", "beverage cases") and distribution of small products, where a large amount/stock of distributable products can be anticipatorily carried on a vehicle.

The group of **depot free** transportation tasks contains the dynamic *TRP*, where a technician with re-usable repair equipment is sent to dynamically occurring customers. Since just an immaterial "service" is provided to the customers (no transport to or from the depot), new customers can be flexibly included in the repairman's tour. The TRP can be considered as a local area problem.

The depot free group also contains the *Pickup and Delivery Problem (PDP)*, where objects or people have to be immediately transported from a Pickup to a Delivery location, without visiting a depot. We speak of a *Multi Load Pickup and Delivery Problem (MLPDP)* if consolidation of several objects is allowed. Otherwise, if objects have to be transported separately, we denote the problem *Single Load Pickup and Delivery Problem (SLPDP)*. Representative "Pickup and Delivery" applications can be found in local and wide area environments:

In the ***local area*** context, *Taxi or Dial-A-Ride services* deal with dynamic transportation requests of people who wish to be transported from location A to location B. The fast transportation of small parcels, for example between two urban area companies, is summarized in the concept of *Express Mail Delivery Services*. While transportation of passengers is associated with strict capacity constraints, in Express Mail Delivery applications, vehicle capacity can be neglected due to the small size of the parcels. Requests in both applications come along with tight time window constraints. In addition, peoples' transportation usually requires compliance with a maximum ride time duration.

In ***wide area*** environments, especially *occasional transportation* (also referred to as "tramp transportation"), possesses a dynamic component (cp. Section 1.1). Orders, mostly of Single Load type, have to be dynamically assigned to a fleet of moving vehicles, which is spread over the operational area (for example: Europe), producing a transportation schedule with minimal cost. In contrast to local area dynamic problems, the *reaction time* between occurrence of the request and the Pickup time window is usually longer. This is also true for the width of the time windows.

Table 2.1 summarizes the most important dynamic planning problems.

	depot bound	**depot free**
local area	VRP (Pickup Tour for Courier Services)	TRP, MLPDP (Taxi/Dial-A-Ride, Express Mail Delivery)
wide area	—	SLPDP (Occasional long-haul transportation)

Table 2.1: Classification of dynamic real-life problems

Other classifications

Another classification option for dynamic real-life problems was proposed by Gendreau and Potvin (1998). They distinguish between planning problems with "routing" (the need to sequence requests within planned routes) and "no routing". The second criterion is again the problem's geographical extension.

The first category "local area/routing" contains *Courier and Dial-A-Ride services* (see Table 2.2). In addition to our original classification scheme, *emergency services*, like ambulance and police, are also now included (second category: "local area/no routing"). These type of problems can be considered as dynamic transportation tasks, nevertheless their objectives differ. The main challenge is the appropriate re-positioning of a vehicle, once service is completed, in order to reach future requests in a preferably short time. Obviously, "routing" is not an important aspect for this type of problem.

	routing	**no routing**
local area	Courier Services, Dial-A-Ride	Emergency Services
wide area	Less-than-truckload trucking	Truckload trucking

Table 2.2: Classification by Gendreau and Potvin (1998)

More critically, the categorization of Full Truckload wide area transportation into the "no routing" group needs to be questioned. In our real-life problem (cp. Section 5.2), which is derived from a big German freight forwarding company, routing is a **necessary** task. A vehicle's tentative planning schedule usually includes more than one subsequent "routed" request. This differing interpretation may be induced by differences of long-haul transportation tasks in the US and Europe. Due to more agglomeration of economical centers and due to generally smaller distances, the average long-haul travel time in Europe tends to be shorter, allowing more than one transportation task to be scheduled.

Finally, the category "wide area/routing" is filled with the term *Less-than-truckload trucking*, indicating load consolidation for wide area occasional transportation. This is a very rare case to our knowledge. Even very large long-haul trucking companies who perform occasional transportation do not (usually) find requests suitable for consolidation: either the capacity does not allow for consolidation, the time windows are completely different, or suitable requests simply do not occur within geographical proximity. Therefore, as indicated in Section 1.1, less-than-truckload requests that have to be transported in wide area environments are usually fed into the driving routes of an existing – medium term – line operation schedule. This, however, veers away from dynamic short-term planning.

Since many real-life applications fall into the group "local area/routing", the authors perform a sub-classification of this category into the groups "capacitated" and "uncapacitated" as well as "many-to-many" and "one-to-many" (see Table 2.3). A vehicle is "capacitated" if the number of goods that can be loaded onto the vehicle is subject to limitations. From a physical standpoint, each vehicle is somehow "capacitated". However, if the goods are quite small and if there is not the chance of overloading the vehicle during a tour, we denote the associated problem as "uncapacitated". A request is of the type "many-to-many" if two locations (Pickup and Delivery, both different from the depot) are involved. Otherwise, if only a single location (Pickup or Delivery) is involved we speak of "one-to-many".

	many-to-many	**one-to-many**
capacitated	Dial-A-Ride	Feeder systems
uncapacitated	Express Mail Delivery	Courier and Repair Services

Table 2.3: Sub-classification of local-area/routing

For the "many-to-many" case, Gendreau and Potvin (1998) mention *Dial-A-Ride Services* and *Express-Mail-Delivery* with the attributes "capacitated" and "uncapaciated", respectively. The category "one-to-many" includes so-called *Feeder systems* in the "capacitated" case and *Courier and Repair services* in the "uncapaciated" case. A Feeder system is a Dial-A-Ride system with a specific target location, for example an airport or a train station, thus having just one Delivery location. Courier services correspond to the previously mentioned VRP tours; however, in the classification on-hand, there is no accentuation on Pickup tasks.

In the following section, two specific real-life planning scenarios are chosen for further investigation.

2.3 Investigated Problem Settings

This section gives a detailed description of the problems that are investigated in this work: the **dynamic local area (capacitated) MLPDPTW** in Chapter 4 and an extended real-life version of the **dynamic wide area SLPDPTW** in Chapter 5. According to the previous discussion of standard problems, both dynamic problem definitions cover relevant practical applications: Dial-A-Ride services and International Truck Transportation with occasional transportation tasks, respectively.

In the following, a catalog of selected attributes is presented for both problems, especially concerning the underlying *network* structure, the *planning horizon*, attributes of *orders*, *vehicles* and *tours*, as well as the *objective* function (cp. Stumpf, 1998). Tables 2.4 and 2.5 include the attributes of both problem settings.

Major differences occur in the *length of the planning horizon.* The local area planning horizon only covers a 10-hour interval; the planning horizon for the wide area problem, however, has been chosen as 5 weeks. While 10 hours are sufficient to consider a typical working day in the Dial-A-Ride context, a longer horizon is needed in wide area applications, since typical requests come along with a transportation distance that cannot be handled within a single day. In addition, the longer planning horizon allows for the inclusion of time restrictions, like EC social regulations.

Even though both problems cover a Pickup and Delivery problem, there are some differences in the *order characteristic.* The first setting allows for load consolidation, due to "less-than-truckload" order size. In the second scenario, all orders are of type "full truckload", hence there is no option of load consolidation. Nevertheless, some of the second scenario orders may possess several Pickup and/or Delivery locations that have to be processed in a fixed "inner order" sequence.

In the first case, a *homogeneous vehicle fleet* is considered, starting and finishing its tours at a central depot. In the second case, the fleet consists of *heterogeneous vehicles* ("vehicle types"), each with a specific starting position and a specific time of availability. Those vehicles do not have to return to their initial starting position or to any depot at the end of the planning horizon.

Arbitrary order-to-vehicle assignment is only allowed in the first scenario, while in the second scenario, the assignment of an open order requires a vehicle of appropriate type *(restricted order-to-vehicle assignment).*

network	
characteristic	- coordinate network - Euclidean distance
travel times	- constant

planning horizon	
	- rolling horizon - 10 hours + optional overtime

orders	
characteristic	- Pickup and Delivery ("depot free")
location in network	- node
transportation object	- goods
divisibility	- not divisible
time windows	- two-sided time window for Pickup - two-sided time window for Delivery - lower limit (EPT, EDT) = hard constraint - upper limit (LPT, LDT) = soft constraint
sequence of orders	- arbitrary
availability of data	- dynamic order arrival - initial amount of static orders (subject to variation)
size of order	**- less than truckload**
acceptance/rejection	- no rejection, transportation of all orders
geographical extension	**- local area**
transshipment	- no transshipment
number of orders	- 1000
frequency of orders	- singular

vehicles	
number	- 50 (subject to variation) - limited
structure of vehicle fleet	**- homogeneous**
vehicle ownership	- own vehicles
initial location	**- central depot**
applicability	- single day tours - multiple usage
time restrictions	- earliest availability time = hard constraint **- maximum duration** = soft constraint
capacity restrictions	- yes
crew	- one driver mode
driver-to-vehicle assignment	- fixed
order-to-vehicle compatibility	**- no restrictions**

tours	
standard tours	- no
shape of tours	**- start at central depot** **- closed,** vehicles have to return to depot
constraints	- no sequence constraints

objectives	
	minimize: weighted sum of travel time, delay, waiting time, overtime

Table 2.4: Characteristics of the dyn. local area MLPDPTW investigated in Chapter 4

network	
characteristic	- coordinate network - Euclidean distance
travel times	- constant

planning horizon	
	- rolling horizon - 5 weeks

orders	
characteristic	- Pickup and Delivery ("depot free"), with **various Pickup and/or Delivery locations per order and fixed inner order sequence**
location in network	- node
transportation object	- goods, requiring specific vehicle type
divisibility	- not divisible
time windows	- two-sided time window for Pickup - two-sided time window for Delivery - lower limit (EPT, EDT) = hard constraint - upper limit (LPT, LDT) = soft constraint
sequence of orders	- arbitrary
availability of data	- dynamic order arrival - small initial amount of static orders (6.3%)
size of order	**- full truckload**
acceptance/rejection	- no rejection, transportation of all orders
geographical extension	**- wide area**
transshipment	- no transshipment
number of orders	- 14025
frequency of orders	- singular

vehicles	
number	- 953 - limited
structure of vehicle fleet	**- heterogeneous, 5 different vehicle types**
vehicle ownership	- own vehicles
initial location	**- depot free**
applicability	- multi-day tours - multiple usage
time restrictions	**- EC social regulations** = hard constraint **- working time regulations** = hard constraint **- general driving bans** = hard constraint - earliest availability time = hard constraint
crew	**- mixed** (one driver mode, team driver mode)
driver-to-vehicle assignment	- fixed
order compatibility	**- restricted vehicle-to-order assignment**

tours	
standard tours	- no
shape of tours	**- individual vehicle starting position** **- open,** no return to starting position
constraints	- fixed sequence due to order specification - vehicle based time restrictions

objectives	
	minimize: weighted sum of empty travel time, delay, waiting time

Table 2.5: Characteristics of the dyn. wide area SLPDPTW investigated in Chapter 5

The problems also differ in *time restrictions*. The first setting only covers a maximum tour duration (soft constraint), while the second setting covers a set of more sophisticated restrictions, like EC social regulations, working time regulations or general traffic bans (all being treated as hard constraint).

In both cases, the *objective* is to minimize a weighted cost function. However, there are minor differences in the chosen components: in the first case, the cost function includes "total travel time", while in the second case, only "empty travel time" is considered. Due to the maximum tour duration soft constraint, an additional penalty term for "overtime" costs is added in the first scenario.

Chapter 3

Literature Review

This chapter includes a detailed literature review of publications on Dynamic Fleet Management. At the beginning, some general statistics on the surveyed literature are given (Section 3.1). This is followed by some exemplary publications showing the variety of dynamic applications in real-life (Section 3.2). Afterwards, algorithm orientated papers are presented categorized into three groups depending on the knowledge of the future (Section 3.3). The remaining sections review the most popular dynamic test instances in the literature (Section 3.4) and outline the results of some papers that do not primarily focus on the algorithmic performance, but on the acceptance of dynamic planning applications in real-life (Section 3.5).

3.1 Statistical Analysis of the Surveyed Publications

Dynamic aspects in transportation have attracted increasing attention in the research community and in practice over the last years. A dynamic transportation problem was *considered first* by Wilson (Wilson et al., 1971; Wilson and Weissberg, 1976; Wilson and Colvin, 1977) at the Massachusetts Institute of Technology (MIT) in Boston. For a dynamic Dial-A-Ride Problem at the city of Rochester (USA), the authors develop an insertion heuristic, which, after the occurrence of a new order, evaluates all possible insertion positions in the existing tours. According to a special selection criterion, the new order is inserted at the best position.

After this early work, it took several years for *dynamic transportation* to become a popular field of research. Today, many publications are available, investigating dynamic real-life applications and proposing new efficient solution methods. Figure 3.1 shows the number of publications concerning “dynamics in transportation” over the *course of time*, based on the 64 sources cited in this survey.[4] In the mid-nineties, the number of publications started to increase, reaching a peak of 12 publications in 2004.[5] Afterwards, a medium level was maintained but with a decreasing trend.

---

[4] Other surveys with classification schemes and literature different from this survey have been published, for example, by Psaraftis (1995), Ghiani et al. (2003), Cordeau et al. (2007), and Larsen et al. (2008).

[5] This finding may be attributed to a special issue on “Real-Time Fleet Management” in Transportation Science in 2004

Steffen Schorpp, *Dynamic Fleet Management for International Truck Transportation,*
DOI 10.1007/978-3-8349-6675-9_3,

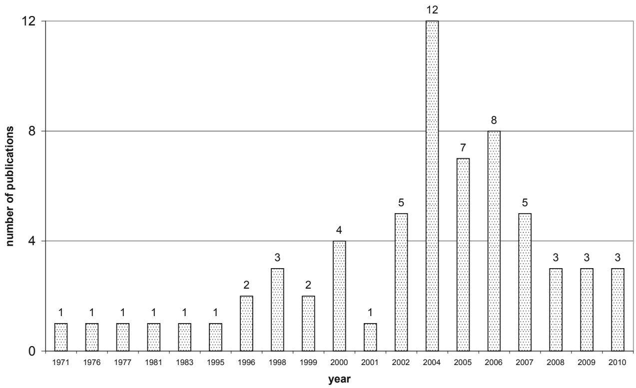


Figure 3.1: Publications on dynamic transportation over the course of time

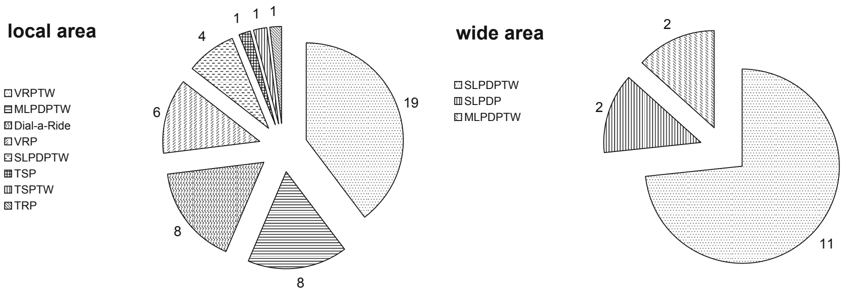


Figure 3.2: Standard problems considered in the selected dynamic publications

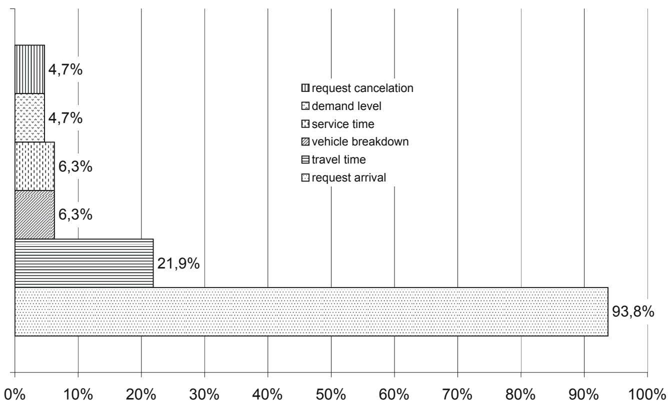


Figure 3.3: Sources of dynamism considered in dynamic publications (in %)

It is also interesting to analyze which *basic theoretical problems* the surveyed publications have dealt with (cp. Figure 3.2). In total, 48 papers considered local area applications, 15 considered a wide area environment, and one paper could not be attributed. Within the local area group, 27 papers are depot bound (VRP, VRPTW, TSP, TSPTW) and 21 papers are depot free (TRP, SLPDPTW, MLPDPTW, Dial-A-Ride). All wide area applications are depot free (SLPDPTW, SLPDP, MLPDPTW).

When evaluating *sources of dynamism* (cp. Figure 3.3), it can be observed that nearly all publications (93.8%) consider the dynamic arrival of requests. Also quite popular are dynamic travel times (21.9%), while vehicle breakdown, service time (each 6.3%), dynamic demand levels, and request cancelation (each 4.7%) are only considered by a few authors.

Before algorithmic solution procedures come to the fore in Section 3.3, several publications that focus specifically on dynamic practical applications will be presented.

3.2 Practical Applications

The following publications primarily focus on dynamic applications in real-life. They give a detailed description of the associated practical planning problems and present solution approaches that have actually been implemented. However, additional real-life applications with distinctive focus on algorithmic solution concepts can also be found in Section 3.3. This section's purpose is to outline the variety of dynamic real-life applications, which, for example, can be seen at the different objects or services provided by vehicles.

The first five papers (see Table 3.1) consider various objects (from petroleum to human patients) that have to be transported. A further paper deals with vehicles providing roadside assistance service, while, in the last application, customer service is constituted by the use of a specific recreational vehicle itself.

authors	**investigated real-life topic**
Brown and Graves (1981)	dispatching of petroleum tank trucks
Bell et al. (1983)	distribution of industrial gases in a VMI environment
Savelsbergh and Sol (1998)	truck dispatching at shipping company
Magalhaes and Sousa (2006)	distribution of pharmaceutical products
Beaudry et al. (2010)	transportation of patients in a hospital
Krumke et al. (2002)	dispatching of mobile roadside assistance units (ADAC)
Ernst et al. (2007)	dispatching of recreational vehicles

Table 3.1: Variety of dynamic real-life applications

Brown and Graves (1981) consider *fleet dispatching of petroleum tank trucks at "Chevron Corp."*. Starting from 80 US terminals, 2,600 loads per day have to be scheduled, delivering motor gasoline, weed oils and jet fuels. Vehicles perform a series of successive "single load" pendular tours between depot and customers. A vehicle consists of several compartments that have to be filled with different types of gasoline. The order quantity is assumed to be static, while new orders occur dynamically. The objective is the minimization of transportation costs and an equitable workload distribution.

The planning problem has to be solved for each depot in a rolling horizon framework. At first, authors model an *integer linear program*, but find out that it cannot be solved exactly within a reasonable time. This is because there is just a single central computer that has to handle a subproblem query from one of the 80 terminals within a fraction of a second.

Hence, a *heuristic* is proposed, which solves a sequence of embedded network flow problems and successively fixes order to truck assignments. Subsequently, the solution is *improved by load exchanges* between two trucks and *Best Re-Insertion*. The new dispatch module produces excellent solution quality and a reduction of transportation costs by about three percent. Additionally, the new planning approach achieves extremely uniform distributions of workload among vehicles.

Bell et al. (1983) report on a *dynamic distribution problem (capacitated VRPTW) at "Airproducts and Chemicals Inc."*, which sells industrial gases (oxygen, nitrogen, argon and carbon monoxide) from 23 depots to 3,500 customers in the US. Ten to thirty vehicles are assigned to each depot. The inventory of storage tanks, located at the customer locations, is monitored by the distributor (Vendor Managed Inventory) and must be maintained above a specified safety stock level.

The customer's demand is dynamic. The only indication for future demand levels is an estimation based on a historical 15-day horizon. Customers with high variability can additionally be phoned to detect their exact inventory levels. On the basis of this demand information, vehicle routes, schedules and quantities for Delivery are planned for each depot. Nevertheless, demands may deviate from the estimated amounts. In addition, unaccounted emergency orders, which have to be served immediately, may occur during the day. The authors model the problem as a *mixed integer problem* and develop a *Lagrange relaxation based algorithm*, which is solved in a rolling horizon manner (with a two to five days horizon). As a result, the authors observe savings of up to $1.72 million annually when replacing the current manual planning with the new planning system.

Beaudry et al. (2010) investigate *dynamic transportation of patients between health care units and service areas in a large hospital in Southern Germany (dynamic Dial-A-Ride)*. The hospital complex consists of 100 buildings and a road network of 15 km. A heterogeneous fleet of 11 ambulances, each carrying special equipment, is responsible for picking up and delivering people in given time windows. Some people require individual transportation, other people can be combined. A vehicle may carry different load combinations: one bed, one wheelchair, one seated person at a time, or up to three wheelchairs. Some transportation tasks require the vehicle to go back to the depot for disinfection afterwards. Other transportation tasks require the Pickup of an accompanying person, who needs to be picked up before the patient and perhaps needs to be brought back after the completion of the transport.

Ninety-six percent of the requests are called in dynamically. Further possible dynamic events are cancelations and updates of requests, as well as late arrivals and vehicle breakdown. The objective function prioritizes patient convenience over both travel time and prevention of early arrival. The authors develop a rolling horizon *Tabu Search metaheuristic* based on the *neighborhoods best re-insertion* and *intraroute re-arrangement*. Tests with a 20-day historical data horizon from the hospital reveal significant reductions in waiting times for patients and a reduction of the number of vehicles.

Krumke et al. (2002) examine a *dynamic service vehicle dispatching problem at the German Automobile Association (ADAC)*, which provides roadside assistance to people whose car has broken during their journey. Service vehicles possess individual capabilities (spare parts, repair kit), cost parameters and home location. In addition to ADAC owned vehicles, it is also possible to access subcontractor's vehicles. The problem can be considered as a multi depot Vehicle Routing Problem, with dynamic requests (100%) and dynamic service times. If a "customer" calls in, the objective is to guarantee service within a short period of waiting time (lateness cost), while keeping operational costs for the service vehicles (driving cost, overtime cost) as low as possible.

The authors propose a *Column Generation approach*, which is applied in a heuristic way. By solving linear subproblems, dual prices are obtained and are used to generate feasible tours with reduced cost. For a test set with 770 events and 200 mobile units, solution quality is within 5% from optimum within 15 seconds and within 2% from static optimum after one minute.

Savelsbergh and Sol (1998) describe a *dynamic planning problem at "Van Gend and Loos BV", the largest road transportation provider in Benelux.* The paper is focused on direct transportation (no consolidation of orders at a depot or hub), which is carried out for order sizes ranging from four pallets to a full truckload. The problem can be considered as an MLPDPTW, with the special characteristic that one request can have several Delivery locations in a predetermined order. Van Gend and Loos BV exclusively use rented vehicles (on average 100) for direct transportation: 50 are rented permanently and the remaining vehicles are rented on a daily basis. The number of vehicles that are rented on a daily basis has to be specified at the beginning of a working day.

The primary planning objective is to minimize the number of vehicles. The secondary objective is the minimization of total traveled distance. Especially the estimation of the right number of vehicles is difficult, because just 40% of a day's orders are known in advance, while 60% arrive during the day of execution. The authors develop a *Column Generation* based solution approach. Since fast reaction times ($<$ 5min) have to be ensured, the underlying pricing problem is solved by an approximization algorithm. Encouraging results of the new solution approach are reported, leading to reductions in total costs ranging from 3.7% to 4.7% a week. On the other hand, the number of vehicles used is slightly higher than before.

Magalhaes and Sousa (2006) deal with a *dynamic application at Cofanor, a distributor of pharmaceutical products, operating in Portugal.* As pharmacies organize stock with Just-In-Time policies, they tend to place several orders with rather small quantities during a single day. These orders are digitally transferred to Cofanor, where a human operator confirms them. Afterwards, a picking process in the distributor's warehouse is started, until the orders (on average 400 per day) can be distributed by vans. The objective is a quick response to customer demands (short lead-time) and keeping Delivery costs (traveled distance) low.

The authors propose a four-phase heuristic (for a capacitated VRP). First, the orders are *clustered by increasing angle*. Then, a route of orders is generated in each cluster using *Best Insertion*. Thirdly, the urgency of all routes is checked and only routes with an "urgent" order are released to phase four (an order is "urgent" if the time between order placement and expected Delivery is greater than a predefined threshold). In phase four, the chosen routes are improved with 2-opt and finally released for execution. The postponement of tours in phase three is used with the intention of receiving further compatible orders. Results of the new heuristic are compared to the results of manual planning: average lead time to pharmacies is reduced by 8.1%, however, traveled distance is increased by 1.9%.

Ernst et al. (2007) report on a *dynamic planning problem at Tourism Holdings Limited, a New Zealand-based company that operates a fleet of more than 4000 recreational vehicles (motor homes and camper vans)* at 10 locations in Australia and 4 locations in New Zealand. The vehicle fleet consists of 50 distinct vehicle types, e.g. varying in the number of berths, number of doors and power engine. The dispatching of vehicles is performed on a 200-day "active scheduling horizon" and can be described as an SLPDPTW. When a customer calls in, an *acceptance/rejection-decision* is made within five seconds. If the request cannot be accepted, alternatives have to be suggested, e.g. different dates or similar products. A further planning task is the adjustment of the plan to dynamic events, like late return, vehicle breakdown, etc.

The authors employ *two planning levels*: the objective of the *first level* is to maximize the number of accepted bookings, while, in the *second level*, the cost for handling the accepted bookings is minimized. Operation costs consist particularly of empty relocation, free upgrade to a higher valued vehicle, accelerated cleaning to hold appointed allocation time, etc. The dynamic first level problem is solved with a linear assignment algorithm. For the second level, a *relaxed linear programming formulation* is solved with *ILOG*. Afterwards, the relaxed conditions are heuristically re-incorporated. Idle time between dynamic events is used to improve the current plan. The application of the new system resulted in 2% savings in operating costs. Simultaneously, human planners won time to intensify their efforts handling exceptions.

In most cases, practical applications do not perfectly coincide with one of the standard problems. Because many additional requirements often have to be accounted for, this section's primary function has been to show the widespread appearance of dynamic problems in real-life. The following section will cover algorithmic procedures (usually related to standard problems), which provide a pool of generic concepts that can be adapted to more specific practical applications.

3.3 Algorithmic Solution Concepts

The reviewed publications within this section can be divided into three groups. The first two groups do not have any knowledge of the future and therefore only perform "myopic" planning. In contrast to the first group (Section 3.3.1), the second group (Section 3.3.2) anyhow tries to anticipate the future. Stochastic information about the future is available only to third group publications (Section 3.3.3), which make explicit use of it with different concepts.

3.3.1 Dynamic Approaches without Knowledge of the Future

There are generally many possible ways of grouping dynamic myopic publications: by *investigated standard problems*, by *sources of dynamism*, by *degree of dynamism*, by *geographical area*, by *associated groups of authors*, etc. In the following, however, the *main algorithmic solution concepts* have been chosen for classification:

- **local search approaches** (3.3.1.1),
- **metaheuristics**, guiding the local search out of local optima (3.3.1.2),
- **heuristic applications of exact procedures** (3.3.1.3),
- **rule-based approaches** (3.3.1.4), and
- **multi-agent systems** (3.3.1.5).

At the beginning of each of the following subsections, a short summary of the selected publications' properties is given in the form of a table (see Table 3.2 for an example). This table includes the dynamic aspects considered in the associated publication.

An "X" in the first column indicates dynamically occurring requests. If available, the degree of dynamism is given in subsequent brackets. The second column reports on further sources of dynamism (e.g. travel time). Afterwards, the considered standard problem is specified: capacitated/uncapacitated (column four), actual problem (column five), time window characteristic hard/soft - if time windows are available at all (column six), and geographical extension (column seven). An "X" in column eight indicates that en route diversion is allowed, and column nine gives information about the employed dynamic test data sets.

	dynamic aspects		capaci-	problem	TW	area	en route	dynamic
	orders	other	tated		constr.		diversion	test data
Shieh and May, 1998	X (50%)		cap.	VRPTW	hard	local		Solomon, 1987
Du et al., 2005	X (100%)		cap.	VRPTW	soft	local		self-generated
Tang and Hu, 2005	X (50%)		uncap.	VRPTW	hard	local		Solomon, 1987
Potvin et al., 2006	X (50%)	travel time	uncap.	VRPTW	soft	local		Solomon, 1987
Chen et al., 2006	X (78%)	travel time	cap.	VRPTW	hard	local		Solomon, 1987 + real-life
Branchini et al., 2009	X (60%)		cap.	VRPTW	soft	local	X	self-generated

Table 3.2: Local search approaches

3.3.1.1 Local Search Approaches

The local search approaches that are subsequently presented consist mainly of two parts. The first part is applied to construct a feasible solution (e.g. with Best Insertion). The second part uses classical techniques (like Re-Insertion or 2-opt) to improve this initial solution. Measures to escape from local optima are not applied.

Table 3.2 summarizes the properties of five selected papers. All of them consider the local area VRPTW with dynamically occuring requests. The degree of dynamism varies between 50% and 100%. In addition, dynamic travel times are included by two publications as a second source of dynamism. The latest publication by Branchini et al. (2009) allows for en route diversion.

Shieh and May (1998) consider a capacitated VRP with hard time windows, where up to 50% of customers occur dynamically. The objective is the minimization of traveled distance. Orders may be rejected. The authors propose a heuristic that uses *Best Insertion* for constructing a feasible solution, *followed by intra- and interroute improvement with OR-opt and 2-opt.* The improvement part is run continuously between the occurrence of two requests.

For testing purposes, the static VRPTW instances of Solomon (1987) are extended by a new column with random Call-In times. Analyses are carried out, comparing the results of the dynamic approach with the best known solutions of the static Solomon instances. The authors run their *simulations in real time* and report an increase in the number of used vehicles by factors of 1.12 to 2.14 and an increase in traveled distance ranging from 7% to 25%. It has to be mentioned that for some problem sets, not all requests could be serviced due to possible late Call-In and hard time window constraints.

Du et al. (2005) regard a capacitated VRP with soft time windows, in which up to 100% of customers occur dynamically. The objective function consists of two levels. The first level goal is to minimize the total distance traveled. When no feasible insertion position can be found, the goal is to minimize delay. The authors propose a heuristic with construction and improvement parts. In total, four construction methods are presented, partially depending on geographical order clustering, similar to the *"sweep algorithm"*:

- find the cluster to which the request belongs to and append the new order at the end of the associated vehicle's queue;
- assign the order to the vehicle with the smallest distance between the last order in the vehicle's queue and the new order's location;
- apply *Best Insertion* with regard to all vehicles;
- find the cluster the order belongs to and apply *Best Insertion* to the associated vehicle queue. If necessary (due to capacity constraints), take a new vehicle from depot; if necessary (due to absence of additional vehicles at the depot), check insertion cost for vehicles in close regions.

The improvement component consists of interroute changes with *Best Re-Insertion and 2-Exchange* (each of two routes is cut into two segments, then the second segments are exchanged) and intraroute changes with *Or-opt and 2-Swap* (exchange of two nodes within a vehicles route). Improvement is executed as pure local descent, i.e. changes are only accepted when an improvement of the objective function is found. Tests are conducted with self-generated data sets, showing best results for construction with Best Insertion followed by improvement with Best Re-Insertion and OR-opt.

Tang and Hu (2005) deal with an uncapacitated VRP with hard time windows, in which up to 50% of the customers occur dynamically. The main goal is to maximize the number of serviced customers. This is achieved by accepting as many customers as feasible. Further goals are the minimization of customer waiting time (defined as the time gap between Call-In and start of service) and the minimization of traveled distance, with higher priority being attributed to the reduction of customer waiting time.

The authors propose a rolling horizon based approach, which is triggered by the occurrence of new orders. For the acceptance decision, not only *Best Insertion* is used: When the first attempt does not result in a feasible plan, additional adjustments with *Best Re-Insertion and OR-opt* are applied in order to create a feasible insertion position for the new order. The order is only rejected if all these attempts fail. Subsequent improvement is carried out with a version of OR-opt, examining the relocation of three, two, or one consecutive nodes in the vehicle's current tour. Occasionally, a number of requests is extracted and re-inserted.

The Solomon (1987) instances are used as test data, with the extension of dynamic Call-In times and modified time window characteristics. For test instances with wide as well as narrow time windows, the authors report "dramatic benefits" with the new approach when compared to a benchmark procedure, based on Best Insertion and OR-opt (objective: minimization of travel time). General results show high quality solutions within a limited computing time.

Potvin et al. (2006) deal with an uncapacitated VRP with soft time windows, in which 50% of the customers occur dynamically. A special focus is placed on *travel times which are subject to several fluctuations*:

- Depending on the time period of the day, the average travel time is multiplied by prespecified coefficients ("long-term forecast"). This is an a-priori known information.
- The moment a vehicle starts traveling on a link, a short-term bias to the travel time coefficient is revealed ("short-term forecast"). This value is chosen according to a uniform random distribution in the interval $[-0.1, +0.1]$.
- The arrival time at a link's destination is furthermore distorted by unforeseen events that may occur along the travel leg. These are modeled as normally distributed perturbations with a mean of 0 and standard deviations ranging from 1 to 32. Here, only delays to the current schedule are considered, thus a negative value is simply reset to 0. Information about the extent of such variations is first known to the algorithm when the vehicle finally arrives at the destination.

The general objective is to minimize an equally weighted sum of travel time, lateness and overtime. The authors propose a solution approach based on *Best Insertion and subsequent improvement with Cross-Exchange* ("two segments of routes are exchanged between two different routes by removing two arcs in each route and by appropriately reconnecting the two segments.") and *Intraroute Exchange*. The procedure is applied at the beginning and at the occurrence of the following events:

- the arrival of a new order,
- when the short-term forecast on travel time is introduced at vehicle departure, and

- when the arrival time at a location is delayed by a "tolerance time limit".

In the last case, the order is reassigned to another vehicle. If the original vehicle arrives before the new vehicle, the algorithm tries to cancel the re-assignment. If the new vehicle has not yet started traveling to the observed location, the associated node is simply removed from its schedule, otherwise the new vehicle reaches the location without serving it (no en route diversion).

As test data, the authors use dynamic extensions to the Solomon (1987) data sets. They observe that an increasing magnitude of dynamic leg travel time perturbations results in harder to solve problems. After considering different levels for the "tolerance time limit", a short waiting time shows the best performance. Events of small magnitude are caught, and reaction is only performed on events of larger magnitude.

Chen et al. (2006) investigate a capacitated VRP with hard time windows, in which up to 77% of orders occur dynamically. *Travel times are also subject to dynamic fluctuations*, modeled as distortions to (a-priori known) time-dependent travel times. The objective is the minimization of a weighted cost function, containing travel and waiting time. New orders may be rejected if a feasible insertion position into the current plan cannot be found. Rejection is also possible if fluctuations in travel time make it impossible to serve an order within its time windows.

The authors propose a heuristic approach which uses *Best Insertion* for route construction, *followed by an improvement routine with OR-opt.* Planning runs are triggered by the occurrence of new dynamic information and by execution of irreversible planning events (e.g. permanent order-to-vehicle assignment or when the vehicle starts traveling to a specific order).

Extended Solomon (1987) data and some real-life data from a logistics company located in Taiwan are used for testing purposes. Dynamic travel time fluctuations are modeled with two types of random variables. First, the interval lengths of time-dependent travel times are distorted, then the corresponding travel times. For reasons of comparison, the authors apply a solution approach that is not able to include dynamic updates in travel times. As expected, the solution approach that considers those changes in travel time significantly surpasses the benchmark procedure in performance. For the second data set, the new algorithm is benchmarked with the manual planning of human dispatchers: The results for a fleet of six vehicles show a reduction in total travel time from originally 875 minutes down to 832 minutes.

Branchini et al. (2009) consider a capacitated VRP with soft time windows, in which *up to 60% of customers occur dynamically.* The objective is the minimization of traveled distance. The authors propose a construction heuristic that uses the initial static requests to distribute the available vehicles equally across the whole service region, in order to accommodate future dynamic customers more easily.

After a request arrival has taken place, the new customers are included with *Best Insertion.* In addition, an improvement procedure is continuously run, investigating *2-opt, OR-opt and Cross-Exchange* neighborhoods. Depending on arrival intensity, these neighborhoods are dynamically reduced ("adaptive") in order to concentrate on high quality solutions. Hence, "long" arcs with small probabilities for improvement are neglected. The authors call their approach *granular search.* The approach further includes a basic "wait first" strategy, en route diversion, as well as vehicle re-positioning to strategic waiting places.

For testing purposes, three instances are generated with a personal data generator. The basic parameter settings are taken from real-life transportation companies. The authors apply several nine-hour *real time simulations* and compare their new approach with Best Insertion and Nearest Neighbor. As expected, significant profit gains were achieved when employing the new sophisticated approach. In addition, a 4% better objective value is reported when all information is made known to the algorithm a-priori.

3.3.1.2 Metaheuristics

The second group of dynamic approaches, the "Metaheuristics", can be seen as a *control level* above one (or several) basic local search procedures (improvement neighborhoods). Metaheuristics usually allow for temporary deterioration of the objective function value in order to escape from local optima. Five representatives are subsequently considered: *Tabu Search, Evolutionary Approaches, Variable Neighborhood Search, Ant Colony, and the concept of a Second Objective Function.* Table 3.3 summarizes the properties of the associated publications.

Tabu Search

In a Tabu Search (cp. Glover, 1989), the algorithm moves towards the best available solution, generated by its underlying neighborhood. This is possible as long as a solution is not stored within the "tabu list", which includes (usually for a given time horizon) already visited solutions. Since the algorithm is not allowed to choose such a tabu list solution, it is forced to explore regions of the search space that would otherwise be left unexplored. Temporary worsening of the objective function value is explicitly allowed in order to escape from local optima.

Five Tabu Search publications, all considering a local area geographical extension, are presented: two focusing on the uncapacitated VRPTW, two focusing on the Dial-A-Ride problem and one dealing with the uncapacitated MLPDPTW. An interesting aspect of the second paper by Ichoua et al. (2000) is the detailed investigation of the impact of "en route diversion".

Gendreau et al. (1999) deal with an uncapacitated VRP with soft time windows for a Courier service application, in which 50% of the customers occur dynamically. New orders may be rejected if they cannot be handled within a feasible solution, e.g. because of hard time window restrictions at the depot. The objective is to minimize a weighted cost function, including total distance traveled and time window violations.

The authors introduce a *parallel Tabu Search algorithm with Adaptive Memory*, based on Taillard et al. (1997). The Adaptive Memory, similar to the concept of Genetic Algorithms, contains a set of feasible solutions. These are generated in the beginning by

	dynamic aspects		capaci-	problem	TW	area	en route	dynamic
	orders	other	tated		constr.		diversion	test data
Tabu Search								
Gendreau et al., 1999	X (50%)		uncap.	VRPTW	soft	local		Solomon, 1987
Ichoua et al., 2000	X (75%)		uncap.	VRPTW	soft	local	X	Solomon, 1987
Attanasio et al., 2004	X (50%)		cap.	DARP	hard	local		Cordeau and Laporte, 2003
Fabri and Recht, 2006	X (100%)		cap.	DARP	hard	local		Caramia et al., 2002
Gendreau et al., 2006	X (100%)		uncap.	PDPTW	soft	local		self-generated
Evolutionary Approaches								
Haghani and Jung, 2005	X (55%)	travel time	cap.	VRPTW	soft	local		self-generated
Pankratz, 2005	X (100%)		cap.	MLPDPTW	hard	local		Solomon, 1987
Hanshar and Ombuki-B., 2007	X		cap.	VRP		local		Kilby et al., 1998
Cheung et al., 2008	X (16%)	travel time	cap.	MLPDPTW	hard	local		self-generated
Okhrin and Richter, 2008		travel time	cap.	VRPTW	hard	local		Solomon, 1987
Variable Neighborhood Search								
Angellelli et al., 2004	X		uncap.	VRPTW	hard	local	X	
Bock, 2010	X	several	cap.	MLPDPTW	soft	wide	X	self-generated
Ant Colony								
Montemanni et al., 2005	X		cap.	VRP		local		Kilby et al., 1998; real-life
Guntsch and Middendorf, 2002	X			TSP				TSPLIB
Second Objective Function								
Xiang et al., 2008	X	several	cap.	DARP	soft	local		self-generated

Table 3.3: Metaheuristics

a stochastic insertion heuristic and subsequent improvement with Tabu Search (neighborhood: Cross Exchange). Based on the solutions found in the Adaptive Memory, new solutions are repeatedly composed and improved with Tabu Search (cp. Figure 3.4).

Afterwards, the best new solutions are added to the Adaptive Memory. In a dynamic environment, the improvement procedure is run until a new event occurs. If the new event is the occurrence of a new request, the latter is inserted into each Adaptive Memory solution. If the event represents the end of service at a customer location, the driver's next destination is identified using the best solution stored in the Adaptive Memory. In order to keep solutions consistent, the other solutions are updated accordingly. Then, the overall improvement process is restarted.

The Solomon (1987) data sets, extended to dynamic aspects, are used for testing purposes. The new approach is benchmarked with several "easier" heuristics: simple successive in-

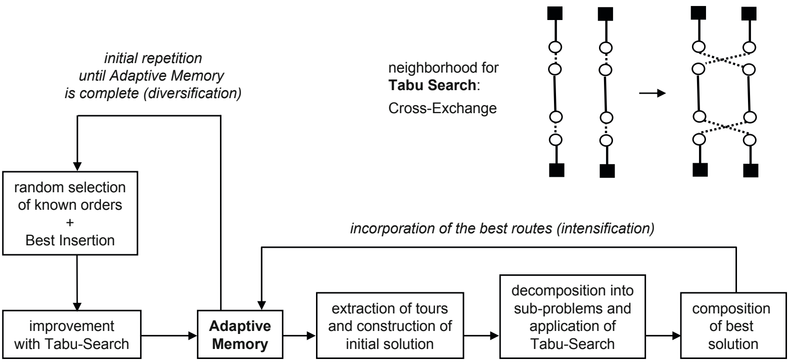


Figure 3.4: Tabu Search algorithm with Adaptive Memory

sertion, successive insertion and improvement with cross exchange, complete solution rebuild with insertion, rebuild and improvement with cross exchange, and parallel Tabu Search with stop at the first local optimum. As expected, results show best performance when the complete approach is run. In addition, the authors show that it is beneficial to optimize the planned routes between the occurrence of two events. With increasing computation time better average results are achieved. Finally, the *parallelization of the algorithm on up to 16 processors* is investigated, resulting in more customers being serviced and in a reduced sum of distance and lateness.

Ichoua et al. (2000) extend the scope of action for the problem and algorithm considered in Gendreau et al. (1999) by allowing **"en route diversion"**. *A driving vehicle may be directed away from its current destination in order to serve a request that has just occurred in the vicinity of its current position.*

The authors introduce a framework with variable anticipation horizon δt. Every time t the solution approach is restarted, decisions in the horizon $t + \delta t$ are "frozen", so that the current algorithm run may schedule planning updates soonest for time $t + \delta t$. During the frozen period, new requests may arrive, which have to be accepted or rejected (only if no feasible insertion position is available). For this purpose, a copy of the current plan is held ready, which is updated if a new best solution is found. In this situation, the authors do not consider the case of two similar orders arriving in the same frozen interval. Both could be accepted separately, but together they could render the plan infeasible.

In order to determine reasonable computation times, the authors suggest three variable rules for calculating the value of δt:

- δt is chosen in such a way that the solution procedure ends before any vehicle arrives at its current destination,
- δt is chosen to be proportional to a moving average of the last l inter-arrival times,
- δt is chosen to be the length of some time horizon X divided by the number of requests on the planned routes found within that time horizon.

In first tests, the third rule succeeded in reducing both the number of unserved customers, as well as total objective value.

The same test data sets as in Gendreau et al. (1999) are used, providing insights into potential gains by applying "en route diversion". Compared to the original heuristic, the number of unserved customers is indeed reduced by 16.8% up to 100%. In addition, the objective function value is decreased by 2% up to 4.3%, indicating substantial benefits through the exploitation of en route diversion.

Attanasio et al. (2004) deal with a capacitated Dial-A-Ride problem, where 50% of the requests occur dynamically. Orders possess one hard time window, which is either the Delivery time window (for outbound trips) or the Pickup time window (for inbound trips). In addition, a maximum ride time of 90 minutes has to be considered. New orders are accepted if they can be feasibly inserted into the current plan without violating any hard constraints. The objective is the minimization of traveled distance.

The authors develop a *Tabu Search approach*, which is a dynamic and parallelized extension to Cordeau and Laporte (2003). An initial solution is generated by relaxing several hard problem restrictions (capacity, maximal route duration, time windows, and user ride time constraints). Then, the Tabu Search, based on a *Best Re-Insertion neighborhood*, explores the solution space including infeasible solutions. After each iteration, the objective function cost parameters are adjusted, raising and decreasing the weight for restrictions that have been violated and complied with, respectively. With the help of a tabu list, solutions are penalized by a factor proportional to the frequency of the addition of its distinguishing attributes. If a new best solution is found, intraroute exchanges are performed.

As test data, the authors use 26 dynamic instances from real-life applications in Montreal/Canada and from a Danish company. A parallelization strategy is applied with an increasing number of processors ranging from one to eight. Results show the benefits of enhanced computing power via parallel computing. A performance statement on how well the proposed algorithm works in dynamic environments (e.g. by comparison with static data or with other dynamic approaches) is not explicitly mentioned.

Fabri and Recht (2006)[6] investigate a capacitated Dial-A-Ride problem with 100% dynamically occurring customers. When a request arrives, it is accepted if a feasible insertion position into the current plan (in compliance with hard time windows for Pickup and Delivery) is available. The objectives are the minimization of rejected orders and the minimization of traveled distance.

The authors employ a dynamic solution approach that extends the work of Caramia et al. (2002) by introducing explicit Delivery time windows and by allowing waiting times. When a new order occurs, all vehicles are successively inspected for a feasible insertion position. If such a position exists, the order is assigned to the cheapest vehicle.

The single vehicle subproblem is solved as follows. A network of possible status vectors is established, in which each order may have the status already delivered ("0"), already

[6] Based on the author's dissertation: Fabri (2008)

picked up ("1"), or waiting for Pickup ("2"). Two vertices are connected by an edge whenever the subsequent vector can be obtained from its predecessor by subtracting 1 to exactly one vector element. The source vertex is the vector $(2, 2, \ldots, 2)$, i.e. all accepted demands are waiting for Pickup, the sink vertex is the vector $(0, 0, \ldots, 0)$, i.e. all accepted demands have been delivered. The problem is to find a shortest path from the source to the sink, subject to time windows and capacity constraints. This task is performed with an *A* algorithm* that reduces computation time by using a lower bound approximization function to estimate the cost of the route from the current vertex to the sink. Between two events, the solution is improved with a Tabu Search algorithm based on the *neighborhoods Best Re-Insertion and* $(1,1)$*-Exchange.*

As test data, the authors use adapted data from Caramia et al. (2002). The "stretch factor", which originally implies the maximum acceptable ratio between actual and minimum expected travel time, is converted into a Delivery time window. Tests with 20 vehicles show significant improvements, ranging from 3.83% up to 10.74% in comparison to results of the original algorithm. It is observed that the new approach produces better solutions for problems with a small number of vehicles and a high number of orders.

In a further work, **Gendreau et al. (2006)** investigate an uncapacitated MLPDP with soft time windows, in which up to 100% of the orders occur dynamically. The objective is the minimization of an equally weighted cost function consisting of travel time, lateness and overtime. The authors rely on the same optimization framework proposed in Gendreau et al. (1999). The *parallel Tabu Search with Adaptive Memory* is primarily changed in terms of the basic neighborhood. Instead of Cross Exchange, an *Ejection Chain* procedure is applied. *A request (Pickup and Delivery) is taken from one route and moved to another route, thus forcing a request from that route to move to yet another route, and so on.* The chain may be of any length and may be cyclic or not.

For testing purposes, three main scenarios, each including five instances with increasing requirements to the solution procedure ("temporal utilization" of vehicles 28%, 57% and 78%, respectively) are generated. See Section 4.4.2 for a detailed presentation of these dynamic test sets. A comparison of the new approach is carried out with the adapted benchmark heuristics from Gendreau et al. (1999). It is worth mentioning that *simulation is run in real time*, producing realistic time intervals between events for running the improvement.

Best results are achieved with the application of the new approach. The results of "complete solution rebuild" show worst performance because previously obtained solution structures get lost. With increasing stress, the solution quality of Tabu Search with Adaptive Memory and benchmark heuristics gets progressively closer. The authors explain this finding by a lack of computation time between two consecutive events. When parallelization with 16 processors is applied, additional improvement ranges from 2.2% to 5.7%.

Evolutionary Approaches

This group of metaheuristics is inspired by *biological evolution.* Candidate solutions to the optimization problem play the role of individuals in a population, and each individual's fitness is determined by its associated objective function value. The procedures initially generate a diversified pool of solutions (parent generation), which is used as origin for improvements (child generations). The improvements are achieved by using mechanisms like "selection", "recombination", or "mutation". The idea of "Genetic Algorithms" goes back to Holland (1975).

Again, all five selected papers deal with a local geographical area. Interestingly, three of five publications explicitly consider dynamic travel times: two in combination with dynamic requests, and one focusing only on dynamic travel times.

Haghani and Jung (2005) consider a capacitated VRPTW with 55% dynamic requests and dynamic travel times. The dynamism of travel time is modeled by varying link travel speed, which is calculated as the link's average speed multiplied by a dynamic factor, depending on the time of day. Information about time-dependent variations in travel time is not ex-ante known to the planning algorithm. The objective is the minimization of a weighted cost function with costs for used vehicles, traveled distance and violation of time windows.

The authors present a rolling horizon based *Genetic Algorithm* with the following *encoding*. A feasible solution for the VRPTW consists of a sequence of four-digit numbers. Each number belongs to a real order, in which the first digit indicates the assigned vehicle, and the last three digits are used as sorting keys for each vehicle's routing. At the beginning, an initial parent population is randomly generated and evaluated for fitness. Afterwards, new individuals (children) are generated by applying *two-point crossover, mutation and vehicle merging*. Individuals of a new generation are selected from both the parent and children generation. The best solution is always passed onto the next generation (*elitist strategy*), while the remaining solutions are chosen with probabilities depending on their *fitness*.

Solution quality is evaluated by a comparison with exact CPLEX solutions (for very small problems with less than 10 demand nodes) and lower bounds (based on a relaxed MIP formulation). Results show gaps of less than 5% for 5 to 25 demand node problems. For a 30 demand node problem with 30 time periods, the gaps increase up to 7.9%. Overall, the authors report excellent results within very short computation times. In addition, the new approach is applied to a larger case study's data, which shows the benefits of reacting to dynamic changes in travel time by comparing the performance of a plan revision strategy with a non-revision strategy.

Pankratz (2005)[7] investigates a capacitated PDP with hard time windows and up to 100% dynamically occurring customers. The objective is the minimization of total traveled distance. All orders have to be served and, if necessary, an additional vehicle is introduced. As solution approach, the author proposes a *"Grouping" Genetic Algorithm*.

Pankratz argues that in an MLPDPTW, the assignment problem of orders to vehicles has more influence on solution quality than the routing problem. This is motivated by the fact

[7] Based on the author's dissertation: Pankratz (2002)

that time windows and precedence constraints considerably restrict the number of routing alternatives for a given allocation of requests. In the presented approach, an MLPDPTW solution ("chromosome") is therefore encoded by clusters of requests assigned to a single vehicle ("genes"). Additionally, a chromosome contains routing information for each gene, which is hidden from the Genetic Algorithm and cannot be directly manipulated by the genetic operators. The encoding is visualized in Figure 3.5.

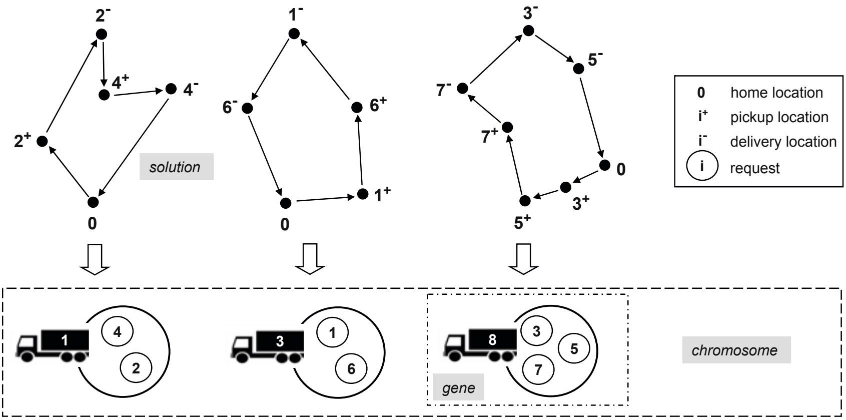


Figure 3.5: Encoding of the "grouping" Genetic Algorithm (cp. Pankratz, 2005)

An initial population is generated by repeated *Best Insertion* of all requests in random order until the desired population size is reached. The algorithm successively selects pairs of individuals with regard to their fitness and generates two children by applying *crossover and mutation operators*. For Crossover, two crossing sections are specified in each parent. Then each parent's section is inserted into the other parent. To yield feasible solutions, some repair actions, e.g. elimination of vehicles or elimination of orders occurring twice, are performed. A subsequent mutation randomly eliminates a gene and re-inserts the associated requests.

If a new request occurs, all irreversible decisions of the "plan in execution" up to this point in time are discovered, and a synchronization of all individuals of the population is carried out. Then, the new request is inserted into all individuals. Subsequently, the Genetic Algorithm is restarted, and, after termination, the best solution is picked as new "plan in execution". Dynamic test data sets are derived from the static PDPTW instances of Li and Lim (2003) (cp. Section 3.4 for details). Two insertion heuristics without improvement are used for comparison: (i) incremental insertion and (ii) total plan revision ("from scratch") every time a new request occurs. Both methods are significantly outperformed by the GA, which produces up to 5% reductions in traveled distance. However, with an increasing degree of dynamism, the gap between GA and insertion heuristics shrinks.

Hanshar and Ombuki-Berman (2007) report on a capacitated VRP with dynamically occurring customer requests and the objective of minimizing traveled distance. A rolling horizon based *Genetic Algorithm* is presented. Similar to Montemanni et al. (2005), dynamic information is batched up to the end of equal time slices and processed in the following time slice. During time slices, "optimization" is run based on known data. At the end of each time slice, the best known solution is chosen for execution. Decisions that have a processing time starting within the next time slot are permanently fixed.

The *encoding* of a VRP solution is performed by a series of positive and negative integer numbers, describing the sequence of orders. While positive numbers indicate open (not fixed) orders, negative numbers are used for identification of order bundles that have already been assigned to a specific vehicle. The series of numbers is traversed from the left to the right, successively assigning orders to vehicles. When a vehicle's capacity is reached, or when a negative number occurs, a new vehicle is introduced.

An initial population of 400 individuals is randomly generated and evaluated. Afterwards, individuals are chosen for *Crossover* according to their fitness. For crossover, a route from each parent solution is randomly selected, and the customer orders present in each route are removed from the *other* parent. Then, the customers are reinserted with *Best Insertion*. This is repeated until a sufficient number of feasible solutions for the next generation is available. Some of the new solutions are subject to the mutation operator that reverses the sequence of orders between two randomly chosen cutting points. Finally, 1% of the worst new solutions is replaced by the 1% best solutions from the parent generation.

Test data and benchmarking results are taken from Montemanni et al. (2005), who have developed an Ant Colony based approach. In addition, the authors develop a Tabu Search approach with the neighborhoods inversion and λ - interchanges (1,0) to (3,3). Nevertheless, best results are obtained with the new Genetic Algorithm, followed by Tabu Search and Ant Colony. The GA outperforms Montemanni's Ant Colony results by 5.26% on average.

Cheung et al. (2008) deal with an MLPDP with hard time windows, in which travel times and the occurrence of new requests (up to 16%) are subject to dynamism. Dynamic orders may be rejected only if there is no feasible insertion position. The goal is the minimization of total travel time.

The authors propose a *genetic solution approach*, which is triggered by the arrival of new information (new orders, changes in travel time). The *encoding* of a solution is performed as follows. For each order, a triple of numbers ("a gene") is stored, where the third number denotes the assigned vehicle, and where the first and second numbers denote the routing positions of Pickup and Delivery in the vehicle's tour. An initial population is generated by first building pendular tours (depot - Pickup - Delivery - depot), with orders lying at a prespecified distance from the depot (subject to variation), and by subsequently inserting the remaining orders into the pendular tours.

When a sufficient population size is reached, solution pairs are selected for *Crossover* (according to fitness). In the Crossover operation, a random number of genes from parents A and B is exchanged. If the *fitness value* of one of the new emerging solutions outperforms both parents, the parents are discarded and replaced by their offspring. Otherwise, both parent solutions are kept in the population. Afterwards, each new generated solution is subject to *mutation*. For a random number of requests, the assigned vehicle is changed. The mutated solution is only accepted if it has a better fitness value than before.

The authors use some self-generated test instances with customer locations evenly distributed in a unit square. While information about the number of dynamic customers (four in all test instances) is available, frequency and magnitude of travel time variations remain undefined. The impact of dynamic data on solution quality is investigated by applying the new algorithm on the associated static case, with resulting gaps between 5% and 10%. According to the authors, the effectiveness of the dynamic re-optimization is quite high.

Okhrin and Richter (2008) consider a capacitated VRPTW with time-dependent and dynamic travel times. For each pair of nodes, four travel time values are specified, associated with different day time intervals. In addition to this a-priori known information, travel time is subject to dynamic fluctuations, which are modeled as normally distributed deviations $N(0,4)$ and $N(0,9)$. The objective is the minimization of total travel time.

As solution approach, the authors propose a *Genetic Algorithm*. The initial population is created by first sorting customers according to the urgency of their time windows and the subsequent application of *Best Insertion*. A new generation of the same size is generated with a selection procedure, followed by *crossover and mutation*. For selection purposes, a random number of individuals is repeatedly chosen from the whole population. In every round, one individual from the chosen subset reaches the next generation (with a probability of 80% for the fittest one, otherwise a randomly selected individual of the remaining subset).

Afterwards, 90% of the selected individuals are subject to the Crossover operator. Partial routes are randomly chosen from two individuals. Then the associated orders are removed from the other respective individual, followed by Best Re-Insertion. The mutation operator is applied to 10% of the offspring. A random customer within each individual is exchanged with the customer that has the most similar time window. Finally, the 1% best solutions from the old population are transferred to the new one, replacing the worst individuals (*"elitist strategy"*).

The authors test their approach with modified Solomon (1987) data and prove the efficiency of the Genetic Algorithm for static planning situations. In addition, the benefits of dynamic reactions to fluctuations in travel time are shown by comparison with a "no reaction strategy".

Variable Neighborhood Search

The concept of Variable Neighborhood Search was proposed by Mladenovic and Hansen (1997). The basic idea of VNS is to search for improvements from the current best solution, first using the *smallest neighborhood* in order to randomly (!) generate one new solution. This solution serves as starting point for another local search procedure, which is executed until a local optimum is found. If the local optimum is a new best solution, the search is re-started from this new solution. Otherwise, the *"radius" of the neighborhood around the original best solution is successively increased.*

Two publications using VNS in dynamic environments are selected. The first one focuses on generating a concept for how to realize VNS in a dynamic situation. The second publication also reports on computational tests. A main feature of the second approach is the explicit consideration of "transshipment options".

Angelelli et al. (2004) investigate an uncapacitated VRP with hard time windows, in which new requests occur dynamically. Orders are classified into priority levels according to their time window's urgency. The objective is to maximize the total priority value of the served requests. Generally, orders may not be rejected, but some orders with time windows that are more distant may be postponed to the next shift.

The authors present a concept for a rolling horizon based solution procedure that is applied in fixed specified time intervals. A plan is made feasible by inserting all unpostponable orders with the help of Best Insertion, re-arrangement of orders, and extraction of postponable orders. Subsequently, a *Variable Neighborhood Search (VNS) improvement procedure* is applied. A neighborhood is defined by the number k of postponable requests, which are extracted from the current solution. These orders are labeled "tabu" and stored in a pool of not assigned postponable orders. As many "non tabu" postponable orders as possible are then attempted to be inserted into the current plan. Computational results are not reported.

An algorithm much like "Variable Neighborhood Search" is also applied by **Bock (2010)**. The author focuses on a dynamic MLPDPTW, including several sources of dynamism: requests, vehicle breakdown, vehicle deceleration, route blockage, and traffic congestion. The main contribution, however, is the integration of **multi-modal transport chains and multiple transshipments**. The author models four *shipment scenarios* for transporting a load from a Pickup to a Delivery location (an exemplary visualization can be found in Figure 3.6):

- direct transportation,
- transportation making use of one transshipment point (depot or hub),
- transportation making use of two transshipment points (two depots, or one depot plus one hub), and
- transportation using several transshipment points (several depots, plus one hub if necessary).

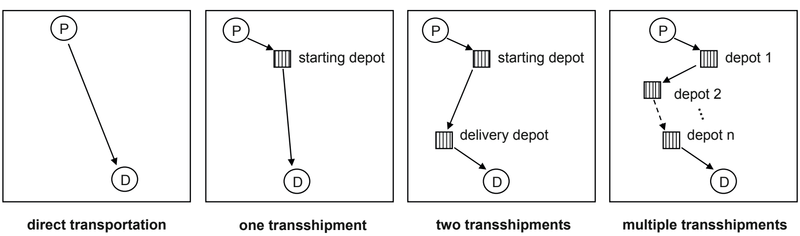


Figure 3.6: Exemplary illustration of available shipment scenarios (cp. Bock, 2010)

The framework also allows for a kind of *en route diversion*. A directly scheduled transportation task, which is already loaded onto a vehicle, can be dynamically exchanged at the next transshipment point, thus enabling the assignment of another task to this vehicle. Construction and improvement procedures are based on *Best Insertion*. In the improvement part, a specific number of requests is extracted and re-inserted, investigating all possible transshipment scenarios. If no improvement can be found, the number of extracted requests is successively increased (VNS). As soon as an improvement has been found, it is accepted and the procedure starts with the first step (=just extracting one request).

For testing purposes, the author generates data sets whose structure is "derived from practical applications". He compares a *time-based* (continuous improvement with fixed anticipation horizon) and an *event-based* simulation technique. Better performance is achieved with the time-based approach. In addition, a rule-based benchmark procedure is applied to the test data. As expected, this procedure is clearly outperformed by the VNS-like procedure. Finally, it is proven that the availability of complex transshipment structures results in more transportation options and therefore yields better overall results.

Ant Colony

The concept of Ant Colony optimization was proposed by Dorigo (1992), who was inspired by the behavior of ants seeking a path between their colony and a source of food. Each ant lays down a *pheromone trail* on the paths it travels. If other ants find such a path, they are likely to follow the existing trail, thus *reinforcing* it. The concept is transferred to Vehicle Routing in order to find "optimal" paths.

For this metaheuristic, two dynamic publications have been chosen that consider dynamically occurring customers for a VRP (Montemanni et al., 2005), and dynamic changes in customers locations for a TSP (Guntsch and Middendorf, 2002).

Montemanni et al. (2005) propose an *Ant Colony approach* for solving a capacitated VRP with dynamically occurring requests. The objective is the minimization of total travel time. In order to handle the dynamic requests, the working day is divided into time slices of equal length, wherein new orders are batched. These new orders are incorporated during the planning run of the subsequent time slice. Over the period of each time slice, the ant colony heuristic is run and the best solution that is found is realized at the beginning of the next time slice.

The ant colony heuristic works as follows. Every ant produces a feasible VRP solution by choosing customers successively according to given arc probabilities. The probability of visiting customer j after customer i depends on two factors: the general attractiveness of the arc (depending on travel time) and the *pheromone level* (indicating how proficient it has been in the past to visit j after i). When an arc has been chosen within one generation of ants, its pheromone level may be locally reduced to favor exploration. Once all ants of the colony have completed their computation, the best known solution is used to globally modify the pheromone trail. In this way, a "preferred route" is memorized in the pheromone trail matrix and future ants will use this information to generate new solutions in a neighborhood of this preferred route. To reduce the impact of "older" solutions, some pheromone information is *evaporated* at the beginning of each time slice.

On the basis of dynamic test data from Kilby et al. (1998), the authors compare their approach with an easy heuristic (Nearest Neighbor construction and Best Re-Insertion improvement). Better performance is achieved by the new Ant Colony approach with an average decrease in total travel time of 3.2%, compared to the easy benchmark heuristic.

Guntsch and Middendorf (2002) deal with a TSP with dynamically changing cities. While the total number of cities is kept constant, a fixed number of random cities is exchanged with other cities (from a pool of cities) every t time units. The objective is the minimization of total traveled distance. The authors propose a *population based Ant Colony approach*, which connects Ant Colony with aspects of a *Genetic Algorithm*. Instead of transferring pheromone information, a set of solutions is transferred from one iteration of the algorithm to the next. This set of solutions is then used to compute the pheromone information for the ants of the next iteration. A specified number of ants generates TSP solutions, in which each routing decision depends on the probability of the optional links. With a probability of 0.9, the arc with the highest probability is chosen. With a probability of 0.1, one of the other arcs is chosen according to their individual probability.

In order to update the pheromones, the authors investigate several strategies for replacing an old solution by the best new generated solution. It turns out that the best strategy is either to simply replace the oldest solution or to randomly choose a solution for exchange (with higher probability for an inferior one). As a consequence of dynamic changes in cities, the solutions in the population are altered infeasible. To overcome this problem, the authors discuss a *Complete Restart* or a *Repair by Best Insertion*. Repair performs better when only minor changes in data have to be included, while Restart is preferable in situations with higher dynamism.

Second Objective Function

Finally, a last metaheuristic concept is considered (cp. Helay and Moll, 1995), which differs slightly from the others. In order to escape from local optima, a *Second Objective Function* is introduced with the goal of temporarily deteriorating the *primary objective function*. This secondary objective function should be rather different from the main one in order to drive search far enough, but should also be partially dependent on the primary one, in order to avoid worsening its value too strongly.

Xiang et al. (2008) consider a dynamic Dial-A-Ride problem with soft time windows. Nearly *all possible sources of dynamism are regarded:* arrival of new requests, fluctuations in travel speed and service time, no-shows of customers and cancelation of requests, traffic jams and vehicle breakdown. An algorithmic solution concept based on *Best Insertion*, supplemented by improvement with basic versions of *intra- and intertour exchanges*, is proposed.

The *primary objective function* minimizes a weighted cost function, including vehicle fixed costs, distance, travel time, waiting, service time, violation of maximum travel time, overtime and delay. The *secondary objective function* is chosen in a similar way, focusing specifically on the costs of empty distance, empty travel time and empty waiting time. For testing purposes, a data set is self-generated. Interestingly, not all possible sources of dynamism are tested at once. Instead, the impact of each source is investigated separately with the following findings: while long term traffic jam and vehicle breakdown cause severe modifications in the schedule, cancelation of customers, travel time fluctuations and service time variances induce only minor changes.

After the treatment of different metaheuristics, we will continue with the next "main algorithmic solution concept", which is based on exact procedures.

3.3.1.3 Heuristic Application of Exact Procedures

The third group of dynamic myopic approaches is based on exact procedures, which are applied to the static subproblems of a dynamic instance (including all available information up to a specific point in time). Since the dimension of those static subproblems is usually quite big and due to limited computation time to solve a static subproblem, the regular application of exact solution procedures is less suitable. Instead, the exact procedures are applied only to a relaxed subproblem combined with some subsequent repair mechanism. Another option actually applies the original exact procedure, but interrupts it after some time, using the best solution found so far.

Those approaches are called *heuristic application of exact procedures*. Due to the character of dynamic problems, "not finding the exact solution" of a static subproblem is not dramatic. As explained in Section 2.1, a "better" solution of a subproblem may not necessarily result in a better overall solution of the dynamic problem.

In the following, a *Column Generation* based approach and a procedure using *Lagrange relaxation* are reviewed. Afterwards, three publications that try to solve the static subproblems with *CPLEX solver* are presented. Finally, the idea of using a *linear assignment procedure* is explained, which is actually solved exactly for each static subproblem (but with the input information for the assignment matrix including some heuristical calculations). Table 3.4 summarizes the properties of the selected publications.

Column Generation

Chen and Xu (2006) consider a capacitated VRP with hard time windows, where up to 75% of customer requests occur dynamically and have to be completely served (no rejection). To fulfill this task, an infinite number of vehicles is available. The objective is the minimization of total distance traveled.

The planning horizon is equally divided into decision epochs (with a length of one or two minutes, according to the scenario). Solutions are successively fixed on a rolling horizon basis up to a prespecified point in time (anticipation horizon). The authors apply a heuristic solution approach, based on *Column Generation*, where a column corresponds

	dynamic aspects		capaci-tated	problem	TW constr.	area	en route diversion	dynamic test data
	orders	other						
Column Generation								
Chen and Xu, 2006	X (75%)		cap.	VRPTW	hard	local	X	Solomon, 1987
Lagrange Relaxation								
Li et al., 2009		vehicle break-down	cap.	VRPTW	hard	local		Solomon, 1987
Application of CPLEX solver								
Yang et al., 1999	X		cap.	SLPDPTW	hard	wide	X	self-generated
Mahmassani et al., 2000	X		cap.	SLPDPTW	hard	wide	X	self-generated
Kim et al., 2002	X (100%)		cap.	SLPDPTW	soft	wide	X	self-generated
Linear Assignment								
Fleischmann et al., 2004	X (49%)	travel time	cap.	SLPDPTW	soft	local		real-life
Powell et al., 2000a	X (70%)	travel time	cap.	SLPDPTW	soft	wide		real-life
Powell et al., 2002	X		cap.	SLPDPTW	soft	wide		real-life

Table 3.4: Heuristic application of exact procedures

to a single vehicle trip. The algorithm consists of two levels. At the first level, a *heuristic is used to generate new columns and to update old columns* (e.g. insertion of new orders). At the second level, a *set-partitioning-type formulation is exactly solved with CPLEX*. Information about dual values of orders is exchanged between both levels in order to guide the local search heuristics. En route diversion is explicitly permitted.

For testing purposes, dynamic extensions to the VRPTW instances of Solomon (1987) are generated. The authors first compare their new approach with the best known solutions for the static Solomon instances. Results are inferior, by on average 3.97% for data sets R1, C1, RC1 and 0.54% for data sets R2, C2, RC2. Afterwards, the new dynamic solution approach is benchmarked with a Local Search Approach (Best Insertion plus improvement with 2-Exchange and OR-opt, with unlimited time) and an "unlimited time version" of the new dynamic Column Generation approach. The results show a 5% better performance of the new approach, when compared to Local Descent. Interestingly, when the new approach, having unlimited time available is compared to its version with limited time, it produces better results only for 70% of the instances.

Lagrange Relaxation

Li et al. (2009) consider a capacitated VRP with hard time windows and dynamism induced by **vehicle breakdowns**. It is observed that a VRP with Pickup tasks has to be treated different from a VRP with Delivery tasks in the case of vehicle breakdown. In the Pickup case, other vehicles can just change their routes to collect the packages from the broken down vehicle's customers. In the Delivery case, however, other vehicles have to change their routes to first Pickup the packages loaded on the broken down vehicle and then deliver them to the corresponding customers (transshipment is only allowed from the broken down vehicle, not between the other vehicles). For both cases, Pickup or Delivery, the authors additionally model the option of having a backup car ready at the depot. If available, this vehicle may be used to fulfill parts of the broken down car's Pickup or Delivery tasks.

The authors develop a *Lagrange relaxation based heuristic*, which is supplemented by a *Best Insertion* algorithm to ensure feasible results. The approach is compared to a "naive manual strategy" and to a pure "Best Insertion" heuristic. The "naive manual strategy" just cancels the services if no extra truck is available at the depot. If such a backup car is available, it is simply sent to continue the tasks of the broken down car. In the case of Delivery, it has to drive first to the breakdown point to collect the loaded packages.

For testing purposes, the static Solomon (1987) instances are taken. The best-known solutions are used as initial routes of the vehicle re-routing problem. Then, one vehicle breakdown is introduced early in the schedule. Some instances are equipped with a backup vehicle at the depot. Solutions show best results for the Lagrangian heuristic: total costs are reduced by 8.53% compared to the "naive manual strategy" and by 4.46% compared to "Best Insertion". In addition, the authors observe generally more service cancelations for Delivery services than for Pickup services if the same algorithm and settings are used.

Application of CPLEX Solver

Yang et al. (1999) consider an SLPDP with hard time windows, where new requests occur dynamically. The objective is the minimization of a weighted cost function consisting of empty distance traveled, delay (deviation from preferred time within hard time windows) and lost revenue (for rejected orders). A mathematical problem formulation, explicitly allowing en route diversion, is given and several strategies to find good insertion positions of new orders are distinguished:

- the load is simply placed at the end of each truck's current job queue,
- the load is placed at the best position in the queue,
- the load is placed at the best insertion position, considering re-sequencing and reassignment.

The third strategy is implemented with the help of a branch-and-cut procedure in CPLEX and produces optimal solutions for the static subproblems. However, the computational burden is quite high, so that the number of demands which can be reoptimized at any given time has to be limited. As test data, the authors use some self-generated instances of relatively small size. The best results are achieved by applying the "optimal" strategy on a limited number of ten variable requests.

Mahmassani et al. (2000) consider the same problem, primarily discussing strategies for how to reduce the number of variables for the "optimal" insertion strategy. They suggest disregarding the vehicle's next order and the shift of a "cut-off time" from the present to the future, successively removing requests from the pool of potential re-assignment orders as long as the number of remaining orders after the "cut-off time" equals the predefined maximum number of orders. In addition, the merging of spatially close orders is discussed.

Kim et al. (2002) publish a related paper, which is also based on the mixed integer model of Yang et al. (1999), where time windows of the SLPDPTW are now treated as soft constraints. The authors investigate the task of maximizing revenue in an oversaturated system, where half of the dynamic demands have to be rejected.

Several concepts for the acceptance/rejection decision are discussed:

- In the first concept, simply all requests are accepted, until a maximal holding capacity of 360 orders is reached.
- A second concept limits the number of accepted demands that have not been picked up yet to a prespecified threshold of 270 (75% of holding capacity), in order to leave more room for improvement operations to the existing demands.
- In the third concept, the minimal additional empty mileage to reach a new order's Pickup location is calculated with Best Insertion over all vehicles. If the empty mileage is below 18 miles, the new order is accepted.

Improvement is based on the "optimal" CPLEX approach, specified in Yang et al. (1999). To comply with a maximum computation time of 10 seconds, the problem size is reduced. Initially, the vehicle to which the new demand was assigned is chosen, then the spatial proximity of other vehicles' orders is calculated (including a check for time window feasibility). Finally, a subset of promising vehicles plus some random vehicles is chosen for improvement. Every time optimization is restarted, a snapshot anticipating the planning situation after 10 seconds is generated, which serves as starting point for the improvement procedure.

For tests with self-generated test data, the authors introduce the following cost parameters: revenue per loaded mile ($1.2), variable cost per mile, including both empty and loaded movements ($0.57), and daily fixed costs for driver and trucks ($45 + $45). The results show that the acceptance strategy based on additional empty mileage produces the highest revenue. The other strategies are inferior by 4.5% (threshold of 270 orders) and 10.6% (maximum holding limit 360 orders). It is indicated that keeping the number of waiting jobs in the queue below the holding capacity (at about 75%) is more beneficial than accepting and holding as many demands as possible. Response time may be improved significantly when the length of job queues is limited.

Linear Assignment Problems

Fleischmann et al. (2004) consider an SLPDP with soft time windows, where up to 49% of the customers occur dynamically. In addition, travel times are subject to a-priori known time-dependent fluctuations (in intervals of one hour) and dynamic disturbances occurring in 5-minute intervals. The objective is the minimization of a weighted cost function including travel time, delay and overtime.

The authors propose a solution approach, which is based on the optimal solution of a *Linear Assignment Problem*, where all vehicles and all open orders are considered simultaneously. The associated assignment matrix additionally includes some dummy vehicles and some dummy orders, which are introduced to enable postponement of orders (when an open order is assigned to a dummy vehicle) and waiting of idle vehicles (when a dummy order is assigned to a real vehicle). Each time a new event (e.g. new order, completion of an order) occurs, the matrix is updated and the assignment problem is re-solved. An order-to-vehicle assignment first becomes effective when the vehicle's preceding order is completed.

For tests, the authors use real-life data from a local area express service and travel time data from a traffic management system in the city of Berlin, Germany. For comparison purposes, (i) simple assignment rules and (ii) a Best Insertion procedure with improvement (OR-opt and vehicle-to-tour re-assignment) are used. Procedures are run on test data with varying degrees of dynamism, ranging from 0% to 49%. For the completely static case, the Best Insertion procedure shows best results, while for test instances with 49% dynamic customers, the new assignment procedure outperforms all the other approaches. The authors determine an increasing advantage of the new assignment procedure for increasing levels of dynamism and attribute these findings to the preservation of high flexibility that is achieved by fixing the order-to-vehicle assignments at the latest possible time.

Further publications considering the SLPDPTW with solution approaches based on the linear assignment problem are published by **Powell et al. (2000a)** and **Powell et al. (2002)**.

3.3.1.4 Rule-Based Decision Making

The fourth group of dynamic myopic approaches can be classified as "Rule-Based" publications, since *easy decision rules* are applied as a reaction to dynamically occurring information. Some of the following papers also include some more advanced procedures (e.g. Local Search) for comparison purposes. The concept of "Fuzzy Logic" is included here, because it basically reproduces human decision making, by transferring it into a form of rule-based computer decision making.

All of the selected papers consider dynamically occurring customers; in addition, one paper also includes dynamic service times. The SLPDP(TW) is regarded three times; the other two papers deal with a TRP and an MLPDPTW, respectively. Table 3.5 summarizes the properties.

Rule-Based Decision Making

Regan et al. (1995) investigate the potential benefits of *en route diversion* for an SLPDP under idealized conditions. The respective publication is also the first explicitly considering *en route diversion*. The objective is the minimization of empty distance traveled. The authors first consider the idealized *one vehicle and two requests case*. In a circular area with the depot in the center, a vehicle starts traveling to the first order's Pickup location. While the vehicle is on its way, the second order arrives (with Call-In time uniformly and randomly distributed within the vehicle's travel time to the Pickup location).

	dynamic aspects orders	dynamic aspects other	capacitated	problem	TW constr.	area	en route diversion	dynamic test data
Rule-Based Decision Making								
Regan et al., 1995	X		cap.	SLPDP			X	self-generated
Regan et al., 1996	X		cap.	SLPDPTW	hard	wide	X	self-generated
Regan et al., 1998	X		cap.	SLPDPTW	hard	wide	X	self-generated
Larsen et al., 2002	X (100%)	service time		TRP		local		self-generated
Fuzzy Logic Approach								
Teodorovic and Radivoj., 2000	X (100%)		cap.	MLPDPTW	hard	local		self-generated

Table 3.5: Rule-based decision making

A first rule-based diversion strategy, "divert if the new Pickup location is closer to the vehicle's current position", yields savings in traveled distance of less than 1%. When, in addition, the potential empty distance, according to the sequence of both orders, is taken into account, the average reduction in traveled distance is more than 6%.

In a second scenario, *several dynamic orders* occur, with an arrival rate rapid enough so that more than one demand may arrive while the vehicle is en route to the Pickup location. The authors introduce a benchmark solution approach, where optimal re-sequencing is performed at the completion of each loaded movement. If en route diversion is allowed in this approach, the results are improved by about 1% to 2%. However, diversion creates a sort of "zig-zag" effect, where a vehicle is en route and then diverts and then diverts again. Thus, the authors recommend limiting the number of times that one diverts before some demand is serviced and not allowing diversion whenever it is locally better.

The exploration of idealized scenarios suggests that a reduction of traveled distance of between 5% and 10% by applying en route diversion is not unreasonable.

In **Regan et al. (1996)**, the authors extend their problem for profitability-based acceptance/rejection decisions, where a new load is only accepted if the *empty-to-loaded ratio* is smaller than a prespecified threshold value. The ratio is calculated by creating an optimal tour, including the candidate load and the already accepted loads. The *additional empty distance is set in relation to the new order's loaded distance.* Again, the advantage of diversion strategies is proved, which result in a 5% to 7% reduction in overall empty-to-loaded ratio.

Regan et al. (1998) describe a simulation framework to dynamic fleet management systems for the SLPDPTW. They discuss three *load acceptance* and eight *load-to-vehicle assignment* strategies.

The following *load acceptance strategies* are proposed:

(i) a new order is accepted if the number of loads waiting in the system is smaller than a prespecified number ("capacity-based strategy"),

(ii) a new load is accepted if it can be feasibly inserted into the current plan ("feasibility-based strategy"),

(iii) a load is accepted if the *empty-to-loaded ratio* is smaller than a specified threshold value ("profit-based strategy").

For the task of *assigning orders to vehicles*, the following strategies are compared:

(i) First Come First Served,

(ii) Nearest Origin,

(iii) Bipartite Assignment of open orders to available vehicles, triggered by time,

(iv) Bipartite Assignment, triggered by the number of open orders and idle vehicles,

(v) Best Insertion plus Intraroute Changes,

(vi) Best Insertion/Intraroute Changes allowing for en route diversion,

(vii) Best Insertion/Intraroute Changes plus re-assignment of loads between vehicles,

(viii) Best Insertion/Intraroute Changes, allowing for en route diversion and re-assignment of loads between vehicles.

The strategies are evaluated with self-generated test data, based on a circular geographic region with a radius of 417 km. A comparison is made on the basis of the performance indicators average *empty distance*, *waiting time* and *operating profit* for high, medium and low demand environments. Somewhat different assumptions are used: while assignment strategies (i) to (iv) are combined with the simple capacity-based acceptance strategy, strategies (v) to (viii) are combined with the profit-based load acceptance including a time window feasibility check.

When comparing assignment strategies (i) to (iv), the authors find the best results with *Nearest Origin* in high demand environments, while *Bipartite Assignment* performs best in moderate demand environments. The more flexible strategies (v) to (viii) produce much lower waiting times and therefore better customer service for all demand intensities. However, good profit values can only be achieved for moderate demand environments; the profit values for high demand environments in particular are not competitive. The authors explain this finding with the fact that a significant fraction of requests is turned away in strategies (v) to (viii) because of the time window feasibility check. In a final suggestion, a hybrid system that chooses an assignment strategy based on the current congestion level of the system is recommended.

Larsen et al. (2002) investigate a Traveling Repairman Problem (TRP) with up to 100% dynamic customers and completely dynamic service times. The objective is the minimization of total traveled distance.

The authors propose four *Rule-Based solution approaches*:

(i) First Come First Served (FCFS),

(ii) First Come First Served with relocation to the geographic median when the vehicle is idle,

(iii) Nearest Neighbor, and

(iv) First Come First Served within four regions of the geographic area.

For testing purposes, the authors use some self-generated data sets with 20, 30 and 40 customers, occurring in a 10km×10km unit square. The data are constructed with degrees of dynamism ranging from 0% to 100% and with effective degree of dynamism ranging from 0% to 60%. When comparing the rule-based strategies, best results are achieved with *Nearest Neighbor*. Strategy (iv) produced slightly higher route lengths and FCFS, as well as FCFS with relocation to the median the longest. For increasing degree of dynamism, the authors report a linear increase in route length across all policies. Interestingly, for increasing *edod* between 48% and 57%, the results show decreasing total travel times. Generally, the results of different rule-based strategies converge with higher levels of dynamism.

Fuzzy Logic Approach

Teodorovic and Radivojevic (2000) investigate a capacitated MLPDP with soft time windows, where all orders occur dynamically. The associated decision problem is split into the subproblems "assignment" and "routing/scheduling" with two different objective functions. For the assignment decision, the objective is to minimize the sum of total distance traveled and waiting time; for the routing/scheduling decision, the goal is to minimize distance and time of detours for new customers.

The authors propose a *"Fuzzy Logic" method that tries to replicate a human dispatcher's decision-making process, based on previous decisions taken by a skilled dispatcher.* For the assignment decision, in a first step a *membership function* is derived. This function transforms (the explicitly calculated) additional vehicle distance and additional waiting time into the categories "big", "medium" or "small".

Afterwards, an *approximate reasoning algorithm* translates the findings into a dispatcher's preference strength. When, for example, both the additional distance and the additional waiting time are "small", the preference for assigning the new order to the associated vehicle is "very strong". Within the pool of vehicles, those currently traveling are favored over idle vehicles. A similar procedure is also applied for the routing/scheduling decision. The authors test their approach with some self-generated instances and report "very promising results".

3.3.1.5 Multi-Agent Systems

Since Multi-Agent Systems differ substantially from the previous approaches, they are treated separately. In contrast to other procedures, there is no global view. Instead, multiple interacting agents with specific objectives decide about subproblems. Solving dynamic myopic SLPDPTW's with a Multi-Agent system was considered by a publication of Mes et al. (2007), whose specifications are given in Table 3.6.

Mes et al. (2007) consider an SLPDP with soft time windows, which is in particular

	dynamic aspects		capaci-	problem	TW	area	en route	dynamic
	orders	other	tated		constr.		diversion	test data
Mes et al., 2007	X	several	cap.	SLPDPTW	soft	local		real-life and self-generated

Table 3.6: Multi-agent systems

applicable for local area scheduling of Automated Guided Vehicles (AGV). Dynamism is primarily induced by requests; in addition, a single random variable is modeled, specifying the total time from arrival at the Pickup location until completion of service at the Delivery location. Since a vehicle moves empty immediately to the assigned Pickup location and waits over there, the random time interval includes: waiting for Pickup, loading at the Pickup location, driving from the Pickup to the Delivery location, waiting for unloading and unloading at the Delivery location.

The major contribution of the paper is the development of a *Multi-Agent Based Procedure*. Instead of a central planning instance, the authors model several agents (for every vehicle and every request) that interact with the help of a market mechanism. The *job agent's* objective is to arrange transportation of the corresponding load before due time at minimal costs, while *vehicle agents* try to maximize their profit by deploying capacity. Both meet at the "marketplace", where job agents request prices for their specific transportation task. Each vehicle agent submits a quote, based on its current scheduling. Afterwards, in a *Vickrey auction manner*, "the best (lowest) price vehicle agent" wins the bid (getting a payment for transportation equal to the second lowest offer). If all quotes are above a certain threshold (calculated with respect to the request urgency), the assignment of the request may be postponed by the job agent.

Improvement is achieved by specific agents: A *fleet agent* is responsible for a subset of vehicles and tries to re-assign jobs between these vehicles. A *shipper agent* is responsible for a set of orders, he may re-allocate orders within the already acquired transportation capacity of its job agents.

The procedure is tested with a data set derived from an AGV system at the Amsterdam Airport, Schiphol. For benchmarking, the authors use hierarchical scheduling methods (cp. Ebben et al., 2005), which distribute vehicles amongst nodes at the top level, while actual load-to-vehicle assignment is performed at the node level. These simple approaches are significantly outperformed by the proposed multi-agent procedure, especially in terms of empty travel time and total costs.

The group of *dynamic myopic approaches without knowledge of the future* has now been considered in detail. The approaches which are presented in the following section also have not available any knowledge of the future. However, there is a decisive difference to the previous ones: They try to identify and to apply ways of anticipating the future.

3.3.2 Strategies Anticipating the Future without Knowledge of the Future

We now give attention to publications which propose strategies for how to construct myopic solutions in order to leave open space (route slack) for the viability of future yet unknown events. These strategies try to reduce the probability that an urgent request (with tight time windows) arrives and the only vehicles that can serve it are already committed, so that servicing this new request may have to be delayed or the request may even be rejected. These strategies also try to postpone final fixation of decisions for as long as possible, in order to have more options to incorporate further new information.

Basically, three concepts have been proposed: *vehicle waiting strategies*, *different objective functions for short and long-term decisions* and *request postponement of non-urgent requests.* In addition, an extension to the Multi-Agent-based procedure by Mes et al. (2007) is presented, which leaves open time slack by applying opportunity-based bid-pricing. In a strict sense, the acceptance-rejection strategies by Regan et al. (1996, 1998) are also directed to "leave open some slack" for profitable future requests.

The **advantageousness of different waiting strategies**, in order to efficiently distribute waiting time along a dynamically constructed route, was compared by **Mitrovic-Minic and Laporte (2004)**. The authors consider an MLPDP with hard time windows for a local area courier service.

Waiting after service allows the accumulation of requests by the planner, which may result in better routing and scheduling decisions. However, waiting may also result in some wasted time that could have been used to serve additional requests. Four waiting strategies are included in the comparison:

(i) *"Drive First"* - A vehicle leaves its current location at the earliest possible departure time. This may result in waiting time at the next location if the vehicle arrives before the time window opens.

(ii) *"Wait First"* - A vehicle leaves its current location at the latest possible departure time, that means it arrives at the next location at the end of this location's time window. An advantage is that more requests are known at the time the vehicle leaves, resulting in more potential for optimization. On the other hand, more vehicles are required, because the strategy tends to concentrate long waiting time in the first part of the route, leaving too little waiting time in the remainder.

(iii) *"Dynamic Waiting"* - The requests in a route are clustered to "service zones" (related to spatial and time distance). Within each service zone, vehicles drive according to "Drive First", between service zones according to the "Wait First" strategy. The strategy is illustrated in Figure 3.7 (cp. Mitrovic-Minic and Laporte, 2004).

(iv) *"Advanced Dynamic Waiting"* - Identical to "Dynamic Waiting" with the extension that total waiting time between service zones is spread proportionally.

The authors' algorithmic approach uses Best Insertion to generate an initial solution, supplemented by Tabu Search improvement, based on an ejection chain neighborhood. Test data, with completely dynamic requests, were derived from two courier companies, operating in Vancouver, Canada. Best results are achieved for the "Advanced Dynamic Waiting" strategy, which produces solutions with up to 8% shorter route lengths compared to "Drive First" and close to or shorter route lengths compared to "Wait First". The number of vehicles is either close to the number of vehicles used in "Drive First" or better. "Advanced Dynamic Waiting" also outperforms "Dynamic Waiting" in all aspects.

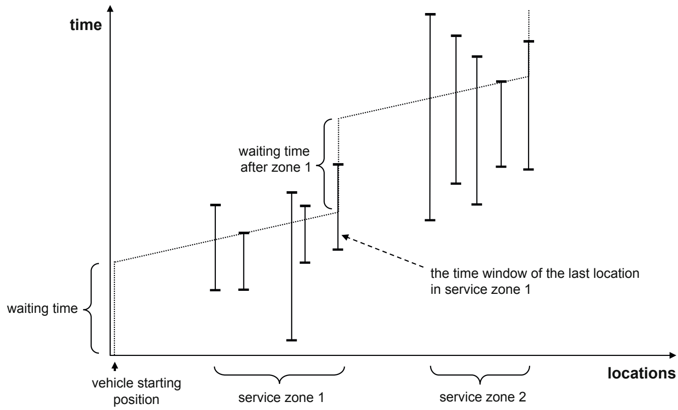


Figure 3.7: Waiting times generated by a Dynamic Waiting strategy

Another paper, considering the **advantageousness of dynamic waiting strategies** was published by **Branke et al. (2005)**. In contrast to Mitrovic-Minic and Laporte (2004), the authors regard a VRP where only one dynamic customer occurs. The objective is to find a waiting strategy in order to maximize the probability that this customer can be serviced within a feasible plan.

Several simple waiting strategies and an Evolutionary Algorithm are presented. The strategy "variable" ("serve all known customers in a tour and finally wait at the last customer's location") and the Evolutionary Algorithm perform best in simulations on a modified test data set based on Beasly (1990). It is shown, compared to the reference strategy "not to wait", that the best waiting strategies are able to reduce the probability of not being able to serve a customer by 10%, while the average length of the detour for a new customer was reduced by approximately 35%.

In a second work, **Mitrovic-Minic et al. (2004)** consider the identical problem to Mitrovic-Minic and Laporte (2004), but now investigate the application of a double horizon based heuristic. By using **different objective functions for short and long-term decisions**, the authors try to achieve better flexibility to incorporate future events. The short-term goal is to reduce traveled distance, while the long-term goal in addition considers maintaining the routes in a state with plenty of options for future requests. Decisions which restrict future planning options are penalized.

The authors compare the *double horizon approach* with a *standard rolling horizon approach*, applying the Tabu Search heuristic introduced in their preceding paper. Again, two test data sets from Vancouver courier services are used, which both range over a 10-hour service period. The short-term horizon is empirically chosen as one hour and two hours, respectively. The authors report superior behavior of the new double horizon based approach, leading to improvements in total route length ranging from 3.6% up to 7.6%.

Pureza and Laporte (2008) investigate a **combined vehicle waiting and request buffering strategy** for a dynamic MLPDP with hard time windows, with the objective of minimizing a weighted cost function (including: number of lost requests, number of used vehicles and total traveled distance). Customers occur dynamically, in addition time-dependent travel times are considered. The authors first prove the advantageousness of a *basic waiting strategy (WE)* compared to "Drive First". A waiting time at the present location is scheduled in order to avoid early arrival (and waiting) at the subsequent location. The waiting time is determined to be just as long as to ensure punctual arrival at the subsequent location's EPT. This enables the consideration of new incoming events in the short-term routing, which may change the decision of the next planned locations, or leave it unchanged without delaying the beginning of service. This basic waiting strategy was also successfully employed in Fleischmann et al. (2004).

Afterwards, *WE* is extended with a version *making use of time-dependent travel times, WE_FP*. It is evaluated whether indirect paths to reach a location yield shorter travel times, thus allowing for a further increase in waiting time at the present location. For a dynamized Li and Lim (2003) PDPTW data set, the authors report a reduction in lost requests of 2% to 8.3% and a reduction of used vehicles by 1.9% to 4.3% when applying WE_FP (compared to the basic version of WE). In a further step, a *request buffering strategy (WE_RB)* is added, which postpones the assignment of some non-urgent new requests to the next route planning cycle. This add-on achieves a further increase in solution quality, especially the traveled distance can be systematically reduced.

Finally, the authors report best impacts of their combined waiting and buffering strategy for a degree of dynamism between 0.4 and 0.6. If the *dod* is too low (<0.2), there is actually no positive effect at all. Only minor positive effects are reported for high degrees of dynamism. Due to dispersion of incoming requests over time, postponement activities in the beginning do not result in a sufficient mass of decision options, thus diminishing the strategy's ability to produce improved results.

In a further *Multi-Agent based publication*, **Mes et al. (2010)** propose a concept of how to improve the pricing technique of an individual *vehicle agent*. They introduce a *time slack measure*, which indicates the maximum amount of time that a job can be postponed by, without causing an increase in delay (for itself or one of the succeeding jobs). When calculating a *bid price* for a possible new job, the *decrease in time slack* between already accepted jobs is considered. With the help of historical demand data, the authors translate the required time slack into an expected profit of future moves, which has to be at least compensated by the reached bid-price of the possible new order. The authors test their **opportunity-based bid-pricing approach** with a small self-generated data set with up to nine nodes and ten vehicles and report a 10% reduction of system-wide logistical costs if the new concept is used by all vehicle agents.

3.3.3 Dynamic Stochastic Approaches with Explicit Knowledge of the Future

Information about the future is usually given in the form of probability distributions, e.g. covering the spatial and temporal occurrence of new orders. A popular approach for handling future information, which is applied in five of the subsequently twelve presented

	dynamic aspects		stochastic information	capacitated	problem	TW constr.	area	solution approach	test data
	orders	other							
Bent and van Hentenryck, 2004	X (80%)		expected total number of requests, probability distribution of temporal request arrival per customer location	cap.	VRPTW	hard	local	sampling “multiple scenario approach”	Solomon 1987
van Hemert and La Poutre, 2004	X		probabilities about “fruitful regions”, where new loads are likely to occur	cap.	VRPTW	hard	local	sampling “probable orders”	self-generated
Ichoua et al., 2006	X (75%)		orders occur according to Poisson process with arrival rate depending on geographic region and time period	uncap.	VRPTW	soft	local	waiting opportunity	self-generated
Hvattum et al., 2006	X (50%)		orders occur according to Poisson process with arrival rate depending on geographic region	cap.	VRP	hard	local	(i) recourse function, (ii) sampling	real-life
Hvattum et al., 2007	X (50%)	amount of demand	orders occur according to Poisson process with arrival rate depending on geographic region	cap.	VRP	hard	local	sampling with improved selection	real-life
Ghiani et al., 2009	X		requests arrive according to known stochastic process	cap.	MLPDPTW	soft	local	sampling (short-term horizon)	self-generated
Kim et al., 2004	X (100%)		spatial and temporal distributions of priority demands	cap.	SLPDPTW	hard	wide	feasibility index (to serve future priority demands)	self-generated
Yang et al., 2004	X (100%)		uniformly distributed customer locations	cap.	SLPDPTW	soft	wide	opportunity costs of serving new jobs	self-generated
Powell, 1996	X		load distribution by origin, destination and Call-In time	cap.	SLPDPTW	hard	wide	multistage stochastic network with approximate recourse function	real-life
Spivey and Powell, 2004	X		the complete future, included in “gradients”	cap.	SLPDP		wide	opportunity cost arcs	self-generated
Larsen et al., 2004	X (23%)	service time	orders occur according to Poisson process with arrival rate depending on geographic region, probability distribution for service times	uncap.	TSPTW	soft	local	probability based reallocation of idle vehicles	self-generated
Liao, 2004		travel time	probability of link travel times, depending on the time of day	uncap.	VRP		local	avoidance of probably congested links	real-life

Table 3.7: Publications with dynamic stochastic solution approaches

dynamic stochastic publications, is *sampling*: the algorithm generates a sufficient number of future scenarios (by drawing from the given probability distribution) and uses the scenarios to approximate a decision's impact on the future.

Another option is the *direct usage* of probability distributions, for example to decide whether a vehicle should wait at its current location or whether it should be relocated to another promising location. The given probabilities may also be used for calculation of *recourse functions*, which include the costs that an assumed scenario does not occur.

Table 3.7 summarizes the main features of the selected *dynamic stochastic publications*. As in the dynamic myopic case, the sources of dynamism are given in the first two columns. Afterwards, the available stochastic information is specified (column 3). In columns 4-7, the associated problem is defined, as well as the geographical extension. The subsequent eighth column includes the solution approach, which is used to integrate the given stochastic information. Finally, information about the used test data sets is provided (column 9).

Bent and van Hentenryck (2004) consider a VRP with hard time windows and with up to 80% dynamic customers. As stochastic information, the algorithm knows the expected total number of customers, customer locations, and the probability distribution of temporal request arrival per customer location. The objective is to maximize the number of serviced customers.

As solution procedure, the authors propose a *Multiple Scenario Approach*: a pool of feasible plans **(sample scenarios)** is maintained, each plan including known and unknown future requests. Whenever a new plan needs to be generated, future requests are randomly drawn out of the known probability distributions. So the resulting plan leaves room for accommodating future requests if they materialize.

A dynamically occurring request is accepted if it can be feasibly inserted into at least one plan in the pool. All plans are continuously kept up-to-date. A plan is deleted, if new information makes it unrealizable. The plan for execution is chosen with the help of a *consensus function*, which selects the plan most similar to the current pool of routings (for every routing, it is calculated how often identical routing decisions, e.g. from location A to B, can be found in other routings). So a *preferably robust plan* is selected, in order to accommodate many dynamic customers in the future.

The authors use modified 100-customer Solomon (1987) test instances and compare the cases of having information about the future available, or not. In the second case, the average number of unserved customers for four different classes of test instances is 1.5, 3.5, 2.3, and 6 on average. In the first case, the average number of unserved customers decreases to 0.75, 1.2, 1, and 2. In general, dramatic improvements by exploiting stochastic information are reported, observing more benefits in environments with a higher degree of dynamism.

In a similar approach, **van Hemert and La Poutre (2004)** deal with a dynamic stochastic VRP, where dynamic loads have to be picked up and transported to a depot within hard time windows. As stochastic knowledge, probabilities are available about "fruitful regions", where dynamic loads are likely to occur. The objective is the maximization of transported loads.

The authors propose an Evolutionary Algorithm, with an initial population of 30 individuals. In addition to known orders, *probable orders*, based on the probabilities for future requests, are generated **(sample scenarios)**. The fitness of the individuals is evaluated by the number of real and *probable* orders which could be feasibly inserted, where the weight of *probable orders* is decreased by a factor α.

When an event occurs, the *best individual* out of the population is chosen as plan in execution. This plan may include anticipated moves, that means vehicles may drive to nodes that have not requested service. The performance of the approach is tested with some self-generated test data. The authors report encouraging results and benefits by the use of information about the future.

Ichoua et al. (2006) examine an uncapacitated VRP with soft time windows, representing an application of Express Mail Service, where parcels are picked up from customers (75% of customer orders occur dynamically) and brought to a central office for further processing. The operational area is partitioned into geographical zones. As information about the future, it is known to the algorithm that orders occur according to a Poisson process, with specified arrival rates depending on geographical region and on time period.

The authors expand the parallelized Tabu Search algorithm with Adaptive Memory, introduced by Gendreau et al. (1999), for a **waiting opportunity**: (i) if the vehicle's next destination is far enough, (ii) if there are not too many other vehicles in the current zone, and (iii) if a new customer is likely to unfold in the vehicle's proximity within a specified time period δ_k, a vehicle is required to wait at its current location for the specified time period.

The approach is tested with self-generated customer data, with associated locations in a 5km$\times$5km unit square. The advantageousness of the new waiting strategy is proved by comparison with the original algorithm: total travel time and lateness can be reduced by 2.3% on average. It is also noted that the new strategy is more effective when it is applied on harder problems (i.e., smaller fleet size or higher request arrival rates.).

Hvattum et al. (2006) observe a dynamic stochastic planning problem at "Linjegoods AS", a distribution company in Norway. The problem is traced back to a capacitated VRP (Pickup), with 50% of the orders arriving dynamically. The geographical area is divided into n$\times$n sectors, where probabilities that a customer shows up in a specific sector are described as a Poisson process. In contrast to Ichoua et al. (2006), the arrival rate depends only on the geographic sector, not on the time of day.

To capture the stochastic elements, the authors first analyze the application of a *two-stage stochastic model with recourse function*, where all unknown information is assumed to be revealed at time t. However, computing the expected recourse cost for a particular solution turns out, even for this simplified case, to be exceedingly difficult.

So the authors proceed with a *rolling horizon based heuristic*, which solves a set of **sample scenarios** (Best Insertion is applied to known customers and to randomly drawn future requests) and then uses "common features" from the resulting solutions to build a preferably "good plan". The decision to serve a customer in the interval under consideration becomes more attractive if such a decision has also been taken in various sample scenario solutions. The objective is to minimize the number of vehicles and total travel time, with greater weight on the first factor.

For comparison purposes, the new approach is benchmarked with a myopic heuristic, which ignores all probabilistic information. While routes produced by the new approach are about 15% shorter, the average number of used vehicles increases slightly. Nevertheless, significant savings are yielded by the new heuristic using stochastic information.

In a subsequent publication, **Hvattum et al. (2007)** modify the selection process from the pool with the **sample scenario** solutions. Instead of only counting the frequency with which a decision can be found in the pool, the authors also include an evaluation step, which tries to avoid overall poor effects. With respect to the objective function, this step discloses the possibility to exclude a few customers with high (solution pool) presence from service and to include a few customers with low (solution pool) presence for service.

In addition, the authors extend their procedure for the case of customers with *stochastic demand* which is revealed first at customer location arrival. If the vehicle discovers that the demand of the customer is higher than the available vehicle capacity, it has to skip the customer completely (split transportation of a load is not allowed) and follow the remaining parts of its tour. The customer has to be serviced later by another vehicle.
Finally, the authors perform some tests for the new program version and report quite good performance, even on problem instances that have radically different properties as compared to the instances for which it was intended.

A similar procedure is proposed by **Ghiani et al. (2009)** for the dynamic stochastic MLPDP with soft time windows. Stochastic information about customer arrivals is used to generate a specified number of **sample scenarios**, which in contrast to Hvattum et al. (2007), cover only a *short-term horizon*. After Best Insertion of randomly drawn requests, the resulting sample scheduling is used for *approximating the future impact of a new real request's insertion*.

The authors benchmark their "anticipatory procedure" with a purely reactive algorithm (not taking into account knowledge of the future) and achieve "dramatic benefits" with the new procedure in objective function value. Since the objective function only covers the minimization of user inconvenience (delay), it should also be mentioned that other important aspects, such as average vehicle utilization, show significantly worse behavior (a decrease of 19%).

Kim et al. (2004) investigate a truck dispatching problem (SLPDPTW) with two types of orders: low price normal orders with wide time windows and high price priority orders with narrow time windows. In an oversaturated system with more than enough requests (100% dynamic), an acceptance/rejection decision has to be made in order to maximize profit. Knowledge of the future is available in the form of spatial and temporal distributions for priority demands, as well as average haul length and required empty distance for this type of order.

Whenever a new request occurs, it is tested whether it can be feasibly inserted into the current plan (allowing for en route diversion). If the test is positive, a priority order is immediately accepted. For a normal order, a **feasibility index** is calculated, based on the current state of the system and the potential inclusion of the normal order. *The index approximates the expected number of vehicles that would be able to serve future priority demands.* The normal order is only accepted if the feasibility index exceeds a prespecified threshold.

Test data sets are self-generated, so that approx. 30% of the demands have to be rejected with an efficient dispatching algorithm. The fraction of priority demands is between 6.25% and 25%. The authors compare their new approach with the rule-based benchmark policies "accept if feasible" and "accept if current number of orders in the system is below a threshold." While the total number of accepted demands is quite similar with all three policies, the number of accepted priority demands can actually be increased by the new solution approach. This results in a significant improvement in total profit.

Yang et al. (2004) deal with an SLPDPTW, where all customer orders arrive dynamically. As information about the future, customer locations are known to be uniformly distributed in a unit square. The objective is the minimization of a weighted cost function for empty movement, for delay, and for lost revenue from job rejections.

The authors propose five planning approaches to support the acceptance/rejection decision. All policies are used to calculate a new order's marginal insertion cost. If marginal insertion cost is smaller than prospective revenue, an order is accepted. In policy (i), marginal cost of serving a new request is calculated over all vehicles by inserting the new request at the end of each vehicle's queue. In policy (ii), all possible insertion positions in each vehicle's queue are considered. Policy (iii) considers the possibility of re-ordering waiting requests within each vehicle's queue. Policy (iv) optimally solves the acceptance and allocation decisions for all open orders with ILOG. Policy (v) in addition incorporates knowledge of the future, by introducing the *opportunity costs of serving new jobs.* Based on the uniform distribution, for a request with central Delivery location it is more likely to find a subsequent order with little empty movement. Hence, **central locations are favoured and remote locations are penalized.**

The paper reports on results obtained with self-generated test data. Optimization policies, simultaneously considering all open orders, appear to outperform the more limited local policies by a significant margin. The worst performance is achieved by policy (i), the best performance by taking into account the future job distribution in policy (v). However, the size of instances which could be optimally solved with ILOG was limited to twenty open orders.

Powell (1996) investigates a dynamic SLPDPTW, where information about the future is available in the form of load distributions by origin, destination and Call-In time. In addition to truck-to-load assignment, it has to be decided whether a driver should be *held in a region or* whether he should be *repositioned empty to a neighboring region* (both, in anticipation of future loads). The goal is the minimization of a weighted cost function, including cost for empty movement, cost for waiting, and cost for rejection of a load.

The author presents a solution approach, which is based on a *stochastic network* with two components: an *assignment network*, including known loads, and a *forecast network*, including forecasted loads as well as known loads lying in the future. Network arcs represent driver-to-task assignments (in the assignment network) and loaded moves, empty moves or waiting times (in the forecast network). An **approximate recourse function** is represented by a cluster of *recourse links*, which capture the expected marginal contribution of each unit of flow into a region in a time period. An approximation of this recourse function value is added to the arc cost and the resulting problem is then solved with a network simplex algorithm on a rolling horizon basis.

The new approach is benchmarked with a completely myopic version of the algorithm, using test data derived from a major truckload motor carrier. Results indicate that the dynamic stochastic approach outperforms the dynamic myopic one by 15%. In addition the author investigates the impact of the total number of trucks (density) on overall profitability and reports substantial improvements ($0.05 per mile) when a larger fleet of vehicles is used.

Spivey and Powell (2004) present a strategy to incorporate advance information into a simple linear assignment model for the dynamic SLPDP. The assignment model explicitly allows "not assigning" a resource/order with the help of arcs connecting each resource/order to a corresponding super-sink.

Information about the future is assumed to be completely available and is made known to the solution approach in the form of different types of *gradients*. That means the assignment problem's arcs are manipulated in order to produce solutions anticipating future information. To realize this task, three different types of gradients are defined: **resource gradients**, **task gradients**, and **arc gradients**. A *resource gradient*, for example, can be viewed as the contribution for not assigning the corresponding resource (vehicle). It is added to the regular cost-value of each arc, which is connected to the specific resource, thus decreasing the vehicle's current attractiveness in anticipation of the future. *Task gradients* work equally for open orders. An *arc gradient* is more specific and captures the impact on the future for each specific arc.

For small problems the gradients are calculated by enumeration of future resources and tasks, for bigger problems a hierarchical aggregation strategy is proposed. The authors compare the solutions achieved with the application of gradients with simple myopic solutions. As expected, all gradient solutions turn out superior to myopic solutions. The best results are achieved with specific *arc gradients*, where solution quality reaches "near optimal levels". However, the computational burden for the arc gradients is the highest, requiring a calculation for every arc, and not just for every node.

Based on a planning problem of a mail service provider in the US, **Larsen et al. (2004)** investigate a dynamic TSPTW, where between 11% and 23% of customer requests and all on-site service times are subject to dynamism. The geographical area is divided into several sub-regions, in which orders occur according to a Poisson process with region specific arrival rates. This information and the probability distribution of on-site service time is a-priori known to the planning approach. Decisions about routing and scheduling for a single vehicle have to be made, with the objective of minimizing the weighted sum of travel time and lateness.

A rolling horizon based solution approach is presented, which uses Best Insertion and subsequent improvement with 3-opt. *A-priori information is only utilized for potential re-allocation of vehicles during idle time* (in anticipation of a better location to serve future loads). Three strategies are compared with a reference strategy of just waiting at the current location:

(i) re-allocation to the nearest prespecified idle point,

(ii) re-allocation to the idle point with the highest arrival rate, and

(iii) re-allocation to the idle point with the highest expected number of immediate requests, depending on vehicle idle time and chosen idle point.

The proposed **re-allocation strategies** are only executed if the probability of receiving at least one new request within the vehicle's idle time is sufficiently high in the chosen idle point's subregion.

The authors perform tests with two data sets, one self-generated, the other based on real-world data. Interestingly, best results in terms of distance and lateness for the first data set are achieved by the reference strategy of just waiting at the current location. For real-world test data, however, strategy (ii) performs best, while the reference strategy proves competitive as well. Results show that using information about the future in the suggested way may not (!) lead to significant improvements.

Liao (2004) reports on a dynamic VRP in the Taichung network in Taiwan and focuses on dynamic travel times. As information about the future, the planning system has available probable link travel times, depending on the time of the day. The objective is the minimization of total travel time.

The author's solution procedure consists of an initial route generation by a Nearest Neighbor heuristic, which is followed by Tabu Search improvement with neighborhood 2-opt. Each time, when changes in link travel times emerge, the travel time matrix is updated with a shortest path algorithm. As result, a vehicle may be re-routed. Probabilistic information about the future is used to generate a **temporal tabu list**, in order to avoid possibly congested links in the traffic network.

In tests based on real-life data, the impact of having available dynamic travel time information is compared to just knowing static travel times. Results show that the objective function value can be significantly improved by real time routing when considering information about the future: decreases in travel time range from 19.74% up to 24.48%.

3.4 Dynamic Test Data

This section reviews the most frequently applied dynamic test instances. Interestingly there are not so many publicly available dynamic data sets: many authors use "self-generated" or "real-life" data sets, but do not explicitly provide these data sets for other authors. The same is true for the results: in many cases, only selected criteria of the objective function are reported. In addition, the reported criteria are often highly aggregated average values. Hence, these data are only partially applicable for comparison purposes.

Figure 3.8 summarizes the selected dynamic test sets. The first column includes the basic problems, the second column lists the associated sources, and the third column shows the underlying static sources, if those have been used to derive the dynamic instances.

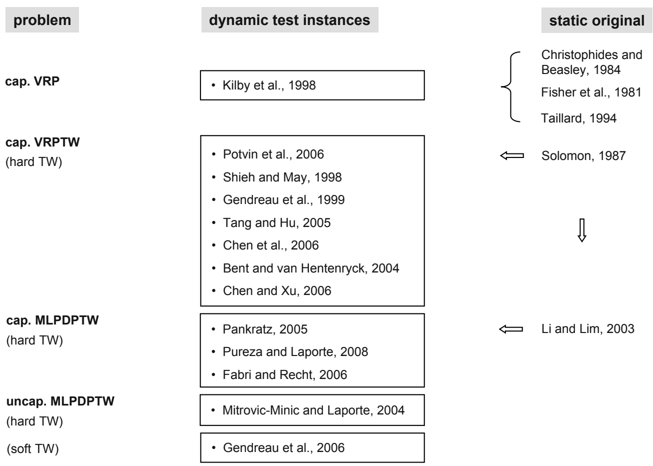


Figure 3.8: Dynamic test instances

Dynamic test data for the **capacitated VRP** were generated by Kilby et al. (1998): the authors take the static instances, published by Christophides and Beasley (1984), Fisher et al. (1981) and Taillard (1994) as a basis and extend them by Call-In times and service times for each task. The *Call-In times* are chosen according to a uniform random distribution throughout the whole planning horizon. The resulting dynamic data sets, however, were used only by a few authors, e.g. by Montemanni et al. (2005).[8]

For testing the **capacitated VRP with hard time windows**, many authors propose dynamic extensions to the test data published in Solomon (1987). Since the original data set evokes so much interest, a detailed summary of the original static data set is given, followed by seven approaches of bringing dynamism into the instances.

In the original 56 data sets, 100 customers are spread in a 100×100 unit square with varying geographical distributions: uniformly and randomly distributed (problem classes R1, R2), clustered (C1, C2), and semi-clustered (RC1, RC2), the last representing a mixture of uniformly and randomly distributed and clustered customers. The problem classes of

[8] The Kilby et al. (1998) data sets are no longer available at the author's homepage. Hence, we refer to http://www.fernuni-hagen.de/WINF/menuefrm/publik.htm. Here the data sets can be found in the folder "Montemanni et al. 2005".

type 1 possess narrow time windows at the depot, so that only a few customers can be serviced in each route; in contrast, problem classes of type 2 possess wider time windows at the depot. Travel times between the customers are taken to equal the corresponding distances, which are calculated as Euclidean distances. Each customer requires an individual service time (10 minutes for R1, R2, RC1, RC2 and 90 minutes for C1, C2). For servicing the customers, an unlimited number of homogeneous capacity constrained vehicles is available at a central depot. The vehicles have to return to the depot within the specified opening time (hard constraint).

The Solomon instances have been employed for tests of many new *static* algorithms, which have produced high quality and even optimal solutions. The availability of these solutions is an advantage when solution quality of "dynamized" Solomon instances has to be evaluated with *optimal static solutions.* For a discussion of advantages and disadvantages of this kind of "performance analysis", see Section 2.1.

Comparison of solution quality is also possible *by relative comparison of two dynamic approaches, applied to the same test data set.* Unfortunately, this is complicated here by the fact that every author generates his "own dynamic extension" to the Solomon instances. Thus, the *"same test data set"* assumption is no longer fulfilled. Subsequently, seven different ways of calculating the *Call-In time* for a specific request i are reported. The "U" denotes a uniform random distribution within a given interval.

- Call-In(i) = EPT(i) $\cdot$ U(0,1). *(Potvin et al., 2006)*
- Call-In(i) = MAX(0, LPT(i) - constant - U(0, LPT(i))). *(Shieh and May, 1998)*
- Call-In(i) = U(0, MIN(EPT(i), t_{i-1})), where t_{i-1} is the departure time from i's predecessor in the best known solution for the static problem. *(Gendreau et al., 1999)*
- Call-In(i) = U($c_1 \cdot$ EPT(i), $c_2 \cdot$ LPT(i)), where c_1 and c_2 ($0 \leq c_1 \leq c_2 \leq 1$) are two parameters. *(Tang and Hu, 2005)*
- Call-In(i) = MAX(0, EPT(i) - 1.5 $\cdot$ t_{Oi} - r), where t_{Oi} denotes the travel time between depot and node i, where r = U(0, EPT(i) - 1.5 $\cdot$ t_{Oi}). *(Chen et al., 2006)*
- Call-In(i) = U((k-1) $\cdot$ H/3, MIN(λ_i, k $\cdot$ $(H/3) - 1$), where k denotes an interval of the planning horizon H, where λ_i denotes the latest time a vehicle can depart from depot, service i and return to the depot. *(Bent and van Hentenryck, 2004)*
- Call-In(i) = U(0.5 $\cdot$ MIN(EPT(i), LPT - $d_i - \Delta - \tau$), MIN(EPT(i), LPT-$d_i - \Delta - \tau$)), with d_i denoting the travel time from depot to customer i, with Δ denoting the time between two consecutive decision epochs, with τ denoting the computational time. *(Chen and Xu, 2006)*

Obviously, there are many different ways of calculating a Call-In time.

In a next step, test data for the dynamic **capacitated MLPDPTW with hard time windows** are investigated. The available test sets are also based on the previous Solomon instances. A transformation of the static VRPTW data into static MLPDPTW data is accomplished by the following two authors: while Nanry and Barnes (2000) simply pair up the customers appearing in the routes of the best known Solomon VRPTW solutions *one* *by one* (regarding the "optimal" order), Li and Lim (2003) *randomly* pair up customer locations within routes of solutions obtained with their own heuristic solution approach. Both approaches render 100 VRPTW requests into 50 MLPDPTW requests.

Dynamic components have been added to the Li and Lim (2003) instances in three different ways:

- Pankratz (2005) first introduces the variable $t_r^{latest}(i)$, which is calculated as follows:

  $t_r^{latest}(i) = \text{MIN(LPT(i), LDT(i) - } t_{PD} - t_{Service}) - t_{Depot,P}$,

  where t_{PD} denotes the direct travel time from Pickup to Delivery of request i, where $t_{Service}$ denotes the service time at the Pickup location, and where $t_{Depot,P}$ denotes the direct travel time from depot to the Pickup location. Afterwards, based on $t_r^{latest}(i)$, dynamic instances are generated with the formula

  Call-In(i) $=$ a $\cdot\, t_r^{latest}(i)$,

  with a varying from 0.1 to 1.0 in steps of 0.1.

- A second approach is proposed by Pureza and Laporte (2008). They calculate

  Call-In(i) $=$ MIN(EPT(i), MAX(U(1,5), LPT - $t_{Depot,P} - \beta$)),

  where $U(1,5)$ denotes an integer number uniformly randomly generated between 1 and 5, where $t_{Depot,P}$ denotes the direct travel time from depot to the Pickup location of request i at time t=0 and where β is chosen to take one of the values 0, 100, 200, 300. According to the authors, the formula does not guarantee service since the Delivery location restrictions are not taken into account and in the case of time dependency, the travel time used in the computation belongs to the specific first period.

- A third option was chosen by Fabri and Recht (2006). They generate dynamic arrival times with the formula:

  Call-In(i) $=$ U(0, MIN(EPT(i), LPT(i) - $t_{Depot,P}$)).

In contrast to the previous dynamic test data, the following two publications for uncapacitated MLPDP test data do not rely on any available static instances.

Dynamic test data for the **uncapacitated MLPDP with hard time windows** were proposed by Mitrovic-Minic and Laporte (2004). The authors generate their own 40 test instances, based on real-life data from two courier companies operating in Vancouver (Canada). Up to 1000 requests (100% dynamic) occur in a 60km$\times$60km geographical area with Call-In time being calculated according to a uniform random distribution over the whole planning horizon.

Finally, dynamic data sets for the **uncapacitated MLPDP with soft time windows** published by Gendreau et al. (2006) have to be considered. The data generation process as well as the achieved results stand out due to very detailed and convenient description.

As this data set is used for benchmarking purposes of the later proposed algorithmic procedures, it is refered to Section 4.4.2 for further analysis.

At the end of this section, it can be noted that there are not many publicly available dynamic test instances. Most instances concentrate on local area VRP(TW) applications, followed by some local area MLPDPTW instances. No dynamic instances at all are available in the category of wide area transportation problems, especially for the SLPDPTW.

3.5 Acceptance of Dynamic Planning Applications in Real-Life

Most of the surveyed dynamic publications report "dramatic benefits" and "high cost reductions". However, most results are obtained in artificial test environments. In exceptional works, **Powell et al. (2000b)** and **Powell et al. (2002)** describe the challenging experience of transferring a dynamic planning algorithm for a wide area SLPDPTW into a running real-life application.

According to the authors, the real-life application of dynamic Decision Support Systems often (and especially in their case) does NOT result in dramatic benefits. The relatively small success of computer-based planning systems is particularly attributed to low "user compliance" (the rate with which a human dispatcher accepts the recommendation of a computer system), which is often below 60% in the truckload trucking area.

Reasons for this behavior can be discovered by comparing the different solution approaches of human dispatchers and of mathematical optimization systems. A human's decision is *state-action based*, producing *locally greedy optimizations* mostly neglecting the effects downstream in space and time. A mathematical model's decision is *cost minimization based*, producing a *global solution*.

Although producing a global solution (on a given data set), there are also some *drawbacks of the computer-based system.* The accurateness of the model's real world description may be limited, in particular in dynamic situations, when subproblems are solved at a point in time where availability of data is rather limited. In addition, the data in the computer may be generally imperfect or incomplete. A human planner may possess information (acquired by phone, conversation or visual inspection) that has not yet been entered into the system. "Implicit information", meaning general experience, for example based on historical events, may also be hardly available to a computer system.

Many decisions are fairly obvious, meaning that human and computer will coincide in these instances. Hence, a problem arises when the "higher reasoning" of the computer produces decisions that differ from the pattern-based reasoning of the dispatcher. Then the human has a dilemma: *is the discrepancy a result of "higher reasoning" or a simple data error?* Especially in *real time problems*, where fast decisions are elementary, it is not easy to find out if a computer's decision is plausible. Often the dispatcher will go with his own intuition.

The authors suggest a *hybrid approach*, generating a computer solution between "global" and "greedy", in order to improve solution acceptance by human dispatchers. They introduce a random variable, which decides if the computer-generated "truck-to-load assignment" is accepted by the human dispatcher (this is more likely if the computer solution exhibits more similarities to a greedy solution). Furthermore, a factor α is introduced, which represents the degree of global optimization: for full global optimization its value is 1, for full greedy optimization its value is 0, intermediate strategies are represented by α-values in the interval (0,1).

In several simulations with dynamic orders and dynamic travel times, the authors evaluate correlation of α and user compliance: when varying α for given compliance probabilities (100%, 70%, 40%), best results are achieved for α-level 0.75, interestingly even in the 100% compliance case.

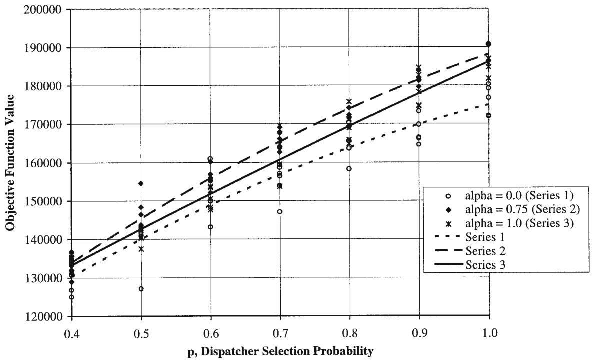


Figure 3.9: Impact of user-compliance and α on obj. function values (Powell et al., 2000b)

Figure 3.9 shows an original graph from Powell et al. (2000b), which includes the effects of varying user-compliance probabilities (x-axis) for given α's (1.0, 0.75, 0). The associated objective values are plotted on the y-axis. The authors discover that as the level of user compliance drops, the value of a globally optimal solution (α=1) over a greedy solution drops dramatically. On the other hand, if user compliance is high (over 90%), the value of global optimization (either $\alpha = 1.0$ or $\alpha = 0.75$) is quite high and demonstrates the usefulness of optimization models.

In a comparison, **Powell et al. (2002)** contrast the hypothetical case of perfect user compliance in global optimization (α-value = 1) with the case of perfect user compliance in completely "manual" planning (α-value = 0). They achieve *better results, in the range of 5% to 10%, when applying pure global optimization and assuming perfect user compliance.*

The problem of user acceptance of computer-generated solutions is also reported in Bell et al. (1983). The authors tackle the problem in a similar way as described in Powell et al. (2000b): they replicate human decision patterns. Neighboring customers are aggregated and treated as one single customer, which results in solutions similar to those the dispatcher is used to see. So the acceptance of computer solutions is increased. In addition, it is reported that once dispatchers felt comfortable with the system, most schedulers began to ask for size expansion of the neighborhoods to allow more options.

Chapter 4

Development and Evaluation of two Dynamic Planning Procedures

In this chapter two dynamic planning approaches are developed: an Insertion based procedure with Multiple Neighborhood Search (Section 4.1) and an Assignment based procedure (Section 4.2). Both procedures are directed - as an intermediate step on the way to the actual real-life planning situation - to the local area capacitated MLPDPTW, for which a detailed problem specification has been given in Section 2.3. In Section 4.3 the procedures' specific characteristics are compared, elaborating the main differences. Afterwards, some test data sets - self-generated as well as taken from the literature - are introduced (Section 4.4). These data sets are used for a comparison of the procedures' performance and also to gain some general insights to dynamic problems (Section 4.5). Finally, one procedure is chosen for adaptation to the actual real-life scenario (Section 4.6).

4.1 Multiple Neighborhood Search

The first procedure (coded in Eclipse 3.4.2 with Java version 1.6) basically consists of two components: *Best Insertion* and *Multiple Neighborhood Search (MNS)*. Best Insertion is used (i) to construct a feasible initial solution out of the available static orders, (ii) to incorporate new dynamically occurring orders as well as (iii) a basis for the improvement procedure. The improvement part is named "Multiple Neighborhood Search" since it investigates several structurally different neighborhoods in order to find solutions with better objective function values.

Figure 4.1 visualizes the *general program framework*: During the planning horizon, new orders arrive dynamically and have to be incorporated by the planning algorithm. At first, a new feasible solution is constructed by Best Insertion, followed by a run of the MNS component. When the replanning run is finished, new instructions are sent to the vehicle fleet in operation.

4.1.1 General Planning Process and Synchronization

Since "plan execution" and "replanning" run simultaneously, some rules for *synchronization* and for the *general planning process* have to be specified. Based on the idea of **rolling horizon planning**, the following approach was chosen (cp. Figure 4.2):

Steffen Schorpp, *Dynamic Fleet Management for International Truck Transportation,*
DOI 10.1007/978-3-8349-6675-9_4,

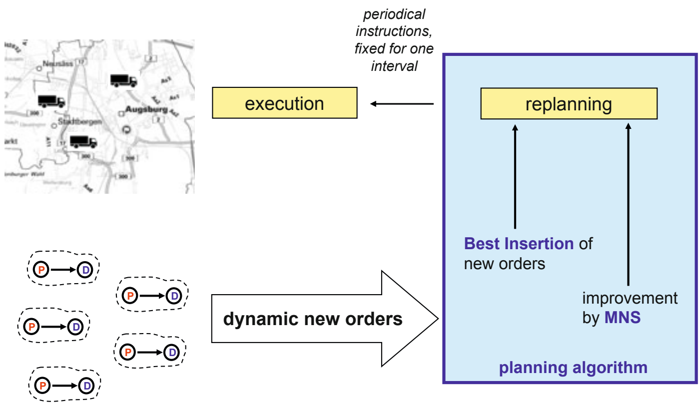


Figure 4.1: MNS: general program framework

The planning horizon is split into time intervals of equal length (here: 10 minutes). At the beginning of each interval, all decisions within the current interval are fixed. Then the current plan is transferred to the vehicles in execution, giving them planning certainty at least for the following 10 minutes. In a next step, it is checked whether new orders have arrived during the last time interval. If this is true, those newly arrived orders are incorporated by Best Insertion. In Figure 4.2, for example, there are three orders, A, B and C, that arrive in the time interval from 8:00 until 8:10. Accordingly, these orders are incorporated at the beginning of the following time interval (8:10 until 8:20) by Best Insertion.

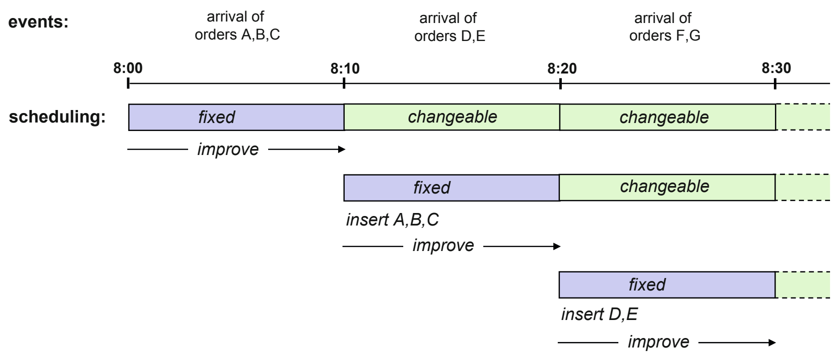


Figure 4.2: MNS: rolling horizon planning

The remaining time in the interval is completely used to run the improvement procedure. Both components, Best Insertion and MNS, have to observe the current interval's *fixed* decisions: changes in scheduling may be applied only in the subsequent intervals (denoted as *changeable*).

In the following, the term *fixed* is specified in more detail as ***time fixed*** and ***vehicle fixed***. The use of *fixed* up to now is synonymous with *time fixed*, which means a fixation of scheduled events, due to a preceding rolling horizon. In addition, some "dependent" parts of the schedule also have to be set to status *time fixed*: Let us suppose the case of time fixation of a departure event. Due to the basic problem specifications, no more re-scheduling is allowed until the associated target location is reached and serviced. Thus, the total scheduling time until servicing of this target location is completed, has to be set as *time fixed*. Therefore, in many cases the *time fixed* horizon will exceed the original 10 minutes.

An event is set to status *vehicle fixed* if it depends on irreversibly made decisions that allow for further changes in scheduling, but do not allow for the event's exchange to another vehicle. This especially covers the situation of an order's Delivery: If the associated Pickup is set as *time-fixed*, then the Delivery task is no longer allowed to be transferred to another vehicle. But nevertheless, it may be subject to re-scheduling within the current vehicle's tour.

If an event has the status *time fixed*, it is automatically *vehicle fixed*, but not vice versa. The difference of *time fixed* and *vehicle fixed* is of special interest for the improvement part.

4.1.2 Best Insertion

In a next step, the **Best Insertion strategy** is considered in detail: in the case of a newly arrived order (with Pickup and Delivery location), the program investigates for each vehicle the cost of all feasible insertion options. Finally, the best insertion option over all vehicles is chosen.

An example is given in Figure 4.3: A vehicle is traveling towards the time fixed Delivery location of order 1, hence there are no more changes allowed before arrival. Its current tour additionally includes the vehicle fixed Delivery of order 2 as well as the unfixed Pickup and Delivery of order 3. For inclusion of the new order's Pickup and Delivery locations, there are four possible positions: three between current tour locations and one at the end. Since Pickup and Delivery may be scheduled at different positions, there are 10 possible scheduling options.

Generally, there are $\frac{n \cdot (n+1)}{2}$ possible scheduling options, with n being the number of non *time fixed* positions. However, depending on vehicle capacity and order size, the number of investigated options can be considerably reduced by excluding infeasible cases.

Another important scheduling aspect is the use of a **waiting strategy** that prevents early arrivals and thus waiting at the target location. Instead, waiting time is scheduled at the current location. The vehicle departure time (and end of the waiting time) is calculated so that it exactly results in an arrival at EPT or EDT at the target location:

departure time (P) = EPT - travel time to Pickup location from current vehicle position
departure time (D) = EDT - travel time to Delivery location from current vehicle position

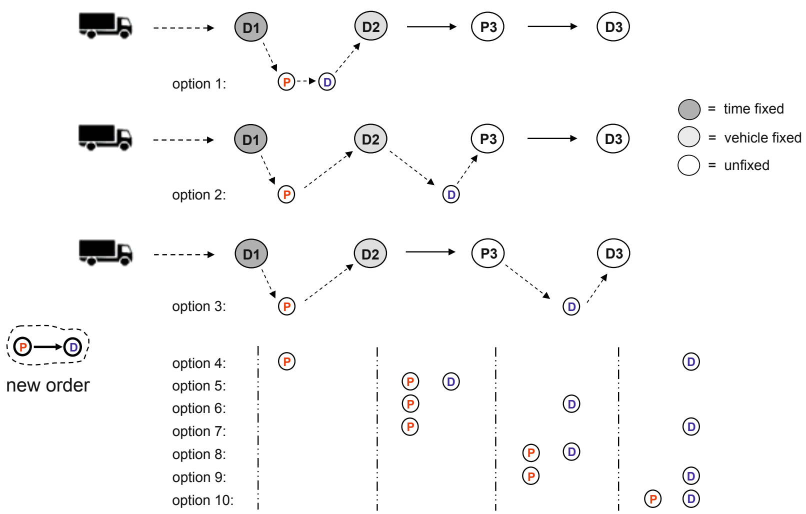


Figure 4.3: MNS: investigated insertion positions

This approach helps to postpone irreversible decisions (time fixation), hence generating more flexibility to react to new possible dynamic information (cp. Section 3.3.2). In cases in which immediate departure results in arrivals within the time window or even delayed, no waiting strategy is applied.

4.1.3 Improvement Neighborhoods

After generation of a feasible solution, the remaining time is used to run the **improvement procedure**, which applies multiple neighborhoods in an alternating manner: *λ-1 Interchange with Tabu List*, *Intraroute Optimal Sequence* and *Complete Solution Rebuild*. The frequency a specific neighborhood comes into operation has to be initially specified by percentage values.

In the following the main ideas behind these neighborhoods are outlined:

λ-1 Interchange with Tabu List selects two promising vehicle tours and investigates the complete λ-1 neighborhood (cp. Osman, 1993), which means that the advantageousness of all possible exchange operations according to the following scheme are evaluated, with the best one being finally chosen:

- an exchangeable request is extracted from each selected tour and re-inserted into the other vehicle's tour ("1 ⇔ 1 exchange"),
- an exchangeable request is extracted from the first tour only and re-inserted into the second vehicle's tour ("1 ⇒ 0 exchange"), and
- an exchangeable request is extracted from the second tour only and re-inserted into

the first vehicle's tour ("0 $\Leftarrow$ 1 exchange").

An illustration is given in Figure 4.4.

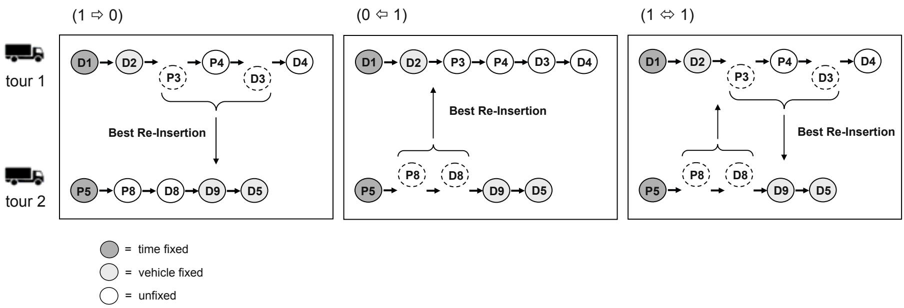


Figure 4.4: Illustration of neighborhood I: λ-1 interchange

The choice of vehicle pairs is performed in the following way: In a preprocessing step, a decreasing cost ranking for all vehicle tours is calculated. When choosing a pair of vehicle tours for exchange operations, preferably a *high-cost* and a *low-cost* vehicle are considered together. This increases the probability of achieving an improvement in objective function value by relieving the busy *high-cost* vehicle.

To avoid the recurring investigation of the same vehicle pairs, after each investigation the associated vehicle pair and the system time is stored in a Tabu List that blocks the vehicle pair for a prespecified time horizon *tabu_time*.

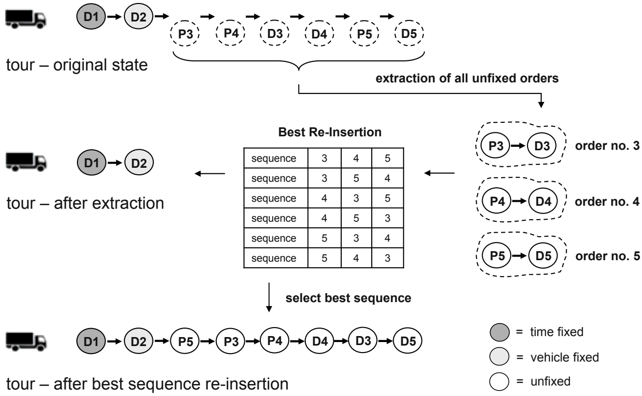


Figure 4.5: Illustration of neighborhood II: intraroute optimal sequence

The second neighborhood **Intraroute Optimal Sequence** extracts all exchangeable requests k within a vehicle's tour. Afterwards, these requests are re-inserted, examining all possible insertion sequences (permutations). Due to an increase in permutations with $k!$, the number of exchangeable requests must be limited: $k = 7$ turned out to be the maximum number that could be handled in acceptable computation time. The choice of vehicles is triggered again by a preprocessing step, in which all vehicles' tours are sorted according to their cost value. The procedure starts with the highest cost vehicle tour. The general idea is visualized in Figure 4.5.

The third neighborhood **Complete Solution Rebuild** does not consider only one or two vehicle tours for exchange operations, it considers all tours. In a first step, the procedure runs through all vehicle tours and extracts every exchangeable request. Afterwards, the exchangeable requests are re-inserted successively at the best insertion positions calculated over all vehicle tours. Again, in a preprocessing step, vehicle tours are ordered according to their cost values, which are used to generate a re-insertion sequence beginning with requests from expensive tours. The approach is illustrated in Figure 4.6.

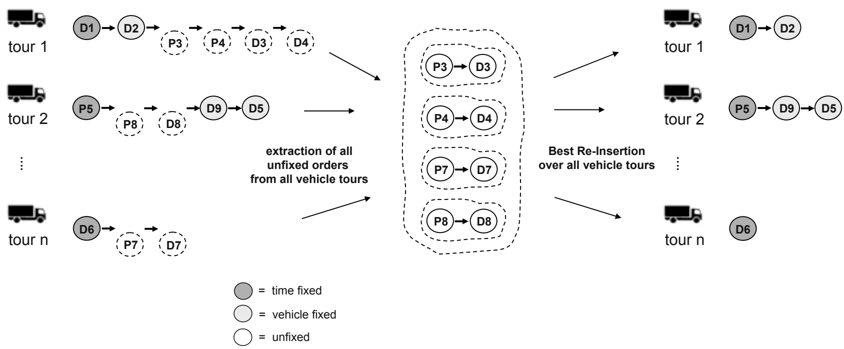


Figure 4.6: Illustration of neighborhood III: complete solution rebuild

Pseudocode notations of all three neighborhoods are given in Appendix A.

Now, all basic components and main ideas of the insertion based MNS procedure have been outlined. In the following, the focus is set on the actual implementation and on the interaction of the specific components. This can be performed best in the context of the utilized simulation framework.

4.1.4 Simulation Framework

Best Insertion and MNS improvement are embedded into a simulation framework that works as follows (cp. Program flow chart, in Figure 4.7):

In a first step, all *input data is read from an Excel file* and stored in the appropriate data classes (vehicles, orders, parameters, etc.). The Excel file contains several worksheets. The structure and contents of the most important *order* and *vehicle* worksheets are explained in the following:

- Table 4.1 shows the typical structure of an order worksheet. The first line indicates

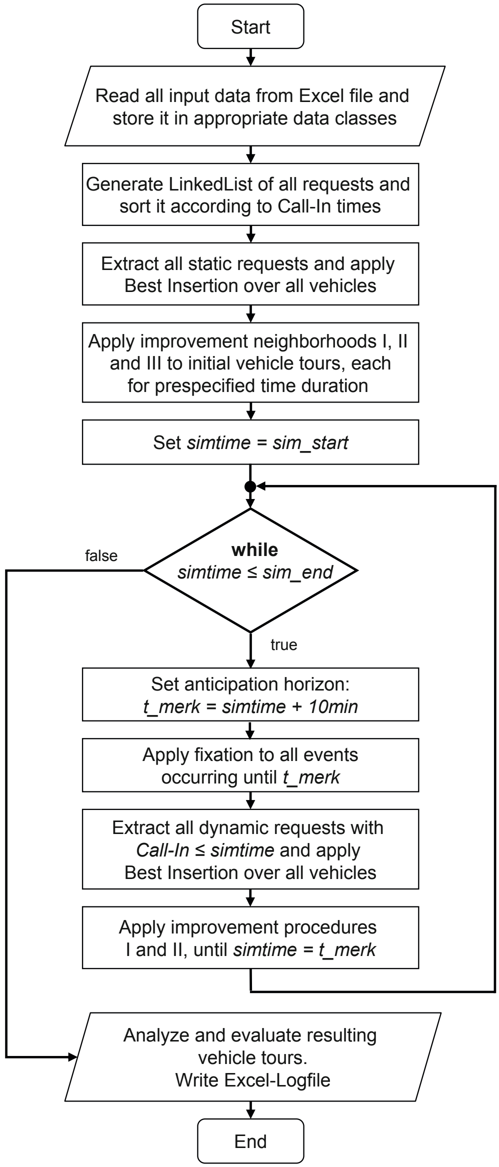


Figure 4.7: Program flow chart: MNS simulation framework

the total *number of orders* to be read, afterwards each line contains a unique order identifier (*no.*), followed by the order attributes *Call-In*, Pickup location (*PL*), Delivery location (*DL*), time window characteristics (*EPT*, *LPT*, *EDT*, *LDT*), required capacity (*weight*, *volume*), *loadtime*, and *unloadtime*.

number of orders:	1000									in min	in min
no.	Call-In	PL	DL	EPT	LPT	EDT	LDT	weight	volume	loadtime	unloadtime
1	09:00	756	356	09:30	10:15	09:45	11:15	1	1	2	2
2	09:00	989	35	09:30	10:15	09:45	11:15	1	1	2	2
3	09:00	504	99	09:30	10:15	09:45	11:15	1	1	2	2
4	09:01	18	275	09:31	10:16	09:46	11:16	1	1	2	2
5	09:02	415	535	09:32	10:17	09:47	11:17	1	1	2	2
6	09:02	474	488	09:32	10:17	09:47	11:17	1	1	2	2
7	09:03	879	264	09:33	10:18	09:48	11:18	1	1	2	2
8	09:03	766	819	09:33	10:18	09:48	11:18	1	1	2	2
9	09:03	416	161	09:33	10:18	09:48	11:18	1	1	2	2
10	09:03	781	822	09:33	10:18	09:48	11:18	1	1	2	2
11	09:04	5	546	09:34	10:19	09:49	11:19	1	1	2	2
12	09:04	80	311	09:34	10:19	09:49	11:19	1	1	2	2
13	09:05	573	968	09:35	10:20	09:50	11:20	1	1	2	2
14	09:05	320	231	09:35	10:20	09:50	11:20	1	1	2	2
15	09:06	155	442	09:36	10:21	09:51	11:21	1	1	2	2
16	09:06	208	487	09:36	10:21	09:51	11:21	1	1	2	2
17	09:06	52	399	09:36	10:21	09:51	11:21	1	1	2	2
18	09:07	181	762	09:37	10:22	09:52	11:22	1	1	2	2
19	09:07	88	10	09:37	10:22	09:52	11:22	1	1	2	2
20	09:07	585	112	09:37	10:22	09:52	11:22	1	1	2	2
21	09:09	266	729	09:39	10:24	09:54	11:24	1	1	2	2
22	09:09	697	292	09:39	10:24	09:54	11:24	1	1	2	2
23	09:09	768	742	09:39	10:24	09:54	11:24	1	1	2	2
24	09:09	650	263	09:39	10:24	09:54	11:24	1	1	2	2

Table 4.1: Excel input file: orders

- Table 4.2 visualizes the typical structure of a vehicle worksheet: the first line indicates the total *number of* (available) *vehicles*, afterwards each line contains a unique vehicle identifier (*no.*), followed by the *depot location*, the vehicle capacity with regard to weight (*cap. weight*) and volume (*cap. volume*), as well as information on vehicle availability (*available from*, *available to*).

number of vehicles:	50				
no.	depot location	cap. weight	cap. volume	available from	available to
1	1001	3	3	09:00	19:00
2	1001	3	3	09:00	19:00
3	1001	3	3	09:00	19:00
4	1001	3	3	09:00	19:00
5	1001	3	3	09:00	19:00
6	1001	3	3	09:00	19:00
7	1001	3	3	09:00	19:00
8	1001	3	3	09:00	19:00
9	1001	3	3	09:00	19:00
10	1001	3	3	09:00	19:00
11	1001	3	3	09:00	19:00
12	1001	3	3	09:00	19:00

Table 4.2: Excel input file: vehicles

After reading this data, in a next step all available *orders are sorted according to their Call-In time* and stored in a LinkedList. All *static requests* with Call-In time before the official start of the planning horizon *sim_start* are *extracted* from this LinkedList *and are inserted* into the best positions over all vehicles, thus generating an initial feasible plan. Then all three neighborhoods of the MNS *improvement procedure* are applied during a prespecified calculation time.

After termination of this initial phase, the *actual dynamic simulation is started*. The simulation time that is generated in the program class *clock* is set to the given initial value *sim_start* and runs with a prespecified simulation speed s.

The dynamic simulation reflects the rolling horizon planning that is visualized in Figure 4.2: According to the anticipation horizon (here: 10 minutes), the time step *t_merk* is set to *simtime + 10 minutes*. Afterwards, *fixation is applied* to all events up to *t_merk*, as well as to all dependent activities scheduled later in time. Then, all *dynamic orders* with *Call-In time* $\leq$ *simtime* are extracted from the LinkedList and are *incorporated with Best Insertion*.

Finally, the remaining time up to *t_merk* is used to apply the MNS *improvement procedure* to the new feasible solution. In some pre-tests with the test instances of Section 4.4 *Complete Solution Rebuild* required more than 10 minutes for the generation of a new feasible solution, therefore, in the dynamic program part only the improvement neighborhoods *λ-1 Interchange with Tabu List* and *Intraroute Optimal Sequence* come into operation.

The described steps are iterated until simulation time reaches the time step *sim_end*, which does not coincide with the latest Call-In time: the latest Call-In time is only a lower bound for *sim_end*, in order to make sure that all dynamic requests have been processed. However, since plan execution continues beyond that time, further simulation time results in the investigation of additional improvement options and should result in a better overall solution.

The simulation ends with a *final analysis:* the generated vehicle tours that include all static and dynamic orders are evaluated with the *results being written into an Excel file.* This file includes the following information:

- The first worksheet (cp. Table 4.3) contains some *general information* (investigated test data file, utilized computing time) and *summarized results* for the different objective function criteria (travel time, waiting time, delay, and overtime).

  In addition, the values of the objective function criteria are broken down into several sources, in order to allow for a more detailed investigation: travel time is split into *travel time to Pickup*, *travel time to Delivery* and *travel time back to depot*. Waiting time is apportioned into *waiting empty for return to depot*, *waiting empty otherwise* and *waiting loaded*. Delay is broken down into *delay at Pickup* and *delay at Delivery* locations. As further information, the percentage values of different vehicle activities during operating time are given, as well as average travel time to Pickup and Delivery locations.

- The second worksheet consists of a *detailed vehicle scheduling for each vehicle*. Table 4.4 exemplarily shows the scheduling results for a single vehicle over a planning horizon of approximately 9 hours. Vehicle activity is explained by the columns *activity-log*, *time interval* and *way*.

  In addition, for each approached location, the associated time window and the actual arrival time is given. In the following three columns, the scheduled activity times are assigned to one of the groups: *waiting*, *traveling* and *loading*. Afterwards, potential *delay* is calculated, followed by two columns including information about the vehicle's capacity status *weight* and *volume*. At the end, some supplemental information like capacity utilization and vehicle utilization over the whole simulated planning horizon – *temporal utilization (fraction of traveling and loading)* – is stated.

test data file:	3_40dyn-glv_TW45_Ab30_stat4h_OM2_ev.xls
computation time (in min):	660
total travel time (in min):	19185
travel time to Pickup (in min):	7486
travel time to Delivery (in min):	10999
travel time back to depot (in min):	700
total waiting time (in min):	3248
waiting empty for return to depot (in min):	112
waiting empty otherwise (in min):	2044
waiting loaded (in min):	1092
total delay (in min):	39
delay Pickup (in min):	19
delay Delivery (in min):	20
total overtime (in min):	0

vehicle activity:		
operating time (in min):	26433	
traveling (in min):	19185	72.57 %
waiting (in min):	3248	12.28 %
loading (in min):	4000	15.13 %
further calculations:		
avg. travel time to Pickup (in min):	7:29	
avg. travel time to Delivery (in min):	10:59	

Table 4.3: Excel output file: general summary

vehicle	activity-log	time interval	way	time window	arrival	waiting	traveling	loading	delay	weight	volume
5	waiting in location 1001	09:00:00 -- 09:28:49				28:49					
	traveling to Pickup of order no. 309	09:28:49 -- 09:37:00	1001 -- 325	09:37:00 -- 10:22:00	09:37:00		8:11	2:00		1,00	1,00
	waiting in location 325	09:39:00 -- 09:40:57				1:57					
	traveling to Pickup of order no. 50	09:40:57 -- 09:42:00	325 -- 497	09:42:00 -- 13:42:00	09:42:00		1:03	2:00		2,00	2,00
	traveling to Pickup of order no. 323	09:44:00 -- 09:57:58	497 -- 461	09:45:00 -- 10:30:00	09:57:58		13:58	2:00		3,00	3,00
	traveling to Delivery of order no. 323	09:59:58 -- 10:24:27	461 -- 8	10:00:00 -- 11:30:00	10:24:27		24:29	2:00		2,00	2,00
	waiting in location 8	10:26:27 -- 10:57:58				31:31					
	traveling to Pickup of order no. 155	10:57:58 -- 11:03:00	8 -- 530	11:03:00 -- 15:03:00	11:03:00		5:02	2:00		3,00	3,00
	traveling to Delivery of order no. 309	11:05:00 -- 11:06:47	530 -- 152	09:52:00 -- 11:22:00	11:06:47		1:47	2:00		2,00	2,00
	waiting in location 152	11:08:47 -- 11:12:31				3:44					
	traveling to Delivery of order no. 50	11:12:31 -- 11:19:22	152 -- 720	09:57:00 -- 13:57:00	11:19:22		6:51	2:00		1,00	1,00
	traveling to Pickup of order no. 161	11:21:22 -- 11:24:50	720 -- 358	11:09:00 -- 15:09:00	11:24:50		3:28	2:00		2,00	2,00
	traveling to Pickup of order no. 470	11:26:50 -- 11:31:09	358 -- 442	11:24:00 -- 12:09:00	11:31:09		4:19	2:00		3,00	3,00
	traveling to Delivery of order no. 470	11:33:09 -- 11:46:49	442 -- 28	11:39:00 -- 13:09:00	11:46:49		13:40	2:00		2,00	2,00
	traveling to Pickup of order no. 510	11:48:49 -- 11:56:38	28 -- 616	11:50:00 -- 12:35:00	11:56:38		7:49	2:00		3,00	3,00
	traveling to Delivery of order no. 155	11:58:38 -- 12:09:40	616 -- 352	11:18:00 -- 15:18:00	12:09:40		11:02	2:00		2,00	2,00
	traveling to Pickup of order no. 512	12:11:40 -- 12:15:42	352 -- 16	11:51:00 -- 12:36:00	12:15:42		4:02	2:00		3,00	3,00
	traveling to Delivery of order no. 510	12:17:42 -- 12:45:43	16 -- 440	12:05:00 -- 13:35:00	12:45:43		28:01	2:00		2,00	2,00
	traveling to Delivery of order no. 161	12:47:43 -- 12:52:24	440 -- 712	11:24:00 -- 15:24:00	12:52:24		4:41	2:00		1,00	1,00
	traveling to Pickup of order no. 597	12:54:24 -- 12:58:10	712 -- 738	12:44:00 -- 13:29:00	12:58:10		3:46	2:00		2,00	2,00
	traveling to Delivery of order no. 512	13:00:10 -- 13:03:49	738 -- 928	12:06:00 -- 13:36:00	13:03:49		3:39	2:00		1,00	1,00
	traveling to Pickup of order no. 602	13:05:49 -- 13:08:25	928 -- 897	12:46:00 -- 13:31:00	13:08:25		2:36	2:00		2,00	2,00
	traveling to Pickup of order no. 617	13:10:25 -- 13:13:17	897 -- 635	13:00:00 -- 13:45:00	13:13:17		2:52	2:00		3,00	3,00
	traveling to Delivery of order no. 602	13:15:17 -- 13:20:07	635 -- 958	13:01:00 -- 14:31:00	13:20:07		4:50	2:00		2,00	2,00
	traveling to Pickup of order no. 73	13:22:07 -- 13:24:20	958 -- 552	09:57:00 -- 13:57:00	13:24:20		2:13	2:00		3,00	3,00
	traveling to Delivery of order no. 73	13:26:20 -- 13:35:28	552 -- 383	10:12:00 -- 14:12:00	13:35:28		9:08	2:00		2,00	2,00
	traveling to Pickup of order no. 641	13:37:28 -- 13:49:50	383 -- 549	13:24:00 -- 14:09:00	13:49:50		12:22	2:00		3,00	3,00
	traveling to Delivery of order no. 597	13:51:50 -- 14:10:29	549 -- 254	12:59:00 -- 14:29:00	14:10:29		18:39	2:00		2,00	2,00
	traveling to Delivery of order no. 617	14:12:29 -- 14:23:44	254 -- 209	13:15:00 -- 14:45:00	14:23:44		11:15	2:00		1,00	1,00
	traveling to Pickup of order no. 173	14:25:44 -- 14:27:46	209 -- 63	11:21:00 -- 15:21:00	14:27:46		2:02	2:00		2,00	2,00
	traveling to Pickup of order no. 153	14:29:46 -- 14:32:41	63 -- 5	11:01:00 -- 15:01:00	14:32:41		2:55	2:00		3,00	3,00
	traveling to Delivery of order no. 173	14:34:41 -- 14:41:35	5 -- 189	11:36:00 -- 15:36:00	14:41:35		6:54	2:00		2,00	2,00
	traveling to Delivery of order no. 641	14:43:35 -- 14:55:14	189 -- 237	13:39:00 -- 15:09:00	14:55:14		11:39	2:00		1,00	1,00
	traveling to Pickup of order no. 212	14:57:14 -- 15:02:33	237 -- 685	11:53:00 -- 15:53:00	15:02:33		5:19	2:00		2,00	2,00
	traveling to Delivery of order no. 153	15:04:33 -- 15:07:24	685 -- 664	11:16:00 -- 15:16:00	15:07:24		2:51	2:00		1,00	1,00
	traveling to Pickup of order no. 748	15:09:24 -- 15:16:03	664 -- 648	14:37:00 -- 15:22:00	15:16:03		6:39	2:00		2,00	2,00
	traveling to Delivery of order no. 748	15:18:03 -- 15:19:52	648 -- 452	14:52:00 -- 16:22:00	15:19:52		1:49	2:00		1,00	1,00
	traveling to Delivery of order no. 212	15:21:52 -- 15:26:45	452 -- 704	12:08:00 -- 16:08:00	15:26:45		4:53	2:00		0,00	0,00
	waiting in location 704	15:28:45 -- 15:59:07				30:22					
	traveling to Pickup of order no. 876	15:59:07 -- 16:06:00	704 -- 854	16:06:00 -- 16:51:00	16:06:00		6:53	2:00		1,00	1,00
	traveling to Pickup of order no. 879	16:08:00 -- 16:13:38	854 -- 711	16:08:00 -- 16:53:00	16:13:38		5:38	2:00		2,00	2,00
	traveling to Delivery of order no. 879	16:15:38 -- 16:28:59	711 -- 346	16:23:00 -- 17:53:00	16:28:59		13:21	2:00		1,00	1,00
	traveling to Pickup of order no. 882	16:30:59 -- 16:39:20	346 -- 755	16:10:00 -- 16:55:00	16:39:20		8:21	2:00		2,00	2,00
	traveling to Pickup of order no. 881	16:41:20 -- 16:47:52	755 -- 25	16:09:00 -- 16:54:00	16:47:52		6:32	2:00		3,00	3,00
	traveling to Delivery of order no. 876	16:49:52 -- 16:56:49	25 -- 354	16:21:00 -- 17:51:00	16:56:49		6:57	2:00		2,00	2,00
	traveling to Pickup of order no. 901	16:58:49 -- 17:01:37	354 -- 519	16:25:00 -- 17:10:00	17:01:37		2:48	2:00		3,00	3,00
	traveling to Delivery of order no. 882	17:03:37 -- 17:10:47	519 -- 731	16:25:00 -- 17:55:00	17:10:47		7:10	2:00		2,00	2,00
	traveling to Pickup of order no. 947	17:12:47 -- 17:23:49	731 -- 746	16:56:00 -- 17:41:00	17:23:49		11:02	2:00		3,00	3,00
	traveling to Delivery of order no. 881	17:25:49 -- 17:43:50	746 -- 824	16:24:00 -- 17:54:00	17:43:50		18:01	2:00		2,00	2,00
	traveling to Delivery of order no. 901	17:45:50 -- 18:05:30	824 -- 133	16:40:00 -- 18:10:00	18:05:30		19:40	2:00		1,00	1,00
	traveling to Delivery of order no. 947	18:07:30 -- 18:24:32	133 -- 369	17:11:00 -- 18:41:00	18:24:32		17:02	2:00		0,00	0,00
	traveling back to depot location 1001	18:26:32 -- 18:37:57	369 -- 1001	-- 19:00:00	18:37:57		11:25				
						-------	-------	-------	-------		
	temporal utilization:	**83.32 %**				**96:23**	**389:34**	**92**			
	(fraction of traveling and loading)					**waiting**	**traveling**	**loading**	**delay**		

Table 4.4: Excel output file: vehicle scheduling

- The third worksheet (cp. Table 4.5) includes a view of the *planning results from an order based perspective*. For each order, some given facts, like *Call-In* time and time window information (EPT, LPT, EDT, LDT) is replicated, combined with the planning results: assigned *vehicle*, actual Pickup time (PT) and actual Delivery time (DT). In addition, the resulting delays at Pickup and Delivery location are calculated.

order no.	Call-In	vehicle	EPT	LPT	PT	delay P	EDT	LDT	DT	delay D
299	08:00	43	13:00	17:00	13:00:00		13:15	17:15	14:00:25	
300	08:00	16	13:00	17:00	14:26:47		13:15	17:15	14:55:46	
301	09:01	3	09:31	10:16	09:31:00		09:46	11:16	10:59:37	
302	09:04	14	09:34	10:19	10:17:36		09:49	11:19	11:11:22	
303	09:05	2	09:35	10:20	10:08:34		09:50	11:20	10:20:22	
304	09:05	36	09:35	10:20	09:59:37		09:50	11:20	10:05:26	
305	09:05	10	09:35	10:20	09:35:00		09:50	11:20	09:55:51	
306	09:06	13	09:36	10:21	09:36:00		09:51	11:21	10:23:52	
307	09:06	1	09:36	10:21	09:55:26		09:51	11:21	10:41:36	
308	09:06	3	09:36	10:21	10:11:50		09:51	11:21	10:37:04	
309	09:07	5	09:37	10:22	09:37:00		09:52	11:22	11:06:47	
310	09:07	4	09:37	10:22	10:17:03		09:52	11:22	10:44:18	
311	09:08	17	09:38	10:23	09:38:00		09:53	11:23	11:22:20	
312	09:09	35	09:39	10:24	10:12:21		09:54	11:24	11:00:54	
313	09:11	1	09:41	10:26	10:15:22		09:56	11:26	10:25:36	
314	09:11	19	09:41	10:26	10:10:47		09:56	11:26	11:21:35	
315	09:11	19	09:41	10:26	10:03:53		09:56	11:26	10:40:48	

Table 4.5: Excel output file: planning results from an order based perspective

Now, the general idea and the planning process of the Multiple Neighborhood Search procedure have been explained. Subsequently, a second planning approach that is based on completely different concepts is presented.

4.2 Assignment Based Procedure

The basic idea of the Assignment based procedure (coded in Eclipse 3.4.2 with Java version 1.6) was proposed in Fleischmann et al. (2004) for a local area SLPDPTW. In the following, the original approach is extended for the multi load case.

The procedure's main feature is to trigger an order-to-vehicle assignment by the result of a classical bipartite assignment problem. This allows for a simultaneous consideration of all vehicles V and all open orders O. The objective is to minimize the overall costs of carrying out all requested transportation tasks.

Every re-planning run has to be prepared in such a way that the underlying problem can be solved for an equal number of n orders and n vehicles. Since the number of vehicles and orders will usually not be identical and in order to allow for vehicle waiting and order postponement, some *dummy orders* ($o \in O^d$, denoting the set of dummy orders; with $|O^d| = |V|$) and *dummy vehicles* ($v \in V^d$, denoting the set of dummy vehicles; with $|V^d| = |O|$) are introduced.

4.2.1 Bipartite Assignment Problem

Hence, a bipartite assignment problem has to be solved over all vehicles $v \in \{V \cup V^d\}$ and all orders $o \in \{O \cup O^d\}$. The assignment costs for each order-vehicle pair are calculated as c_{vo}.

A problem formulation is given as follows:

Model of the Bipartite Assignment Problem:

data:

V = set of real vehicles
V^d = set of dummy vehicles
O = set of real orders
O^d = set of dummy orders
c_{vo} = assignment cost of vehicle v to order o

variables:

$$x_{vo} = \begin{cases} 1, & \text{if vehicle v is assigned to order o} \\ 0, & \text{otherwise} \end{cases}$$

objective function:

$$minimize \quad \sum_{v \in \{V \cup V^d\}} \sum_{o \in \{O \cup O^d\}} c_{vo}\, x_{vo}$$

s.t.

$$\sum_{v \in \{V \cup V^d\}} x_{vo} = 1 \qquad \forall\, o \in \{O \cup O^d\}$$

$$\sum_{o \in \{O \cup O^d\}} x_{vo} = 1 \qquad \forall\, v \in \{V \cup V^d\}$$

$$x_{vo} \in \{0, 1\} \qquad \forall\, v \in \{V \cup V^d\}, \forall\, o \in \{O \cup O^d\}$$

The given problem formulation is recurringly solved by the exact procedure proposed in Jonker and Volgenant (1987). Even in the case of major problem sizes, the calculation time is far less than a second.

A resulting assignment of a real order to a real vehicle becomes effective immediately (in the case of a waiting vehicle) or after completion of the vehicle's current trip (in the case of a traveling vehicle) and results in a trip to the order's Pickup location. As in the MNS procedure, early arrival (before the destination's time window has opened) is prevented by scheduling of some waiting time at the current location.

The length of such a waiting time is calculated identically to the previous procedure so as to achieve exact arrival at the destination's time window opening. Therefore, the real order to real vehicle assignment (result of the bipartite assignment problem) is not fixed until the actual departure of the vehicle has been carried out, thus allowing for re-assignment during such a waiting period.

4.2.2 Assignment Matrix

The underlying assignment matrix contains four types of possible assignments:

- sector I: $v \in V \land o \in O^d$ *assignment of real vehicle and dummy order*

→ waiting empty or execution of next scheduled Delivery (if available)

- sector II: $v \in V \wedge o \in O$ *assignment of real vehicle and real order*
  → start traveling to real order's Pickup at next time of availability
- sector III: $v \in V^d \wedge o \in O^d$ *assignment of dummy vehicle and dummy order*
  → no impact
- sector IV: $v \in V^d \wedge o \in O$ *assignment of dummy vehicle and real order*
  → postponement of real order

The matrix configuration and the associated impact of possible assignments in the different sectors is visualized in Figure 4.8. The size of the first sector is stable since the number of real vehicles and dummy orders is not subject to changes ($|V| = |O^d|$). The other sectors' sizes increase if a new order occurs, and decrease if an order is removed due to an ultimate assignment (*departure to Pickup location*). Generally, the relation $|V^d| = |O|$ has to be maintained in order to keep the total matrix quadratic. Hence, a newly occurring order does not only result in an additional "real order" column but also in an additional "dummy vehicle" row; the departure to a Pickup location does not only result in the deletion of the associated "real order" column but also in the deletion of a "dummy vehicle" row.

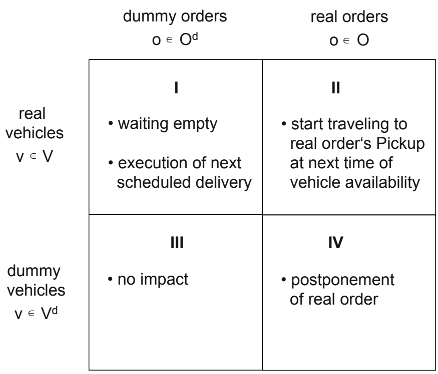


Figure 4.8: Assignment matrix: general configuration and impact of assignment

In the following, the meaning and impact of assignments in different matrix sectors is discussed in detail. Furthermore, it is explained how the respective cost values c_{vo} are chosen (cp. Figure 4.9):

Sector I

The assignment of a dummy order to a real vehicle may result in two possible effects on the real vehicle: if there are further tasks (Delivery locations) in the vehicle's current schedule, these *tasks are simply executed as planned.* Otherwise, if there are no more tasks available, the vehicle has to *wait empty* at its current location. Such an assignment occurs in situations when the number of real orders is smaller than the number of real vehicles, and also when available orders possess far distant time windows, thus being postponed.

Cost values for *further execution of scheduled Delivery tasks* are set to zero, while *waiting empty* is penalized with a given parameter *c_empty*.

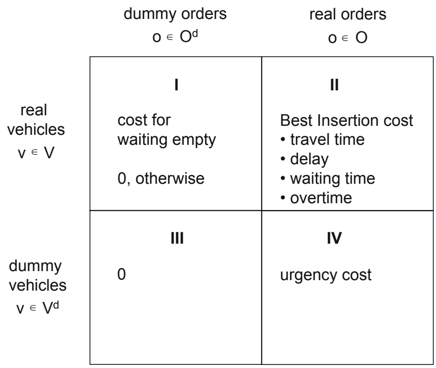


Figure 4.9: Assignment matrix: choice of cost values

Sector II

Here, a real order is assigned to a real vehicle. This implies that the *real vehicle starts traveling to the Pickup location of the assigned real order* at the next time of availability. A *waiting* vehicle is available immediately, while a *traveling* vehicle reaches the status *available* for the next time when it arrives at its current destination. As already mentioned, the departure and hence the ultimate order-to-vehicle assignment may be delayed by some waiting time if immediate departure results in waiting time at the destination.

Cost values c_{vo} are chosen according to the costs that result from inserting the order o at the best position of vehicle v's scheduling. These costs are calculated with respect to the overall objective function, including weighted penalty costs for *travel time*, *delay*, *waiting*, and *overtime*.

The applied **Best Insertion** procedure is illustrated in Figure 4.10. It differs slightly from the Best Insertion applied at the MNS procedure, since it only allows for a Pickup's insertion at the first (*unfixed*) position. The associated Delivery, however, may be placed at every subsequent position as long as this complies with the capacity constraints. In Figure 4.10 there is one fixed event (arrival at Delivery 1), so the new order's Pickup is inserted right behind that Delivery. The further exemplary schedule includes three more Deliveries (D2, D4 and D3), which induces four possible insertion positions for the new Delivery: directly after the new Pickup, after D2, after D4, and after D3.

Generally, the number of investigated insertion positions is significantly lower than in the previous MNS case. In total, only $n + 1$ positions have to be checked, with n being the number of unfixed scheduled locations.

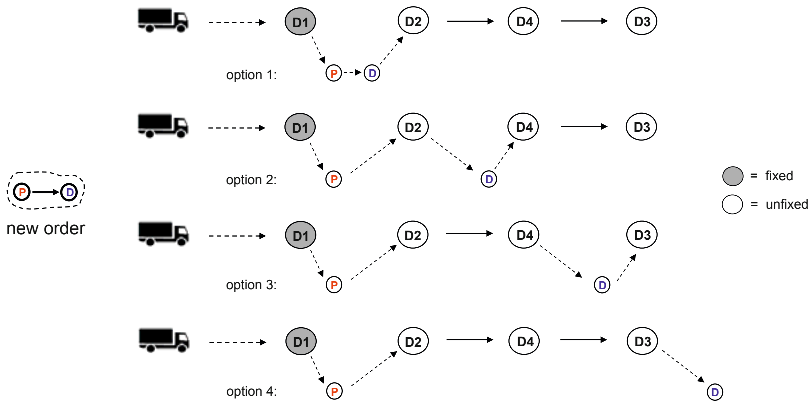


Figure 4.10: Assignment based procedure: investigated insertion positions

Sector III

The existence of this assignment type is a direct consequence of the demanded flexibility. In order to generate the three other assignment sectors for every planning run, an appropriate size of the matrix is required, including dummy columns and dummy rows at any time. The associated assignment cost are set to zero.

Sector IV

Results in this sector indicate the assignment of a dummy vehicle to a real order, which can be interpreted as the order's postponement. Such an assignment decision can have two possible reasons. On the one hand, there may be less real vehicles available than real orders. On the other hand, the order's time window may lie in the far distant future, so that a current assignment in sector II would just produce extensive waiting time.

The associated costs $c\_urgency_o$ are individually calculated for each real order o according to the following formula:

$$c\_urgency_o = \begin{cases} delta_{min,o} - 1 & \text{if } slack\_EPT_o > \text{a} \\ delta_{max,o} + (b - slack\_LPT_o)^2 & \text{if } slack\_EPT_o \leq 0 \ \&\ tw\_width_o < \text{c} \\ delta_{median,o} & \text{otherwise} \end{cases}$$

Before the meaning of these three options is explained, the calculation of the involved variables $slack\_EPT_o$, $slack\_LPT_o$, tw_width_o, $delta_{max,o}$, $delta_{min,o}$, $delta_{median,o}$ is described (a, b and c are parameters):

(i) $slack\_EPT_o = EPT_o - avg.\ travel\,time\ to\ Pickup - simtime$

(ii) $slack\_LPT_o = LPT_o - avg.\ travel\,time\ to\ Pickup - simtime$

$$\text{(iii)} \quad tw\_width_o = \begin{cases} \frac{LPT_o - EPT_o}{avg.\,travel\,time\,to\,Pickup}, & \text{if simtime} \leq EPT_o \\ \frac{LPT_o - simtime}{avg.\,travel\,time\,to\,Pickup}, & \text{otherwise.} \end{cases}$$

$$\text{(iv)} \quad delta_{max,o} = max\,\{c_{vo} - c_{vo'},\, v \in V\}, \quad with\ o' = |O^d|$$

$$\text{(v)} \quad delta_{min,o} = min\,\{c_{vo} - c_{vo'},\, v \in V\}, \quad with\ o' = |O^d|$$

$$\text{(vi)} \quad delta_{median,o} = median\,\{c_{vo} - c_{vo'},\, v \in V\}, \quad with\ o' = |O^d|$$

Variables (i) $slack\_EPT_o$ and (ii) $slack\_LPT_o$ measure the absolute temporal gap from current simulation time to the beginning and ending of the Pickup time window of order o, respectively. The calculation includes the average travel time to a Pickup location, since the actual travel time to the specific Pickup location cannot be calculated exactly (it depends on the vehicle to which the postponed order o will be assigned later on).

Variable (iii) tw_width_o is a measure for the remaining width of the Pickup time window, relative to the average travel time to a Pickup location. The calculation is visualized in Figure 4.11. In the first case, the current simulation time is before EPT, hence the total time window from EPT until LPT is still available. In the example, the average travel time to Pickup fits six times into the remaining time window slot ($tw\_width = 6$). In the second case, current simulation time exceeds EPT, hence the *remaining time window width* is shorter ($tw\_width = 4$).

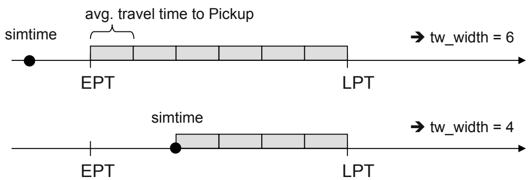


Figure 4.11: Exemplary calculation of *tw_width*

Variables (iv) and (v) are determined in order to "enforce" or to "prevent" a real order's sector II assignment, respectively. If the cost value for order o in sector IV is chosen as greater than $delta_{max,o}$, a certain assignment in sector II is enforced, at least in situations with a higher (or equal) number of real vehicles than real orders. On the other hand, if the cost value for order o is chosen as $delta_{min,o} - 1$, an assignment in sector II can be prevented with certainty. Variable (vi) $delta_{median,o}$ is calculated in order to have an intermediate value that allows "good" real order to real vehicle assignments (the better 50%) and postpones expensive assignments.

An example is given in Figure 4.12. Here, five real vehicles and two real orders are available. This results in a 7×7 - matrix. In sector I, four vehicles are assumed to have further scheduled Deliveries, hence the chosen cost values are zero. Vehicle number 2 has no further scheduled Deliveries, thus the cost values are set to *c_empty* (here: 5). Sector II shows exemplary Best Insertion cost values (e.g. cost values of 30 and 20, if real vehicle number 1 is assigned to real order number 1 and 2, respectively) and sector III includes (as specified) only zero values.

In sector IV, the first order is assumed to be *urgent*, hence the cost values are set to $delta_{max,o} + 1$ (=51), while the second order is assumed to be *not urgent* with a resulting cost value of $delta_{min,o} - 1$ (=14). The grey highlighted fields show an optimal assignment. As intended, the first real order is assigned to a real vehicle (vehicle number 5 at the cost of 10), while the second real order will be postponed (assignment of dummy vehicle number 1 at the cost of 14).

		dummy orders $o \in O^d$					real orders $o \in O$	
		1	**2**	**3**	**4**	**5**	**1**	**2**
real vehicles $v \in V$	**1**	0	0	0	0	0	30	20
	2	5	5	5	5	5	40	20
	3	0	0	0	0	0	20	60
	4	0	0	0	0	0	50	80
	5	0	0	0	0	0	10	30
dummy vehicles $v \in V^d$	**1**	0	0	0	0	0	51	14
	2	0	0	0	0	0	51	14

Figure 4.12: Assignment matrix with "urgent" (o=1) and "non urgent" (o=2) real order

Now, all auxiliary variables from (i) to (vi) have been defined and illustrated. Getting back to the general formula chosen for $c\_urgency_o$, three *urgency* levels can be differentiated. At the *first level*, the opening of the Pickup time window lies in the far distant future ($slack\_EPT > a$). The value of a can be interpreted as a measure in minutes and has to be chosen with a significant gap to zero: an assignment would result in more than a minutes of waiting. Therefore, such an order is considered *not urgent*, with sector IV costs being chosen *low* ($delta_{min,o} - 1$) to prevent a sector II assignment.

The *second level* depicts the situation of an *urgent* order. The time window is already open ($slack\_EPT_o \leq 0$) and the remaining time window width is smaller than c times of the average travel time to Pickup ($tw\_width_o < c$). Accordingly, the sector IV costs are chosen *high* in order to enforce a sector II assignment.

As a first term, $delta_{max,o}$ is selected, combined with a second term $(b - slack\_LPT_o)^2$. The first term enforces sector II assignments in the case of a smaller or equal number of real orders to real vehicles. It also works if the number of real orders, considered *urgent*, is smaller than or equal the number of real vehicles. The second term, depending on $slack\_LPT_o$, allows for a further differentiation of *urgent* orders if there are too many of them (no. of urgent orders > number of vehicles). The constant b is chosen as a high value in order to ensure $b - slack\_LPT_o > 0$ for all possible scenarios.

The *third level* reflects the intermediate situation between *not urgent* and *urgent*. An assignment in sector II is allowed if it belongs to the better 50% of possible assignments. The other 50% of worse assignments are blocked by the cost value $delta_{median,o}$, which instead attracts an assignment in sector IV. A definite sector II assignment, however, cannot be guaranteed, since it depends on the current number of *urgent* orders. Possibly, all "good" real vehicles are already "occupied".

So far, the configuration of all assignment matrix sectors and the associated cost calculations have been explained. Next, the following questions are answered: by what kind of events a matrix update is triggered and what sections of the matrix are affected in the specific cases.

4.2.3 Events and Matrix Updates

The Assignment based planning procedure basically knows two kinds of events that trigger a replanning run:

- first, the **occurrence of a new order,** and
- second, the event of **departure to the next Pickup or Delivery location**.

For the *first event*, an additional real order column is added to the assignment matrix, including the sector II Best Insertion cost and the sector IV postponement cost. In addition, a dummy vehicle row is added to keep the matrix quadratic (cp. Figure 4.13).

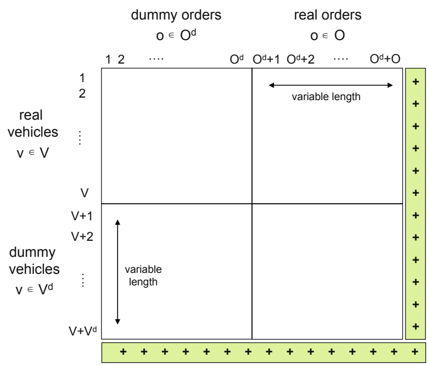


Figure 4.13: Assignment matrix update: occurrence of a new order

For the *second event*, *departure to Pickup* and *departure to Delivery* can be differentiated (cp. Figure 4.14).

In the *Pickup case*, the associated real order that is still part of the matrix must no longer be assigned to another vehicle. Hence, the real order column is completely removed from the matrix together with one row of dummy vehicles. In addition, the Pickup and the Delivery under consideration are actually inserted into the associated vehicle's scheduling, which requires a re-calculation of the complete vehicle row. Finally, all sector IV postponement values have to be updated, due to their dependency on the vehicle's row.

In the *Delivery case*, a fixation is applied to the scheduled Delivery event. This prevents any more Pickups from other open orders from being inserted before this Delivery location.

Hence, the complete vehicle row has to be re-calculated followed by an update of the sector IV postponement values.

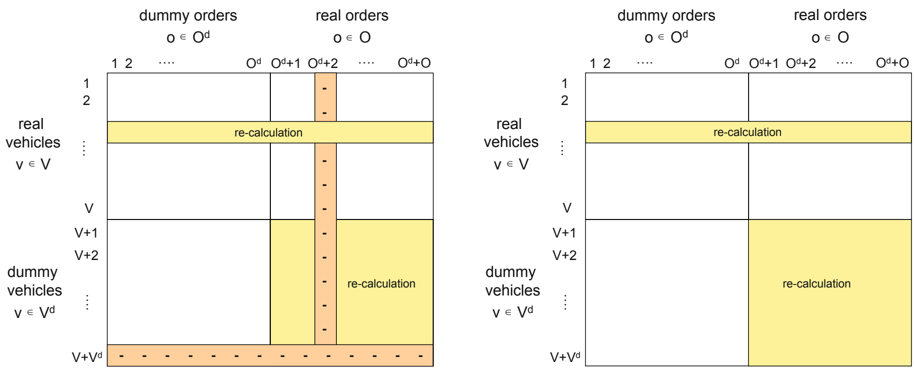


Figure 4.14: Assignment matrix update: start traveling to Pickup (left) or Delivery (right)

For better handling of the applied time discrete simulation, two additional events are defined:

- **new order-to-vehicle assignment/checkup after arrival,** and
- **end of simulation.**

These events do not trigger a matrix replanning, but help to control the simulation. A detailed description of the procedure's workflow, embedded into a time discrete simulation framework, is given subsequently.

4.2.4 Procedure Workflow and Simulation

The general procedure workflow is shown in Figure 4.15. In a first step, all input data are read from an Excel input file and stored in the appropriate data classes. The Excel file configuration is identical to the files used in Section 4.1 for the MNS procedure (see Tables 4.1 and 4.2 for typical worksheets with order and vehicle data). There are only some changes in parameter values that have to be handed to the program: instead of the anticipation horizon, now the parameters *c_empty*, *average travel time to Pickup*, *average travel time to Delivery*, and the *postponement parameters (a, b, c)* have to be specified.

Afterwards, a LinkedList, referred to as *tasklist*, is generated including all order entries with Call-In time and a simulation endtime entry. After sorting this list by increasing event time, the simulation time is set to the time of the first entry. Then all entries with current simulation time are removed from *tasklist*, being subsequently written into another LinkedList *current_events*.

Now, all events are successively removed from *current_events*, inducing different planning steps. For internal processing and sorting, each event has a specific identifier, which is chosen as a single alphabetic character ("A", "B", "C", and "D"). The events in *current_events* are sorted according to the alphabetic order of the internal identifiers, hence all "A" events are processed first, followed by the "B" events, and so on:

- **Occurrence of a new order ("A"-event):** For the new real order, the insertion costs into each real vehicle's scheduling are calculated, then the assignment matrix is updated accordingly. Afterwards, it is checked if there are further type "A"-events in *current_events*. If this is true, the same procedure is applied to the associated new orders. Otherwise, the assignment problem is solved for the updated matrix. The resulting assignments are analyzed by the method *Check Assignment results*, which will be explained later.

- **New order-to-vehicle assignment/Checkup after arrival ("B"-event):** This kind of event may be triggered by a new order-to-vehicle assignment (in such cases, the routine *Check Assignment results* has written a "B"-event for a waiting vehicle into *current_events*), and also if a vehicle arrives at a Pickup or Delivery location. In both cases, vehicles are available immediately.

  In a first query, it is checked if a sector II assignment (real order and real vehicle) is available. If this is true, it is calculated in a second query whether immediate departure would result in any waiting time at the destination. If there is no such waiting time, a type "C" event is written into *current_events*, which initiates the immediate departure to the Pickup. Otherwise, if there is any waiting time, the vehicle status is changed to *waiting*. In addition, the vehicle variables *waiting_order* and *end_wait* are updated. *End_wait* is set exactly to the departure time that results in an arrival at the destination's time window opening. Finally, a type "C" event is written into *tasklist* in order to initiate departure at the right time.

  The second branching covers the case of no available sector II assignment. It is checked if there are further scheduled activities (Deliveries) available for the concerned vehicle. If this is true, the travel time to and possible waiting at this next Delivery location is calculated. The results are handled identically to the *new order Pickup case* above: if there is no waiting time, a type "C" event is written into *current_events*, initiating the immediate departure to the Delivery. If there is waiting time, the associated vehicle attributes are updated (vehicle status = *waiting*, *waiting_order* and *end_wait*, accordingly), followed by a type "C" entry being written into *tasklist* to initiate departure at the right time.

  Finally, there is the case of a vehicle having no sector II assignment and no further scheduled activities. For such a vehicle, the status is changed to *waiting*. In addition, the vehicle attributes *waiting_order* and *end_wait* are set to the default values "-1" and "∞", respectively.

- **Start traveling to Pickup or Delivery location ("C"-event):** This kind of event is ultimately fixing a departure to a Pickup or Delivery location. It may be an immediate departure, or a departure after some waiting time.

  In a first step, it is checked if there is a current sector II assignment. If this is true, the order's Pickup and Delivery are ultimately inserted into the associated vehicle's scheduling. Then the vehicle status is set to *driving* and the attributes *waiting_order* and *end_wait* are set to the default values "-1" and "∞", respectively.

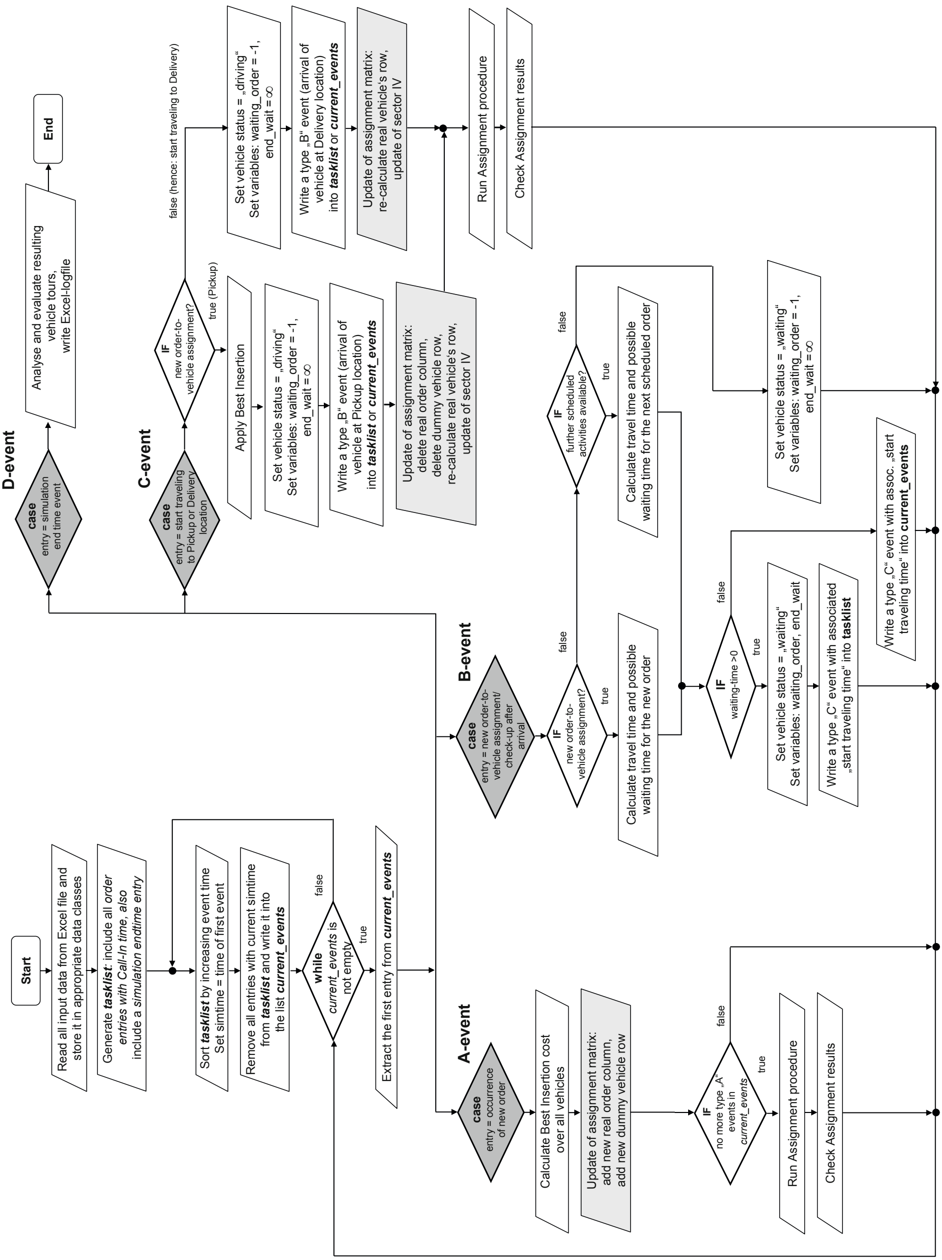


Figure 4.15: Assignment based procedure: workflow

Afterwards, a type "B" event is written into the *tasklist* with a specified event time equal to the arrival time at the order's Pickup. In the special case of a zero travel time (Pickup location = current location), the type "B" event has to be written into *current_events* in order to ensure processing at the current simulation time. Finally, the assignment matrix is updated (as already explained above), followed by a run of the assignment procedure and an analysis of the results (method: *Check Assignment results*).

If there is no current sector II assignment, the departure to the next scheduled Delivery is initiated. Vehicle status is set to *driving* and the attributes *waiting_order* and *end_wait* are set to the default values "-1" and "∞", respectively. Afterwards, a type "B" event is written into *tasklist* (or into *current_events*, in the case of a zero travel time) with a specified event time equal to the arrival time at the associated Delivery location. Finally, the assignment matrix is updated (as explained above), followed by a run of the assignment procedure and an analysis of the results (method: *Check Assignment results*).

- **Simulation end time event ("D"-event):** This kind of event stops the simulation. It is chosen with a sufficient time lag to the last possible Call-In time and also to the latest time windows to ensure the processing of all events. This event triggers analysis and evaluation of the resulting vehicle tours. The results are written into an Excel logfile.

  The output file has the same structure as in the MNS procedure. The first worksheet contains a *summary of results* (cp. Table 4.3), the second worksheet contains a *detailed vehicle scheduling for each vehicle* (cp. Table 4.4), and the third worksheet shows the *planning results from an order based perspective* (cp. Table 4.5).

Now, the method **Check Assignment results** is considered in more detail. It is called each time when a planning run of the solution procedure for the bipartite assignment problem is finished. Its main objective is to analyze the planning results for *waiting* vehicles that could be immediately affected by new assignment decisions.

Vehicles with status *driving*, however, are not considered, since they cannot be immediately affected by the planning results. Current assignment decisions concerning those (driving) vehicles may be updated several times until their next time of availability (arrival at Pickup or Delivery location). Therefore, an early consideration does not make any sense.

The pseudocode of the method *Check Assignment results* is given in Table 4.6.
Basically, three options are checked for each waiting vehicle:

- **Case 1:** *A new real order is assigned to the real vehicle and the vehicle is not waiting for another task.*

  Here, a "B" type event is written into *current_events* to check the new order-to-vehicle assignment and to initiate potential departure.

- **Case 2:** *A new real order is assigned to the real vehicle and the vehicle is waiting for a different task (departure to a new real order's Pickup or to a Delivery).*

```
For (int i=1, i ≤ number of real vehicles, i++) {
    If (vehicle_status(i) == ‘‘waiting’’) {
        If (real order is assigned to vehicle(i)) {
            If (assigned order(i) ≠ ‘‘waiting_order(i)’’) {
                //hence: there was none or a different assignment before
                Write a ‘‘B’’-type event into current_events
                If (‘‘end_wait’’ ≠ ∞) { //hence: there was a different assignment before
                    Remove obsolete ‘‘C’’-type entry from tasklist or current_events
                    Set variables: ‘‘waiting_order(i)’’ = -1 and ‘‘end_wait(i)’’ = ∞
                }
            }
        }
        Else { //hence: no real order is currently assigned to vehicle i
            If (‘‘waiting_order(i)’’ ≠ -1 & scheduled task = Pickup) {
                //hence: there was an assignment before
                Remove obsolete ‘‘C’’-type entry from tasklist or current_events
                Set variables: ‘‘waiting_order(i)’’ = -1 and ‘‘end_wait(i)’’ = ∞
                If (further scheduled activities (Deliveries) are available for vehicle i) {
                    Apply re-scheduling for vehicle i, based on current simtime
                    Write a ‘‘B’’-type event into current_events
                }
            }
        }
    }
}
```

Table 4.6: Pseudocode of method *Check Assignment results*

Here, also a “B” type event is written into *current_events* to check the new order-to-vehicle assignment and to initiate potential departure. In addition, an obsolete “C” type event from the formerly scheduled task (end of the waiting time to arrive punctually at the old order’s Pickup or Delivery location) has to be removed from *tasklist* or *current_events*. The vehicle attributes *waiting_order* and *end_wait* are reset to the default values “-1” and “∞”, respectively.

- **Case 3:** *No real order is assigned to the real vehicle, but there was a real order assignment before.*

  Here, in a first step the obsolete “C” type event from the formerly assigned real order has to be removed from *tasklist* or *current_events*. In addition, the vehicle attributes *waiting_order* and *end_wait* are reset to the default values “-1” and “∞”, respectively.

  Afterwards, it is checked if there are further scheduled Delivery events available for the considered vehicle. If this is true, a re-scheduling is applied based on the current simulation time. Then a type “B” event is written into *current_events* to initiate potential departure to the next Delivery location.

In the following section, a comparison of both procedures that have been introduced in Section 4.1 and 4.2 is carried out.

4.3 A Comparison of Both Procedures

Both procedures, MNS and the Assignment based procedure, can be characterized according to the **configuration framework for dynamic algorithms** that was proposed in Section 2.1. The particular attributes are summarized in Tables 4.7 and 4.8 for MNS and the Assignment based procedure, respectively.

The *first procedure* uses a *classical Best Insertion algorithm combined with improvement measures making use of multiple neighborhoods* as **technique of adjustment**. The *second procedure* applies an *optimal bipartite assignment algorithm* that triggers order-to-vehicle assignment. For calculation of the order-to-vehicle assignment costs, procedure one uses a classical Best Insertion technique that investigates all possible insertion options. In contrast, the second procedure applies a *specific Best Insertion variant* that only allows for the insertion of a Pickup at the first scheduling position.

Insertion based procedure with Multiple Neighborhood Search	
technique of adjustment	- *classical Best Insertion + Multiple Neighborhood Search*
reaction of adjustment	- *constructive,* updating former planning results
frequency of adjustment	- *time driven* replanning
duration of adjustment	- *time limit based* with fixed anticipation horizon
synchronisation of adjustment	- *extensive simultaneity* of plan execution and planning
scope of adjustment	- *restricted,* not all real-life options are allowed

Table 4.7: Characteristics of the first planning approach, according to the *Dynamic Algorithm Configuration Framework*

Assignment Based Procedure	
technique of adjustment	- *optimal bipartite assignment*, based on specific Best Insertion
reaction of adjustment	- *from scratch*
frequency of adjustment	- *event-based* replanning
duration of adjustment	- *zero time*
synchronisation of adjustment	- *extensive simultaneity* of plan execution and planning
scope of adjustment	- *restricted,* not all real-life options are allowed

Table 4.8: Characteristics of the second planning approach, according to the *Dynamic Algorithm Configuration Framework*

The **reaction of adjustment** is developed as follows. In the *first procedure*, new dynamic information is incorporated in a *constructive* way, simply updating the results of the last planning run and not discarding the old solution. The *second procedure* actually performs a *from scratch* re-planning. A new dynamic information may result in a complete change of all prior assignment decisions as long as these decisions have not yet been ultimately *fixed.*

The next characterizing attribute is the **frequency of adjustment**. In the *first procedure*, a new replanning run is triggered periodically by elapsed time (*time-based*). Thus, the time available to the algorithm for inclusion of new information and for improvement operations is known to be at a prespecified constant level, equal to the *anticipation horizon* (**duration of adjustment**).

The *second procedure*, however, performs an *event-based* replanning technique. This ensures immediate reaction, but, on the other hand, does not guarantee predictable information on the time available for replanning runs. Consequently, an algorithm is chosen that ensures a *zero-time* duration of adjustment.

Synchronization of adjustment is performed in a similar way in both approaches. *Plan execution* and *replanning* run *simultaneously*, with the algorithm being allowed to change all decisions which are not *fixed.* In the *first approach*, an anticipation horizon, which has to be chosen as > 0 time units, is introduced and triggers *fixation* of all events that are due in the horizon, as well as fixation of all dependent events. The associated rolling horizon approach produces planning certainty for the vehicles in execution and defines a clear field of changeable scheduling options to the planning algorithm, thus keeping plan in execution and plan updates consistent.

The *second approach* works without such an anticipation horizon. Since replanning time is near zero, it is assumed that a decision can be executed immediately. *Fixation* is not applied in order to achieve simultaneity, but only in the case of a departure event that induces ultimate *fixation* to parts of the scheduling.

Since technically possible real-life options like "en route diversion" or "transshipment" are not supported in *both procedures*, the **scope of adjustment** must be described as *restricted* twice.

It can be concluded that both procedures use quite differing concepts to cope with dynamic information. In order to prove and compare the performance of these planning approaches, some suitable test data sets are needed.

4.4 Test Data

In the following, two test data sets for the dynamic MLPDP with soft time windows are presented. The first one is *self-generated* with the help of an own data generator and covers the originally intended "capacitated" dynamic MLPDPTW (real-life application: Dial-A-Ride services). The second data set is adopted *from the literature* and was chosen because of its good documentation and availability of detailed results. This data set covers the slightly differing case of an "uncapacitated" dynamic MLPDPTW (real-life application: Express Mail Delivery). Both procedures from Sections 4.1 and 4.2 are capable of solving the uncapacitated problem version by simply setting each vehicle's capacity to infinity.

4.4.1 Self-Generated Test Scenarios

The self-generated data sets (partially based on Ivankina, 2004) basically consist of three components: an underlying *node network*, *dynamic and static orders* and *available vehicles*. Subsequently, different specifications of these three components are presented, which can be combined in all possible ways. A specific selection (node network, order characteristic, vehicle characteristic) is handed as input information to a data generator, which produces the desired test instance.

Node Network

Two node networks are available (cp. Figure 4.16). Each network contains 1000 possible customer locations characterized by an x- and y- coordinate. The distance between these nodes is calculated using the Euclidean metric. Vehicles are located at a central depot with coordinates (x=0, y=0).

- **network 1** is a circle with a radius of 15km. Within the circle, there is a randomly located cluster, covering 20% of total circle area and containing 40% of all nodes.

The remaining 60% of nodes are equally distributed across the remaining circle area.

- **network 2** is also circle shaped, with a radius of 15km. However, the number of randomly located clusters is increased to three, with two clusters covering 10% of the circle area and incorporating 25% of nodes each. The third cluster covers 5% of the circle area and possesses 20% of all nodes. Again, the remaining nodes are equally distributed across the cluster free circle area.

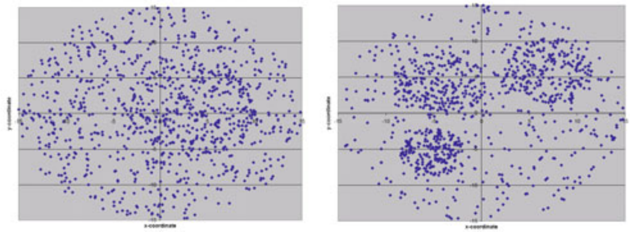


Figure 4.16: Node network 1 (left) and node network 2 (right) - distribution of customers

Static and dynamic orders

The second choice concerns the order characteristic. For every instance, 1000 orders are generated by randomly choosing Pickup and Delivery locations out of the available network nodes. Each order is specified to consume one unit of available vehicle capacity. Furthermore, at every Pickup and Delivery location, a loading/unloading time of 2 minutes has to be scheduled.

In the following, some basic settings for the dynamic and static order generation are explained:

The dynamic orders' Call-In times are chosen as equally distributed in the time interval [09:00, 17:00] with a 30-minute time gap between Call-In and EPT. Hence, the latest time window opening may be scheduled at 17:30 (for an order with Call-In at 17:00). The Pickup time window has a width of 45 minutes (LPT = EPT + 45 min), the Delivery time window opens 15 minutes after EPT and has a width of 90 minutes (LDT = EDT + 90 min). This time window characteristic is denoted (45,15,90).

The static orders are assumed to be known before 9:00. Here, EPT is chosen to be equally distributed in the interval [09:00, 13:00]. In contrast to the dynamic requests, the time windows of static orders have a width of 4 hours, for both, Pickup and Delivery. As in the dynamic case, the Delivery time window opens 15 minutes after EPT. This time window characteristic is denoted (240,15,240)

These basic setting can be changed as follows:

- The *time gap between Call-In and EPT* may be varied. The basic setting with a time gap of 30 minutes is supplemented by five additional options: 0 min, 5 min, 10 min, 15 min, 20 min, and 25 min.

- The *time window width of dynamic orders* may be varied. The basic setting of (45,15,90) is complemented by the options (15,15,30), (30,15,60) and (60,15,120).

In addition, the total share of dynamic and static orders has to be specified. Here, varying degrees of dynamism between 0% and 100% can be chosen in steps of 10%.

Vehicles

At first, some basic assumptions, which are not subject to changes, are explained. Each vehicle starts and ends its tour at the depot that is specified by the underlying network. Regular operating time is defined as in the interval [09:00, 19:00]; afterwards penalty costs for overtime are charged. It is assumed that a vehicle drives at an average speed of 30km/h.

There are two settings that can be changed:

- The *number of vehicles* may be varied in the interval between 42 and 50.
- The *vehicle capacity* may be varied between 1 and 7 units.

With the specified characteristics, the data generator is capable of producing 38 808 different instances (combinations of networks, order and vehicle characteristics). Table 4.9 shows *six main scenarios* and the associated number of instances that have been chosen for detailed consideration in Section 4.5.

Scenarios 1 and 2 allow for an investigation of the impact of varying degrees of dynamism on solution quality. In addition, it can be analyzed whether the use of the different underlying networks 1 and 2 causes any variations in solution quality. In scenario 3, the reaction time that is given to the procedure in order to handle new information is successively decreased. Scenario 4 focuses on the impact of reduced and increased time window opening time. In scenario 5, the number of vehicles is successively reduced.

Finally, scenario 6 allows for an analysis of the two procedures' behavior in the case of varying vehicle capacity. This is of special interest for a comparison with the second test scenario from the literature, which (as already mentioned) deals with the uncapacitated problem version.

4.4.2 Benchmark Data from the Literature

The chosen data set created by Gendreau et al. (2006) is of special interest, since the authors do not only publish average benchmark results, but also detailed objective function values for each of the 15 instances. In the following, the generation process of the instances is explained in detail. Afterwards an analysis is conducted in order to compare the specific characteristics of the data set scenarios. Finally the benchmark results, which were reported by Gendreau et al. (2006) are listed.

The authors generate a 5km×5km unit square as underlying geographical area with depot location at point (2.0km, 2.5km). The area is divided into 4×5 rectangular zones, each

scenario 1 (11 instances)

node network:	1
order characteristics:	*variation of the degree of dynamism from 0% up to 100%,* dynamic time window characteristic (45,15,90), time gap Call-In to EPT 30 min
vehicle characteristics:	50 vehicles, capacity 2

scenario 2 (11 instances)

node network:	2
order characteristics:	*variation of the degree of dynamism from 0% up to 100%,* dynamic time window characteristic (45,15,90), time gap Call-In to EPT 30 min
vehicle characteristics:	50 vehicles, capacity 2

scenario 3 (7 instances)

node network:	2
order characteristics:	degree of dynamism 100%, dynamic time window characteristic (45,15,90), *variation of the time gap between Call-In and EPT*
vehicle characteristics:	50 vehicles, capacity 2

scenario 4 (4 instances)

node network:	2
order characteristics:	degree of dynamism 100%, *variation of dynamic time window characteristic,* time gap Call-In to EPT 30 min
vehicle characteristics:	50 vehicles, capacity 2

scenario 5 (9 instances)

node network:	1
order characteristics:	degree of dynamism 100%, dynamic time window characteristic (45,15,90), time gap Call-In to EPT 30 min
vehicle characteristics:	*variation of number of vehicles from 42 to 50*, capacity 3

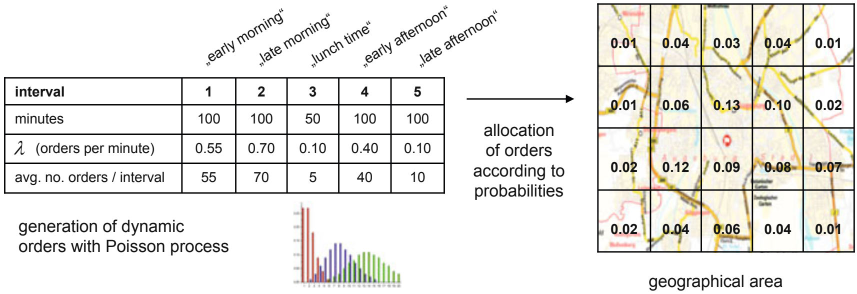
scenario 6 (7 instances)

node network:	2
order characteristics:	degree of dynamism 100%, dynamic time window characteristic (45,15,90), time gap Call-In to EPT 30 min
vehicle characteristics:	50 vehicles, *variation of vehicle capacity from 1 to 7*

Table 4.9: Investigated test scenarios generated with own data generator

possessing a specific probability between 0.01 and 0.13. A fixed number of vehicles (10 or 20) is available at the depot. Those vehicles move with a constant average speed of 30 km/h and have to return to the depot at the end of their trips. Distances within the unit square are calculated with the Euclidean metric.

For generation of dynamic requests (cp. Figure 4.17), the selected planning horizon (450 min or 240 min) is divided into five time periods: early morning, late morning, lunch time, early afternoon, and late afternoon. The lunch time period is half the length of the others, which are of equal length. Thus, for a 450 min planning horizon, a lunch time period has a duration of 50 minutes, while the other time periods have a duration of 100 minutes.



interval	1	2	3	4	5
minutes	100	100	50	100	100
λ (orders per minute)	0.55	0.70	0.10	0.40	0.10
avg. no. orders / interval	55	70	5	40	10

Figure 4.17: Generation process of dynamic requests

The arrival of new requests is modeled with a Poisson process, having different arrival intensities for the respective time periods. Generally, two sets of intensity parameters λ (requests per minute) are available: (0.75, 1.10, 0.25, 0.40, 0.10) and (0.55, 0.70, 0.10, 0.40, 0.10), which lead on average to 33 and 24 requests per hour, respectively. Each time a new request occurs, its Pickup and Delivery locations are allocated to one of the zones of the geographical area according to the associated probabilities.[9]

In addition, a service time of five minutes has to be scheduled at each location.

Under these basic conditions, the authors create three test scenarios *with increasing stress* for the planning procedure:

- A 450-minute horizon with arrival intensity of 24 requests per hour and a vehicle fleet size of 20 ("req_rapide_x_450_24"),
- a 240-minute horizon with arrival intensity of 24 requests per hour and a vehicle fleet size of 10 ("req_rapide_x_240_24"), and
- a 240-minute horizon with arrival intensity of 33 requests per hour and a vehicle fleet size of 10 ("req_rapide_x_240_33").

[9] The determination process of exact locations within the zones is not specified in the publication.

scenario I: req_rapide_x_450_24	**scenario II:** req_rapide_x_240_24	**scenario III:** req_rapide_x_240_33
• **run-time:** 7.5 hours (8:00 – 15:30) • **Ø arrival intensity:** 24 orders/h	• **run-time:** 4 hours (8:00 – 12:00) • **Ø arrival intensity:** 24 orders/h	• **run-time:** 4 hours (8:00 – 12:00) • **Ø arrival intensity:** 33 orders/h
• **total number of orders (x=1..5):** (169, 176, 206, 215, 202) **Ø no. of orders/h (x=1..5):** (22.53, 23.47, 27.47, 28.67, 26.93)	• **total number of orders (x=1..5):** (84, 94, 93, 90, 85) **Ø no. of orders/hour (x=1..5):** (21, 23.5, 23.25, 22.5, 21.25)	• **total number of orders (x=1..5):** (144, 112, 111, 119, 153) **Ø no. of orders/hour (x=1..5):** (36, 28, 27.75, 29.75, 38.25)
• **Ø arrival profile of orders:** (1h, 2h, 3h, 4h, 5h, 6h, 7h, 8h) (33, 36, 42, 18, 21, 21, 6, 3, 0)	• **Ø arrival profile of orders:** (1h, 2h, 3h, 4h) (34, 34, 20, 8)	• **Ø arrival profile of orders:** (1h, 2h, 3h, 4h) (47.33, 54.66, 22.11, 8.04)
• **width of time windows:** **P:** Ø 1h48, min: 0h00, max: 5h33 **D:** Ø 1h41, min: 0h00, max: 6h43	• **width of time windows:** P: Ø 0h58, min: 0h00, max: 2h40 D: Ø 0h55, min: 0h00, max: 2h53	• **width of time windows:** P: Ø 0h53, min: 0h01, max: 2h48 D: Ø 0h47, min: 0h00, max: 3h10
• **gap Call-In to EPT and LPT:** **EPT:** Ø 1h32, min: 0h00, max: 6h53 **LPT:** Ø 3h21, min: 0h30, max: 7h13	• **gap Call-In to EPT and LPT:** EPT: Ø 0h45, min: 0h00, max: 3h35 LPT: Ø 1h44, min: 0h30, max: 3h45	• **gap Call-In to EPT and LPT:** EPT: Ø 0h39, min: 0h00, max: 3h20 LPT: Ø 1h33, min: 0h30, max: 3h44
• **gap EPT to EDT:** Ø 1h13, min: 0h06, max: 6h25	• **gap EPT to EDT:** Ø 0h41, min: 0h06, max: 2h44	• **gap EPT to EDT:** Ø 0h35, min: 0h06, max: 3h01
• **20 vehicles**	• **10 vehicles**	• **10 vehicles**
• operating time: 9000 min (8:00 – 15:30) • traveling: 180*(2+2)+ 20*2=760 min • loading: 180*(5+5)= 1800 min • waiting: 6440 min	• operating time: 2400 min (8:00 – 12:00) • traveling: 96*(2+2)+ 10*2=404 min • loading: 96*(5+5)= 960 min • waiting: 1036 min	• operating time: 2400 min (8:00 – 12:00) • traveling: 132*(2+2)+10*2=548 min • loading: 132*(5+5)= 1320 min • waiting: 532 min
• **utilization: 28%**	• **utilization: 56.8%**	• **utilization: 77.8%**

Figure 4.18: Characteristic of three test scenarios, each including five instances (x=1...5)

For each of these scenarios, five instances are generated (x=1...5), resulting in an overall number of 15 dynamic instances. Figure 4.18 shows a detailed analysis of the associated data sets. The first column scenario entries are exemplarily explained in the following.

At the beginning, general information on the *planning horizon* (7.5 hours) and on the *average order arrival intensity* (24 orders per hour) is repeated. Afterwards, the *total number of orders* and the resulting *average number of orders per hour* are given for the five instances. Due to stochasticity within the generation process, the numbers differ slightly from instance to instance. In a next step, an average *order arrival profile* is shown for all instances: e.g. indicating that in the third hour of the planning horizon, a peak of 42 orders is assumed to arrive, whereas there are only 3 expected orders in the seventh hour.

The *average width of the time windows* is analyzed to be 1h48min and 1h41min for Pickup and Delivery, respectively. In addition, a range in form of minimum and maximum values is given for both, Pickup and Delivery time window widths. Another interesting point is the *gap between Call-In and the Pickup time window (EPT and LPT)*, which is listed subsequently. While average *reaction time* (LPT - Call-In) is 3h21min, the analysis also shows the possibility of a very short *reaction time* of 30 minutes. Furthermore, the *gap between EPT and EDT* is specified with an average length of 1h13min, a minimum value of 0h06min and maximum value of 6h25min.

In the next row, the number of *available vehicles* (20) is listed. This is followed by a calculation of the *approximate utilization* if all requests were served in regular operating time:

The total regular operating time of 20 vehicles (each available for 7.5 hours) is 9000 minutes. The approximated total travel time is 760 minutes and consists of two minutes for each trip to one of the 180 Pickup locations, of two minutes for each trip to the 180 Delivery locations and of two minutes for each of the vehicles to return to the depot at the end of the day.[10] The approximated total travel times[11] and the loading times (1800 min) are subtracted from the total regular operating time, which results in a vehicle waiting or idle time of 6440 minutes. This induces an approximate vehicle utilization of 28%.

The increasing stress from scenario one up to three can be easily reconstructed by increasing utilization values of 56.8% and 77.8%, in scenarios 2 and 3, respectively.

The solution approach that was applied to the proposed test data sets (Parallel Tabu Search with Adaptive Memory) has already been explained in Section 3.3.1.2. The objective function includes *travel time*, *delay* and *overtime*, which are weighted equally. The achieved benchmark results for all 15 instances are listed in Table 4.10: The first column shows the declaration of the associated instance, the subsequent columns include objective function values for travel time, delay and overtime (in minutes). These values are taken as benchmarks.

req_rapide_x_450_24	**travel time**	**delay**	**overtime**
x = 1	539 min	1 min	0 min
x = 2	614 min	3 min	1 min
x = 3	629 min	2 min	1 min
x = 4	700 min	6 min	5 min
x = 5	694 min	0 min	0 min

req_rapide_x_240_24	**travel time**	**delay**	**overtime**
x = 1	336 min	65 min	55 min
x = 2	386 min	68 min	53 min
x = 3	352 min	139 min	98 min
x = 4	359 min	38 min	31 min
x = 5	348 min	75 min	52 min

req_rapide_x_240_33	**travel time**	**delay**	**overtime**
x = 1	473 min	4392 min	699 min
x = 2	402 min	867 min	297 min
x = 3	455 min	853 min	303 min
x = 4	434 min	1104 min	348 min
x = 5	495 min	4121 min	687 min

Table 4.10: Benchmark results for the 15 test instances of Gendreau et al. (2006)

[10] The average travel times to Pickup locations, to Delivery locations and back to the depot are based on an ex-post analysis of the planning results that have been achieved with the procedures from Sections 4.1 and 4.2.

[11] The calculation of total travel time also includes empty travel times, e.g. the trip to the first Pickup location or the last trip back to the depot. The ex-post analysis of the planning results shows that most of the trips to Pickup locations are, however, performed by already loaded vehicles.

4.5 Computational Results and Performance Analysis

This section contains computational results for the self-generated data set and the data set from the literature. The first subsection 4.5.1, which includes the test results for the self-generated data sets, focuses on the relative comparison of both introduced dynamic procedures for varying dynamic test scenarios. The second subsection 4.5.2, which includes the test results for the data sets from the literature, does not only allow for a relative comparison of both procedures, but also for a comparison with other benchmark results. Finally, it is analyzed if the differences of the applied data sets (capacitated vehicles, in the case of the self-generated instances and uncapacitated vehicles, in the case of the instances from the literature) lead to major differences in the procedures' planning performance.

4.5.1 Computational Results for Self-Generated Test Scenarios

In Section 4.4.1, six main scenarios with different variations were generated. All associated instances are solved with *MNS* and with the *Assignment based procedure* on a desktop PC (Intel Core 2 CPU, 2.40 Ghz, 3 GB RAM). MNS simulations are run with simulation speed s = 1 (real time), for the Assignment based procedure an event based simulation is applied. In the following, the results of both procedures are visualized in three diagrams per scenario, which include the three objective function criteria: travel time, delay and overtime.

Table 4.11 shows the parameter and penalty cost settings that are chosen for MNS and for the Assignment based procedure. Parameterization has been accomplished for a basic scenario consisting of node network 2, 100% dynamic requests, time window characteristic (45,15,90), and a vehicle capacity of 2.

MNS		**Assignment**	
internal parameters		**internal parameters**	
initial improvement		avg. time to Pickup:	8 min
duration:	60 min	avg. time to Delivery:	11 min
neighborhoods I:II:III	1:1:1	matrix calculations	
general improvement:		a:	150
neighborhoods I:II	2:1	b:	600
tabu time:	30 min	c:	3
anticipation:	3 min	c_empty:	10 000
penalty costs		**penalty costs**	
c_traveling (per min):	90	c_traveling (per min):	300
c_delay (per min):	10 000	c_delay (per min):	1200
c_wait (per min):	0	c_wait (per min):	60
c_overtime (per min):	10 000	c_overtime (per min):	1200

Table 4.11: Parameter settings for self-generated test data sets

Figures 4.19, 4.20, and 4.21 exemplarily show the parameterization of the MNS anticipation horizon. The anticipation horizon is varied between 1 minute and 10 minutes in steps of 1 minute. Figure 4.19 illustrates the resulting travel times, Figure 4.20 includes

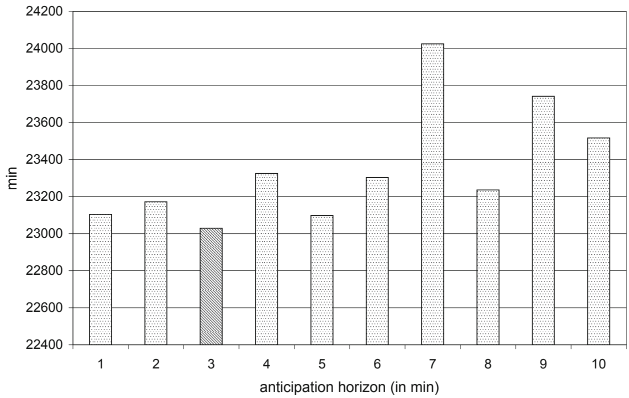


Figure 4.19: Travel time for diff. anticip. horizons (100% dyn. orders, netw. 2, cap. 2)

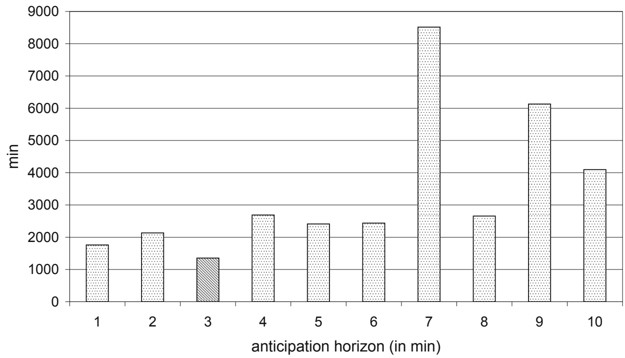


Figure 4.20: Delay for diff. anticip. horizons (100% dyn. orders, netw. 2, cap. 2)

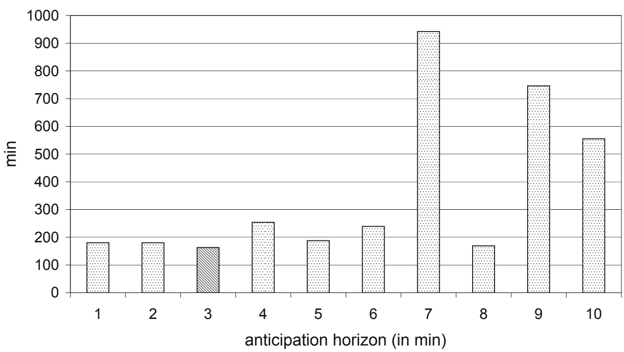


Figure 4.21: Overtime for diff. anticip. horizons (100% dyn. orders, netw. 2, cap. 2)

the resulting delay, and Figure 4.21 contains the resulting overtime. Best results in all three categories are achieved with an anticipation horizon of 3 minutes. Therefore, this parameter setting is chosen for the subsequent tests.

The other parameters are determined in analogical manner for the same basic scenario. At this point, the question of whether it is sufficient to test all instances with the same parameter settings may arise. Especially, in the case of a low degree of dynamism, other parameter settings may perform better. However, since these tests are focused on a relative comparison of both proposed dynamic procedures, such an individual parameterization for every instance is not necessary. On the contrary, results for varying data set characteristics, based on the basic parameterization, allow for additional insights into the procedures' robustness.

Scenarios 1 and 2 investigate the procedures' performance for varying degrees of dynamism and different underlying node networks. The first column of Figure 4.22 shows the results of the objective function criteria travel time, delay and overtime for network 1. The second column of Figure 4.22 illustrates the results for network 2.

In the first case (network 1), in eight of eleven instances (exceptions: degree of dynamism 70%, 80% and 90%) the *Assignment based procedure* generates significantly better results in travel time (-3.8% on average) and delay (-47.5% on average). Overtime remains at a relatively low level for both procedures. Due to an outlyer at a degree of dynamism of 80%, the average overtime of the *Assignment based procedure* is worse compared to *MNS* (+32.4%). In the second case (network 2), the situation is quite similar: for all degrees of dynamism, except 90%, the *Assignment based procedure* performs better. In comparison with *MNS*, an average reduction in travel time (-5.5%), delay (-53.1%) and overtime (-12.7%) can be achieved.

The choice of the underlying node network induces some variations in the objective function values; the general conclusion of a better performing *Assignment based procedure*, however, is identical for both node networks. Therefore, the choice of different node networks (within the range of the available test data) is not assumed to generate significant changes to the overall conclusions.

A general behavior of solution quality in dependency of the degree of dynamism cannot be observed. The intuitive assumption would be a better solution quality with decreasing degree of dynamism, since there is more information available at an earlier time.

There may be several reasons for not discovering such a behavior. A dynamic algorithm is specialized to run on a dynamic instance, thus its performance on a more or less static instance may be worse. In addition, there is the question of the appropriate parameterization of the dynamic algorithms when applying them on static instances. The parameterization for a degree of dynamism of 100% may be not the best choice for a low degree of dynamism.

Another aspect could be related to the workload that has to be handled by the algorithm. In the static case, there is a huge initial workload with a huge solution space. In the dynamic cases, however, the information is revealed little by little, which reduces the

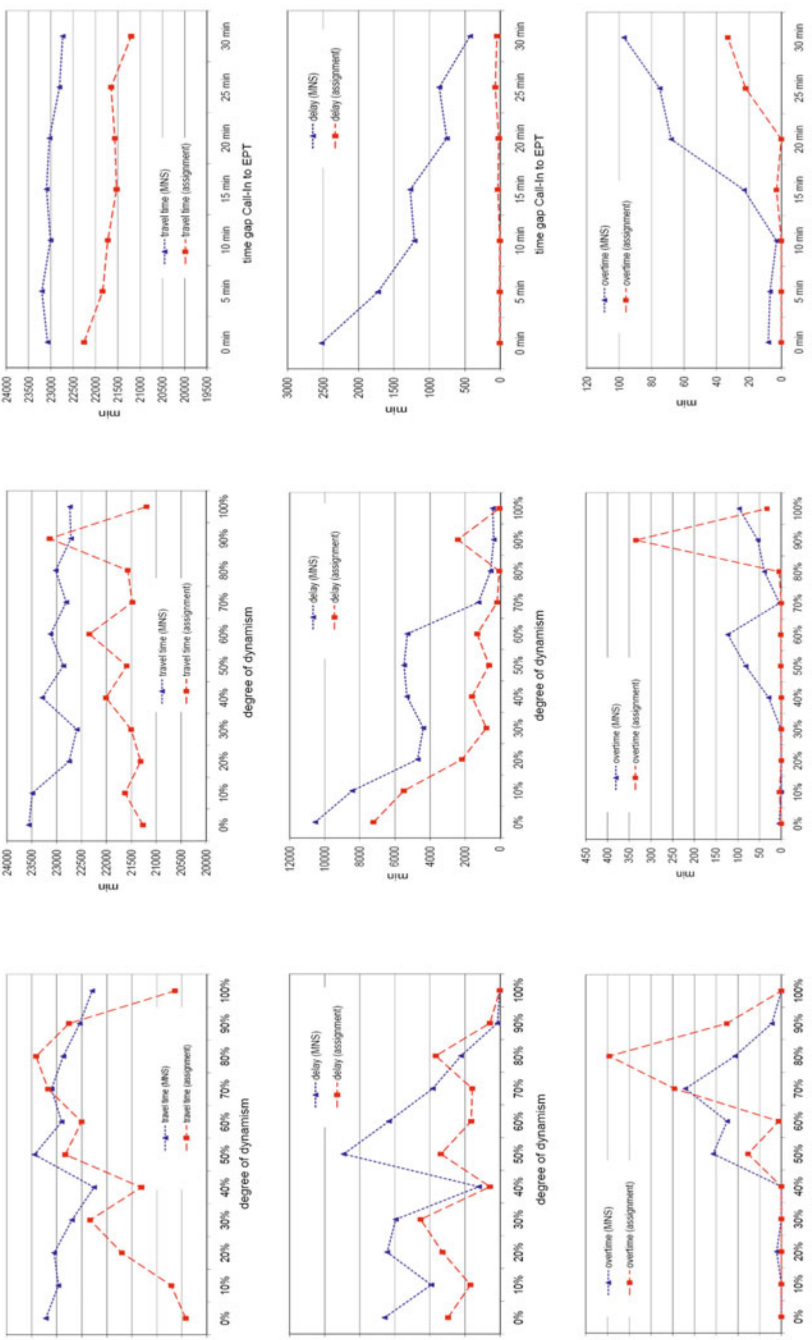


Figure 4.22:
Column 1: travel time/delay/overtime for varying degrees of dynamism (netw. 1, cap. 2)
Column 2: travel time/delay/overtime for varying degrees of dynamism (netw. 2, cap. 2)
Column 3: travel time/delay/overtime for varying gap between Call-In and EPT (100% dyn. orders, netw. 2, cap. 2)

possible alternatives. In this way, the revealment of dynamic information may "guide" the dynamic algorithm, by retaining some currently unimportant information. Similar experiences have been noted by other authors. Larsen et al. (2004) find "high variability over the entire dod spectrum" and conclude that "lower dod problems are harder to solve, because they involve larger instances than their higher dod counterparts".

In **scenario 3**, the time gap between Call-In and Earliest Pickup Time (EPT) is varied between 0 minutes and 30 minutes, in steps of 5 minutes. The underlying basic scenario contains the nodes of network 2, vehicles with a capacity of 2, the time window characteristic (45,15,90), and 100% dynamic customers.

The results are shown in the third column of Figure 4.22. Travel time decreases slightly for both procedures when the time gap is increased. The delay of the *MNS* procedure can be reduced significantly when a higher time gap is chosen. The delay of the *Assignment based procedure* remains on a permanently low level. The improved results which are achieved with longer time gaps are plausible, since there is a longer reaction time available to the algorithm. This especially benefits the improvement process of *MNS*.

Reductions in travel time and delay come along with a small increase in overtime for both procedures. This negative effect, however, is outweighed by far through the improvements in the other objective function criteria, and can be attributed to an internal trade off that accepts a small worsening in overtime in order to achieve significant better results for travel time and delay. In all instances of scenario 3, the better performance of the *Assignment based procedure* is beyond question.

Scenario 4 deals with a variation of the time window characteristic. The original characteristic (45,15,90) is changed to the shorter time window characteristics (15,15,30) and (30,15,60), as well as to the longer time window characteristic (60,15,120). As further settings, node network 2, vehicles with capacity 2 and 100% dynamic customers are chosen.

By trend, it can be observed that smaller time windows result in an increase in travel time, delay and overtime (cp. Figure 4.23, column 1). When comparing the performance of both dynamic procedures, an interesting behavior can be found. While the *Assignment based procedure* produces better results for the two "longer" time window characteristics, the picture changes for the two "shorter" characteristics. Here, the application of *MNS* produces better results in all three objective function criteria.

In **scenario 5**, the impact of a reduced number of available vehicles is investigated. Starting at the original number of 50 vehicles, the number is reduced successively to 42. As further settings, node network 1, vehicles with capacity 3, time window characteristic (45,15,90), and 100% dynamic customers are chosen.

The *Assignment based procedure's* advantage in solution quality persists up to a level of 46 available vehicles (cp. Figure 4.23, column 2). Afterwards, for a smaller number of vehicles, *MNS* achieves better solution quality. For a number of 44 down to 42 vehicles, there is a dramatic worsening of the *Assignment* results in delay and overtime. An increase of delay and overtime can also be recognized for *MNS*, but with a much more moderate rise.

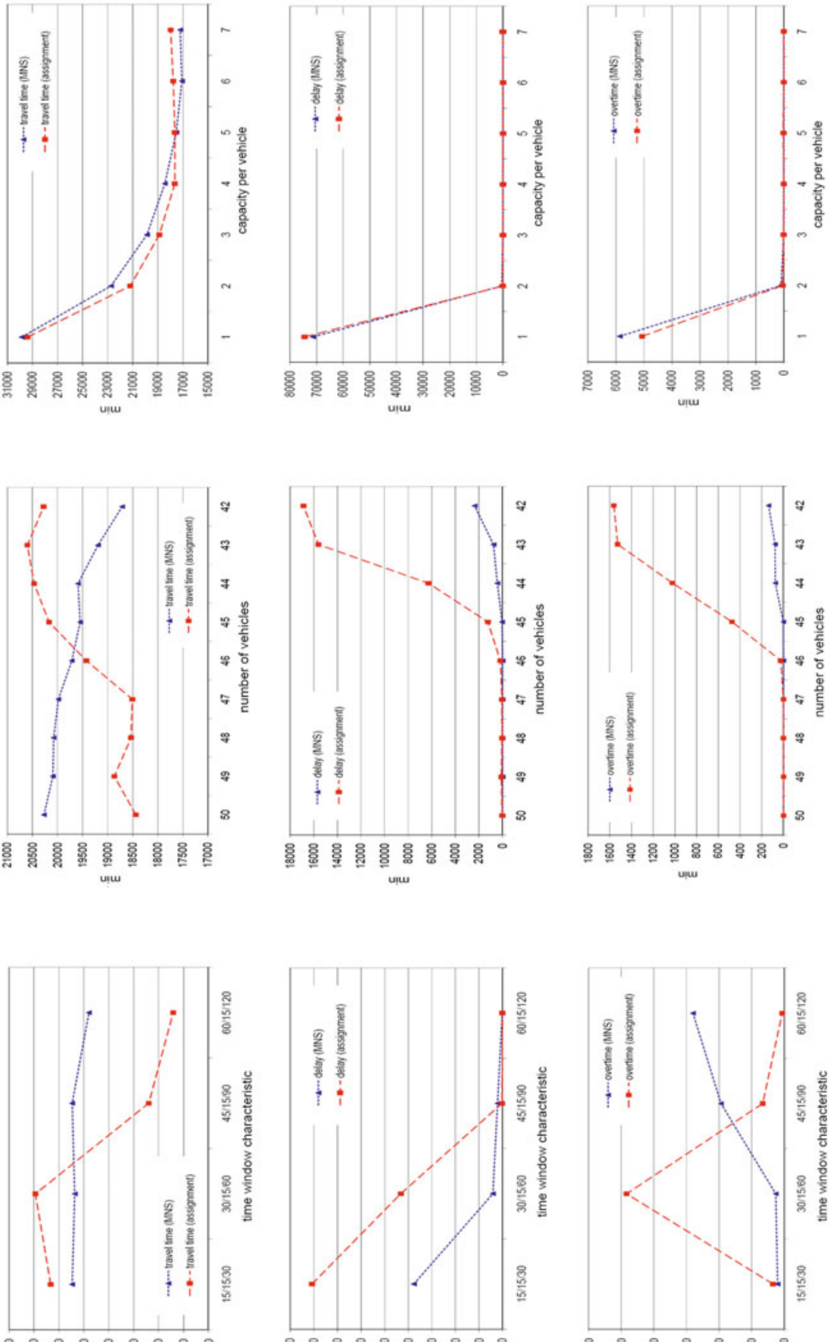


Figure 4.23:
Column 1: travel time/delay/overtime for varying time window characteristics (100% dyn. orders, netw. 2, cap. 2)
Column 2: travel time/delay/overtime for varying number of vehicles (100% dyn. orders, etw. 1, cap. 3)
Column 3: travel time/delay/overtime for varying vehicle capacity (100% dyn. orders, netw. 2)

For the travel time criterion, an antithetic behavior of the procedures' results can be observed. With the reduction of the number of vehicles, the travel time generated by the *Assignment based procedure* shows an increasing trend. Travel time generated by *MNS*, however, shrinks with a decrease of available vehicles.

The first behavior is the intuitive one. However, there are some reasons the behavior of the *MNS* results can be attributed to:

- With a sufficient number of vehicles, an urgent order is preferably transported by an immediately available vehicle to ensure on time arrival. For this purpose, some extra travel time may be accepted. However, if there is a situation of many *urgent* orders, minimization of travel time may become the decisive aspect. Suppose two orders, whose immediate execution results in an identically high level of penalty costs for delay. MNS will schedule these orders according to travel time criteria, since there is no more possible differentiation based on the urgency costs.

- There is also a second reason for the decrease in travel time: if the number of vehicles is reduced, there is also a direct reduction in total travel time, since less vehicles have to return from their last tour position back to the depot.

Generally, it can be concluded that *MNS* has the better capability to cope with a situation of a scarce number of vehicles.

Finally, in **scenario 6**, the vehicle capacity is varied in the interval from 1 to 7 (cp. Figure 4.23, column 3). As further settings, node network 2, vehicles with capacity 2, time window characteristic (45,15,90), and 100% dynamic customers are chosen.

Considering the travel time results, a better performance of the *Assignment based procedure* can be observed at the capacity levels from 1 to 4; afterwards, at capacity levels from 5 to 7, *MNS* produces better results. Delay and overtime results of *Assignment* and *MNS* are not very different. From capacity levels 2 up to 7, the results are nearly identical. Only at a capacity level of 1 there is a small variation: while *MNS* achieves a better result in category delay, the *Assignment based procedure* produces a smaller overtime.

In summary, better results for the self-generated test data sets are achieved with the *Assignment based procedure*. Nevertheless, in some scenarios (with major deviations from the parameterized basic scenario), the *Assignment based procedure* shows a less robust behavior with a significant decline in objective function value. Especially in the situation of vehicle scarcity, the results in delay and overtime reach inacceptable levels.

When compared to the *Assignment based procedure*, *MNS* generates inferior results on average. However, for some degrees of dynamism (around 90%), the produced results are competitive or even better. In addition, a good adaptability to changing conditions is exhibited: especially in situations of short time windows, scarcity of vehicles, and also for less capacity restricted situations proper results can be observed.

4.5.2 Computational Results for Benchmark Data from the Literature

In Section 4.4.2, the three test scenarios published by Gendreau et al. (2006) are presented and analyzed in detail. All associated 15 instances are solved with MNS and with the Assignment based procedure on a desktop PC (Intel Core 2 CPU, 2.40 Ghz, 3 GB RAM). MNS simulations are run with simulation speed s = 1 (realtime). In the following, the results of both procedures are visualized in three diagrams per scenario, which include the three objective function criteria: travel time, delay and overtime.

Table 4.12 shows the parameter and penalty cost settings that are chosen for MNS and for the Assignment based procedure. Parameterization was accomplished for scenario req_rapide_1_240_33.

MNS		**Assignment**	
internal parameters		**internal parameters**	
initial improvement		avg. time to Pickup:	2 min
duration:	60 min	avg. time to Delivery:	2 min
neighborhoods I:II:III	1:1:1	matrix calculations	
general improvement:		a:	150
neighborhoods I:II	2:1	b:	600
tabu time:	30 min	c:	3
anticipation:	1 min	c_empty:	10 000
penalty costs		**penalty costs**	
c_traveling (per min):	90	c_traveling (per min):	100
c_delay (per min):	10 000	c_delay (per min):	300
c_wait (per min):	0	c_wait (per min):	3
c_overtime (per min):	10 000	c_overtime (per min):	3

Table 4.12: Parameter settings for test data sets from the literature

The results for the first **scenario req_rapide_x_450_24** with the lowest stress level are visualized in Figures 4.24, 4.25 and 4.26. In all five instances (x = 1..5), a better performance of *MNS*, compared with the *Assignment based procedure*, can be observed: on average, travel time is reduced by 20%, delay by 68% and overtime by 59%.

In comparison with the *benchmark* results from literature, however, the results generated by *MNS* are outperformed itself. Especially in travel time, the benchmark results are on average 11% better than MNS. At least in categories delay and overtime, MNS achieves three improvements of the benchmark results: for instances x = 1 and 2, delay and overtime are reduced to zero; for instance x = 4, the benchmark result is undercut by 50% in delay and by 60% in overtime.

The results for the second **scenario req_rapide_x_240_24** - with medium stress level - are visualized in Figures 4.27, 4.28 and 4.29. A comparison of *MNS* and the *Assignment based procedure* again results in significant advantages of MNS: on average, travel time is reduced by 15%, delay by 38% and overtime by 29%. The Assignment based procedure only achieves slightly better delay and overtime values in one instance (x = 5); travel time, however, remains on a definitely inferior level.

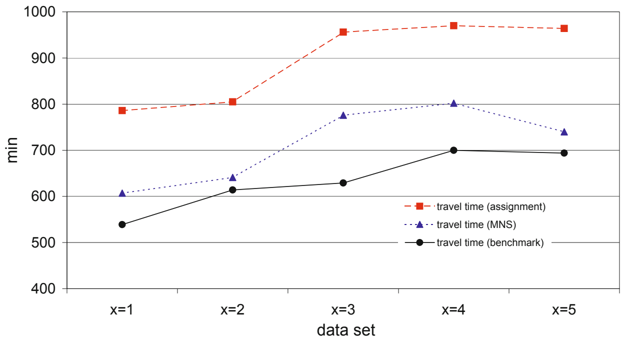


Figure 4.24: Travel time for test scenarios *req_rapide_x_450_24*

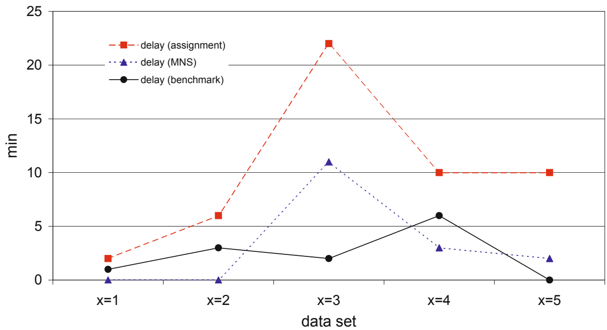


Figure 4.25: Delay for test scenarios *req_rapide_x_450_24*

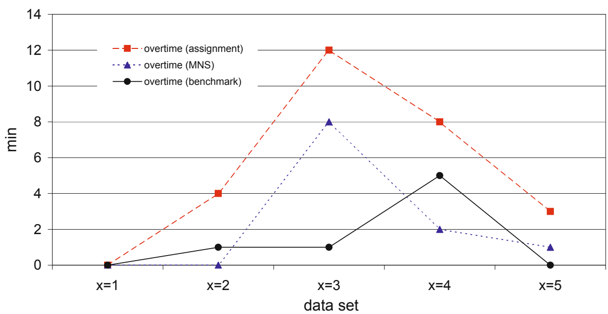


Figure 4.26: Overtime for test scenarios *req_rapide_x_450_24*

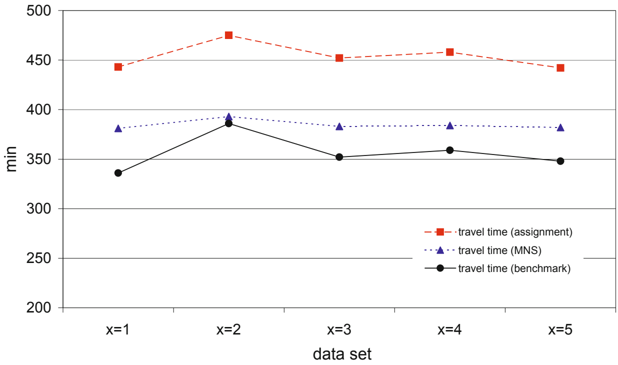


Figure 4.27: Travel time for test scenarios *req_rapide_x_240_24*

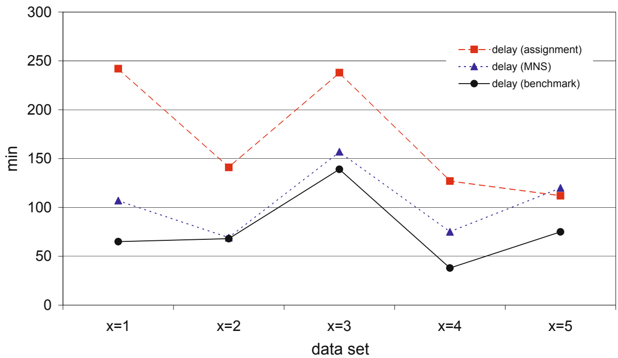


Figure 4.28: Delay for test scenarios *req_rapide_x_240_24*

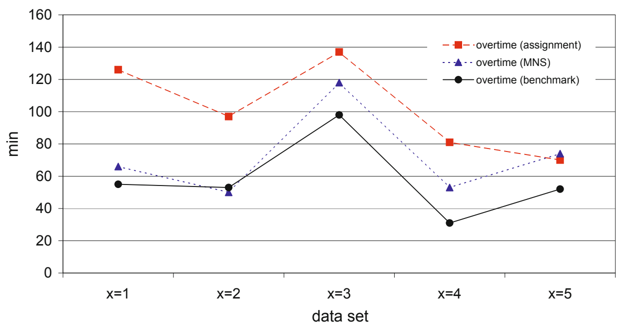


Figure 4.29: Overtime for test scenarios *req_rapide_x_240_24*

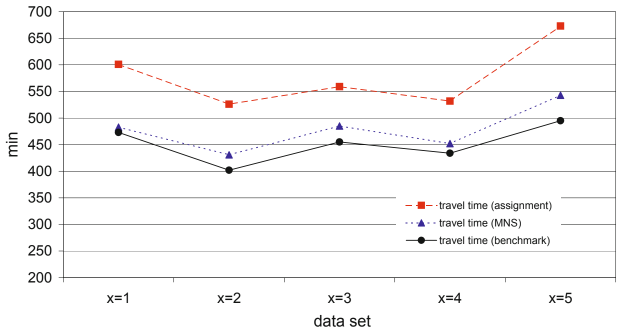


Figure 4.30: Travel time for test scenarios *req_rapide_x_240_33*

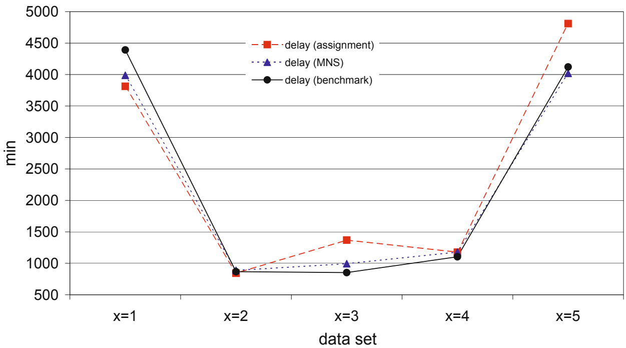


Figure 4.31: Delay for test scenarios *req_rapide_x_240_33*

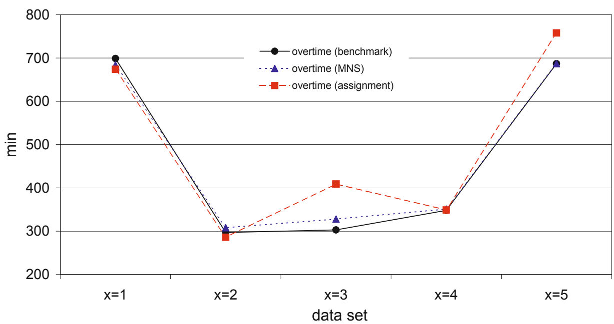


Figure 4.32: Overtime for test scenarios *req_rapide_x_240_33*

As in the first scenario, the *benchmark* results cannot be reached on average. The travel times reported by Gendreau et al. (2006) are still 7% better than the *MNS* results. For instance x = 2, the MNS results for travel time, delay and overtime are at least on a par with the benchmark.

The last **scenario req_rapide_x_240_33** exhibits the highest stress level. The associated results are presented in Figures 4.30, 4.31 and 4.32. The comparison of *MNS* and *Assignment* results still shows better performance of MNS, but MNS's advantage is diminishing. On average, travel time is better by 17%, delay by 7% and overtime by 4%. In three scenarios (x = 1,2 and 4), the Assignment results in delay and overtime reach the same or a better level than MNS.

In comparison with the *benchmark* results, *MNS* generates absolutely competitive results (cp. Figure 4.33). The average delay is 2.4% better (!) than the results published by Gendreau et al. (2006). Average travel time and overtime are only 6% and 1% worse, respectively. For instances x = 1,2,4, and 5, MNS achieves the same or even better results in delay and overtime.

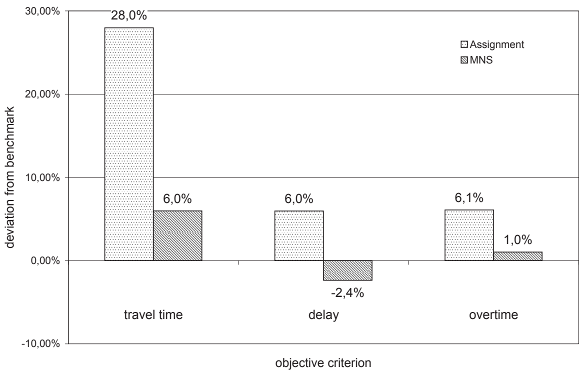


Figure 4.33: Relative comparison of results for test scenarios *req_rapide_x_240_33*

In summary, for the comparison of MNS and the Assignment based procedure, a clearly better behavior of MNS can be observed. For all 15 instances, MNS produces much better results in travel time and for 12 instances, there are also better results in delay and overtime. The best results of the Assignment based procedure are achieved for scenario 3, for which the parameterization was performed. The gap of the results between MNS and Assignment increases all the more as the test data differ from the parameterized instance. Consequently, the solution quality produced by Assignment drops from scenario 3, over scenario 2, to scenario 1. MNS is much more capable of persisting on a data set with differences to the parameterized one.

Beyond that, MNS is able to attain results on a par with the benchmark results. For the parametrized scenario 3, competitive results or even better results are generated. To put these results in perspective, it should be mentioned that Gendreau et al. (2006) applied much more computation power (parallel computing network with 16 processors) to achieve the benchmark results.

4.6 Selection of a Procedure for the Real-Life Application

At this point, the results of both data sets have been presented. Now, the question of what procedure to adapt to the real-life scenario in Chapter 5 arises. In the following, the pros and cons of both procedures are discussed, and one is finally selected.

Comparing the solution quality, the conclusions for the two test data sets do not coincide exactly. For the self-generated data sets, better **performance (in the objective function criteria travel time, delay and overtime)** of the Assignment based procedure is observed, while for the benchmark data from the literature, the contrary is true (better performance of MNS). How can that be explained?

The most plausible reason is the use of capacitated vehicles in the first and of uncapacitated vehicles in the second test data set. MNS seems to be better in the uncapacitated context and the Assignment based procedure in the capacitated case. This assumption, is supported by the findings of scenario 6 of the self-generated test data set. Here, MNS is able to achieve superior results in the case of extended vehicle capacities of 6 and 7 units, whereas better results for the low capacity instances (1 up to 4 units) are achieved by Assignment.

In terms of **robustness**, an advantage of MNS can be detected for both data sets. While the Assignment based procedure generates better results on average for the self-generated test scenarios, some data variations (e.g., reduction of available vehicles) result in an inacceptable worse solution quality of the Assignment results. In such cases, MNS shows a clearly better adaptability. The same behavior can be observed for the data set from the literature. The more different the scenario is from the scenario used for parameterization, the more the solution quality decreases in comparison with MNS and the benchmark results.

A further aspect is the **parameterization** process itself. The time discrete simulation of the Assignment based procedure is shorter than the MNS simulation runs that are accomplished in real time. Nevertheless, the parameterization of the Assignment based procedure necessitates more effort than the parameterization of MNS for both test data sets. This is due to a higher number of parameters in Assignment, a strong mutual dependency of the Assignment parameters and a high sensitivity of solution quality to changes in parameter settings.

In addition, there is also the **complexity** aspect. Which procedure is better suited for illustration in practice? This is a subjective assessment. In this work, lower complexity is associated with MNS. This is attributed to its very intuitive planning steps and the relatively insensitive parameters (compared to Assignment).

The discussed pros and cons are summarized in Table 4.13: Plus and minus signs indicate better and worse assessment in the associated category, respectively.

Except for the category *performance on self-generated data set (capacitated),* MNS achieves all pluses. This result makes the decision of which procedure to adapt to the real-life sce-

MNS	Assignment
self-generated test data set (capacitated)	
performance (objective function criteria):	
−	+
robustness:	
+	−
test data set from literature (uncapacitated)	
performance (objective function criteria):	
+	−
robustness:	
+	−
parameterization effort	
+	−
complexity of procedure	
+	−

Table 4.13: MNS or Assignment? - pros and cons

nario rather difficult. In particular, because the real-life scenario is a strictly capacitated problem (SLPDPTW).

Thus, the application of the Assignment based procedure, which performed very well for such capacitated scenarios, seems to promise "better results". On the other hand, there may be some problems with robustness, especially in situations with changing time window characteristics, in situations of vehicle scarcity, or with general inhomogeneity in the real-life data.

MNS, in contrast, is expected to produce only "good" results, inferior to the Assignment ones. However, MNS promises a higher level of robustness. The risk of a complete failure, due to unknown variations of the data set, is reduced. The aspects of an easier parameterization and of a better illustration to people in practice should also not be dismissed.

Finally, after a detailed weighing up of the arguments, the safe way is preferred, selecting MNS for adaptation to the real-life scenario.

Chapter 5

Real-Life Application at an Internationally Operating Freight Forwarding Company

At the beginning of this chapter, a detailed analysis of the general requirements for long-haul transportation in Europe is conducted (Section 5.1). Afterwards, the specific planning situation at the cooperating freight forwarding company is investigated (Section 5.2). In Section 5.3 the main adjustments applied to the existing MNS procedure are outlined. In Section 5.4 the preprocessing of the available real-life test data set and the derivation of benchmark objective function values are illustrated. Finally, computational results, generated with the adapted MNS procedure for the real-life test data set are presented (Section 5.5).

5.1 General Requirements for Long-Haul Transportation in Europe

This section introduces the main requirements that have to be considered for planning long-haul transportation tasks in Europe. For this purpose, four important categories of requirements are analyzed:

- Social Regulation[12] EC 561/2006,
- Social Regulation AETR,
- Directive[13] EC 2002/15 on working hours, and
- General Driving Bans.

The first three aspects are *driver related*, while the fourth aspect is of general type.

[12] A *regulation* (German: *Verordnung*) immediately becomes effective, without explicit transfer to national law. (Ministry of Social Affairs, Baden-Württemberg, 2010)

[13] A *directive* (German: *Richtlinie*) does not become effective immediately. It has to be transferred to national law first. (Ministry of Social Affairs, Baden-Württemberg, 2010)

Steffen Schorpp, *Dynamic Fleet Management for International Truck Transportation,*
DOI 10.1007/978-3-8349-6675-9_5,

5.1.1 Social Regulation EC 561/2006

In March 2006 the European Parliament and the Council of the European Union passed the new Social Regulation EC 561/2006 (in the following: EC 561) in order to harmonize driving and rest period restrictions in the European Union. EC 561 came into effect on 11.04.2007. All subsequent *article references* refer to the document: European Union (2006b).

EC 561 was motivated by the fact that it was possible with the former regulation "to schedule daily driving periods and breaks so that a driver could drive for too long without a full break, leading to reduced road safety and a deterioration in the driver's working conditions *(legislation preamble (16))*." This is supported by findings of the European Safety Council, whereupon driver over-fatigue is responsible for 20% of commercial road transport crashes (Kok et al., 2009).

The new regulation is applied to carriage by road, when the gross vehicle weight, including any trailer, exceeds 3.5 tonnes. It is applied to transports, undertaken either exclusively within the European Union or between the European Union, Switzerland and the countries party to the Agreement on the European Economic Area (Norway, Liechtenstein and Iceland).

However, for cross border transports between countries with EC social regulation and countries having ratified the AETR (European Agreement concerning the Work of Crews of Vehicles engaged in International Road Transport, e.g. Russia, Belarus, Albania, and Turkey), AETR must be applied for the whole journey – and not EC 561. Figure 5.1 shows a map with the European countries and their respective membership in EC 561 or in AETR.

Figure 5.1: Application of EC 561 and AETR

The most important aspects of EC 561 will be introduced in the following, differentiated between single and team driver mode. For this purpose, short definitions, basic rules and exceptions are outlined for the categories *daily driving time*, *weekly and fortnightly driving time*, *breaks*, *daily rest period*, and *weekly rest period*.

Single Driver Mode

- **Daily driving time:** The daily driving time is defined as the total accumulated driving time (measured with a digital tachograph) between the end of one daily rest period and the beginning of the following daily rest period. The word “daily” is not necessarily congruent with a weekday. Therefore, the actual meaning would be better represented by the word “interval”. Daily driving time is limited to 9 hours (basic rule) and may be extended to 10 hours twice a week (exception).
(article 8(2), article 4k)

- **Weekly and fortnightly driving time:** The weekly driving time is counted for a real week (from 0:00 on Monday until 24:00 on Sunday). It must not exceed 56 hours (basic rule). Furthermore, the total driving time during two consecutive weeks must not exceed 90 hours (basic rule). There is no exception available.
(article 6(2,3))

- **Breaks:** A break is defined as a non driving period. After a driving period of 4.5 hours, a driver shall take an uninterrupted break of at least 45 minutes (basic rule), unless he takes a rest period. The break may be split into a break of 15 minutes, followed by a break of 30 minutes (exception).
(article 7)

- **Daily rest period:** The daily rest period is a time between two daily driving time intervals, during which a driver may freely dispose of his time. It regularly has a duration of at least 11 hours (*regular daily rest period*, basic rule). Three times a week, however, it may be reduced to at least 9 hours (*reduced daily rest period*, exception). For this reduction no compensation is required. The combination of an uninterrupted 3-hour rest period with a subsequent 9-hour rest period is also counted as a *regular daily rest period.*
(article 4,g)

  In addition, a *24-hour-rule* has to be considered: Within each period of 24 hours after the end of the previous daily rest period, a driver shall *have taken* a new daily rest period. This means that for a *regular daily rest period* (*reduced daily rest period*) the daily rest period must be started after a 13-hour (15-hour) interval of activities (driving, loading, breaks, waiting, etc.) at the latest.
(article 8(2))

  Figure 5.2 shows two ways of scheduling driving times and daily rest periods: scheduling with basic rules and scheduling with exceptions.

- **Weekly rest period:** The weekly rest period is a time in between two weekly sequences of driving intervals and daily rest periods, during which a driver may freely dispose of his time. The weekly rest period has a regular duration of at least 45 hours (*regular weekly rest period*, basic rule), but may be shortened to a minimum

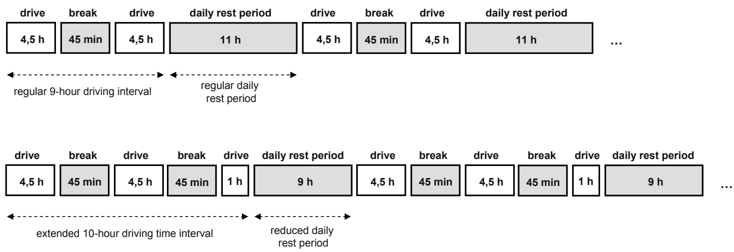


Figure 5.2: Basic rule scheduling vs. scheduling with exceptions

of 24 hours (*reduced weekly rest period*, exception).

Full compensation is needed for this reduction, which means that a 21-hour rest period has to be attached (en bloc) to another (either daily or weekly) rest period of at least 9 hours. The compensation has to be accomplished before the end of the third week after the reduced weekly rest period. See Figure 5.3 for illustration.

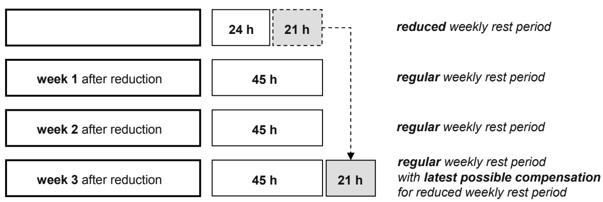


Figure 5.3: Latest compensation for a reduced weekly rest period

In addition: two reduced weekly rest periods must not succeed each other. Therefore, the maximum number of "open, not compensated" reduced weekly rest periods may be *two*, before compensation of the first reduced weekly rest period has to be accomplished at the latest. Generally, a weekly rest period shall not start later than six 24-hour periods from the end of the previous weekly rest period.
(article 4h, article 8(6))

When the crew consists of **two drivers**, modified restrictions have to be considered for the scheduling. The five previous categories are reviewed for this new adjusted situation, focusing on the differences to the single driver mode.

Team Driver Mode

- **Daily driving time:** The maximum daily driving time is now 18 hours (9 hours for each driver, basic rule). Again, each driver is allowed to use two weekly 10-hour extensions (exception). This may result in an interval driving time of up to 20 hours if both drivers use one 10-hour extension in the same interval.
- **Weekly and fortnightly driving time:** The weekly driving time must not exceed 112 hours (56 hours for each driver, basic rule) and the total driving time during

two consecutive weeks must not exceed 180 hours (90 hours for each driver, basic rule). Exceptions are not available for this category.

- **Breaks:** After each driving period of 4.5 hours, a change of driver operating the steering wheel is required. Times as a co-driver in a moving vehicle are counted as breaks, therefore an explicit vehicle stop of 45 minutes is no longer required.

- **Daily rest period:** Every daily rest period must have a duration of at least 9 hours. There is no more differentiation between *regular* and *reduced*, and hence, no more exceptional rule.

  The *24-hour-rule* is changed to a *30-hour-rule*: Within each period of 30 hours after the end of the previous daily rest period, both drivers shall *have taken* a new daily rest period. This means that the daily rest period has to be started after a 21-hour interval of activities (driving, loading, breaks, waiting, etc.) at the latest. *(article 8(5))*

- **Weekly rest period:** The rules are identical to the single driver mode.

Not all European transport activities are covered by EC 561, some are covered by AETR. According to the Ministry of Social Affairs, Baden-Württemberg (2010), an adaptation of AETR to EC 561 standards is planned. Nevertheless, AETR is still applied today and will therefore be considered in the following.

5.1.2 Social Regulation AETR

AETR regulations are similar to EC 561. The following section summarizes the most important aspects (cp. United Nations Economic Commission for Europe, 2006), in particular the differences to EC 561 (cp. Table 5.1). All aspects not explicitly mentioned are identical to EC 561.

In AETR, a **driving time** limit per week is not given, just a 90-hour limit for the total travel time of two subsequent weeks *(article 6(1))*. Regulation of **breaks** includes the *additional option of splitting* a 45-minute break into three breaks of at least 15 minutes *(article 7(2))*.

Daily rest periods have to be taken within each 24-hour interval with a duration of at least 11 hours and may be reduced to 9 hours three times a week. In contrast to EC 561, *compensation before the end of the following week is mandatory*. The daily rest period may be split into *three separate rest periods, with one part of at least 8 consecutive hours*. If the daily rest period is split, the total minimum length increases to 12 hours *(article 8(1))*. For team driver mode, AETR specifies *just 8 consecutive hours of daily rest period* within a 30-hour period, instead of 9 hours in EC 561 *(article 8(2))*.

Weekly rest period regulation is less restrictive in comparison to EC 561: AETR just says that within each week one rest period shall be extended to 45 hours. An explicit maximum time gap between two subsequent weekly rest periods is not specified, just the requirement that the next weekly rest period shall be scheduled after six *daily driving periods (article 6(1))*.

EC 561	AETR
Driving time limit per week: 56 hours, fortnightly 90 hours	No driving time limit per week, fortnightly 90 hours
Splitting option for 45-minute break: 15 min + 30 min	Splitting option for 45-minute break: $3 \cdot 15$ min
Reduced daily rest period: no compensation	Reduced daily rest period: mandatory compensation, until the end of the following week
Regular daily rest period: splitting option 3 hours + 9 hours	Regular daily rest period: splitting option into three parts $x + y + z = 12$ hours, with one part ≥ 8 hours
Team driver daily rest period: 9 hours	Team driver daily rest period: 8 hours
Maximum time gap between weekly rest periods: six 24-hour intervals	Maximum time gap between weekly rest periods: six daily driving periods
Not two subsequent reduced weekly rest periods	Two subsequent reduced weekly rest periods allowed
Maximum number of "open, not compensated" reduced weekly rest periods: 2	Maximum number of "open, not compensated" reduced weekly rest periods: 3

Table 5.1: Differences between EC 561 and the AETR

Suppose a truck in team driver mode: a *daily driving period* lasts 30 hours, which results in *six daily driving periods* lasting 180 hours. In addition, between driving periods possibly some additional waiting times may be scheduled, so a weekly rest period may be taken much later than six 24-hour intervals (144 hours) after the previous one.

A reduction of a weekly rest period to 24 hours is possible, requiring (identically to EC 561) a 21-hour compensation before the end of the third following week *(article 8(3))*. However, scheduling *two subsequent reduced weekly rest periods* is not strictly prohibited, which allows for a maximum number of three "open, not compensated" reduced weekly rest periods.

Social regulations EC 561 and AETR in particular deal with *driving time*. However, depending on the degree of other activities, it may be possible that *restrictions on total working time* also become stringent.

5.1.3 Directive EC 2002/15 on Working Hours

Driving time regulations are supplemented by the general EC Directive on working hours "for persons performing mobile road transport activities" (*EC 2002/15*, see European Union, 2002), which is transferred to German law in *§21a Working Time Act* (Federal Ministry for Labor and Social Affairs, Germany, 2009).

Here, working time does not only involve driving time, but also loading and unloading time, cleaning and technical maintenance, and all other work intended to ensure the safety of the vehicle and its cargo or to fulfil the legal or regulatory obligations (e.g., customs), etc. In addition, the regulation also includes times during which a driver cannot dispose freely of his time (e.g., during periods awaiting loading or unloading, where their foreseeable duration is not known in advance)”. *(article 3,a)*

The main aspects of regulation EC 2002/15 are outlined in the following:

- **Weekly working time:** A week is defined identically to EC 561, ranging from Monday 0:00 to Sunday 24:00. Maximum weekly working time is limited to 60 hours. In addition, over a four-month horizon, the average weekly working time shall not exceed 48 hours.
(article 4,a)

- **Breaks:** A break has to be scheduled after 6 hours of consecutive working time. The length of the break must be 30 minutes, if working hours total between six and nine hours. The length of the break must be 45 minutes, if working hours total more than nine hours. Breaks may be subdivided into periods of at least 15 minutes each.
(article 5)

- **Night work:** The night is defined as the time between 0:00 and 7:00. If at least four working hours fall during this time interval, it is referred to as *night work*. If night work is performed, the daily working time shall not exceed ten hours in each 24-hour period.
(article 7,1)

Since 23.03.2009, EC 2002/15 also contains “self employed (independent) drivers” who perform transports with their own vehicle. This modification, however, has not yet been adapted to German law (Vogel, 2010).

5.1.4 Traffic Bans

The last main group of regulations for scheduling vehicles is traffic bans *at weekends*, *public holidays* and *special annual periods*. This group is not covered by a common regulation in the European Union. Instead, each country has its own regulation, from “relatively strict” up to “nonexistent”, making the situation quite complex.

In Germany, for example, vehicles with more than 7.5 tonnes gross vehicle weight are not allowed to drive on Sundays between 0:00 and 22:00. Exceptions only exist for vehicles transporting fresh and perishable products, as well as pre- and post-carriage of multi-modal transports. Additional traffic bans exist on public holidays from 0:00 until 22:00 and on Saturdays in the summertime (03.07.-28.08.) from 7:00 until 20:00 on selected highways (cp. Vogel, 2010).

The European countries can be roughly classified into three groups:

(i) with general traffic bans,

(ii) with partial traffic bans, and

(iii) without traffic bans.

Figure 5.4 shows the ten countries (Germany, France, Czech Republic, Switzerland, Liechtenstein, Austria, Slovakia, Croatia, Italy, and Slovenia) of group (i) and their main aspects of regulation. In order to keep track and due to the variety of individual rules and exceptions, only a very aggregated view is chosen.

Germany **Sunday** and public holidays **0:00 – 22:00**, special summer holiday restrictions	Switzerland/Liechtenstein **Sunday** and public holidays **0:00 – 24:00**, general night traffic ban 22:00 – 5:00	Croatia **Sunday** and public holidays **12:00 – 23:00**, previous day of public holiday 15:00 – 23:00
France **Sat 22:00 – Sun 22:00** and public holidays (previous day 22:00 – public holiday 22:00), special route restrictions Paris	Austria **Sat 15:00 – Sun 22:00** and public holidays 0:00 – 22:00, special summer holiday restrictions	Italy **Sunday** and public holidays **8:00 – 22:00** (Oct.-May), **7:00 – 24:00** (June-Sep.), special summer holiday restrictions
Czech Republic **Sunday** and public holidays **13:00 – 22:00**, special summer holiday restrictions	Slovakia **Sunday** and public holidays **0:00 – 22:00**, special summer holiday restrictions	Slovenia **Sunday** and public holidays **8:00 – 21:00**, special summer holiday restrictions

Figure 5.4: European countries with general traffic bans

Countries of group (ii) with less restrictive and less general partial traffic bans are: Poland, Luxembourg, Hungary, Portugal, Spain, Bulgaria, Romania, Greece, and Turkey. These countries only have traffic bans for public holidays, selected highways, special annual periods or for hazardous goods transportation.

In group (iii), there are (more or less) no such regulations at all. Countries belonging to this group are: Sweden, Norway, Finland, Denmark, The Netherlands, Belgium, United Kingdom, Ireland, Ukraine, Belarus, Russia, Estonia, Latvia, Lithuania, Malta, Cyprus, Albania, Macedonia, Bosnia-Herzegovina, Serbia and Montenegro.

5.1.5 Inspection of Compliance

The driver-related restrictions are controlled with the help of a digital driver card (see Figure 5.5). For every traveling activity of a vehicle, such a digital driver card has to be logged in to the vehicle's digital tachograph. The tachograph writes information about driving time, breaks, other working time, and rest periods, etc. onto the digital driver card. This information is stored on the card for 28 days, then it is overwritten by new information. Thus, a freight forwarding company must download data from the digital driver's cards at least every 28 days and store each driver's activity log for one year.

According to Directive EC 2006/22 (European Union, 2006a) "on minimum conditions for the implementation of EC 561", random checks have to be performed both at the freight forwarding company and directly on-the-road. In Germany, checks at the freight forwarding companies are performed by local *commercial regulatory authorities (German: Gewerbeaufsichtsämter)*, while checks on-the-road are carried out by the *police* and the *Federal Office for Goods Transport (German: Bundesamt fuer Güterverkehr - BAG)*. Starting from 01.01.2010, at least 3% of days worked by drivers' shall be controlled, with not less than 30% of the total number (of checked working days) being checked at the roadside and not less than 50% being checked at the premises of freight forwarding companies.

Figure 5.5: Digital driver card (Kraftfahrtbundesamt, 2010)

According to the Ministry of Social Affairs, Baden-Württemberg (2010), infringements against EC social regulations are penalized in the range of € 5000 up to a maximum of € 15000. However, this doesn't mean that every small violation is penalized: especially in situations where a driver cannot find a suitable stopping place or where a traffic jam causes delay, this is handled with courtesy. In the most cases of penalty, systematically recurring infringements are punished (Commercial Regulatory Authority, 2010).

In Germany, a non-compliance with the Sunday or public holiday traffic ban is penalized with € 75 for the driver and € 380 for the vehicle owner (Vogel, 2010).

5.1.6 Some Recent Literature

The incorporation of the introduced general requirements for International Truck Transportation into tour planning algorithms has only been considered, by some of the latest publications. All of those publications deal with static problem definitions and only partially include the actual real-life requirements.

One of the first publications considering EC 561 is Goel (2009). The author presents two algorithms for the VRP with hard time windows which comply with the *basic rules* of EC 561 (only single driver mode). *Exceptions* are completely neglected "in order to increase the robustness of the generated plan": it is argued that exceptions are of particular importance if delays, e.g., due to bad traffic conditions, result in longer driving times than expected.

While the first algorithm uses "naive" scheduling (break and rest periods are scheduled when the respective accumulated driving time is exhausted), the second procedure takes into account the fact that it can be beneficial to schedule rest periods before the respective accumulated driving time is exhausted. The primary objective of both approaches is to minimize the number of vehicles, the secondary objective is to minimize total traveled distance.

For testing purposes, Goel modifies the Solomon (1987) data set: the planning horizon is extended to six days, a handling time of one hour per customer is introduced, and the average vehicle speed is increased to five units per hour. Results show a better performance of the second procedure. However, a drawback of the used data set is its short planning horizon of only six days, excluding any tests of scheduling weekly rest periods.

Kok et al. (2009) also deal with the incorporation of EC 561 into a VRP with hard time windows. They propose an extended Dynamic Programming Heuristic (only single driver mode), with two implemented versions: one considers all basic rules of EC 561, but no exceptions (*basic method*), the other additionally considers the exceptions. In a first step, the authors benchmark the *basic method* with the results published by Goel (2009), using the same modified Solomon (1987) data set. Results show a significant reduction in the average number of vehicles (-18.26%) and in the average traveled distance (-5.41%) compared to Goel (2009).

In a second step, the authors compare the different versions of their algorithm with each other. Further improvements are reported when considering the exceptions of EC 561: the number of vehicles is reduced by 4.28% and the total distance traveled by 1.54% (in comparison to the basic method). It is also mentioned that computation time more than doubled when incorporating the exceptions, which is attributed to additional checks. Finally, a version of the algorithm is tested extending the *basic method* by EC 2002/15 on working hours. This leads (in comparison to the basic method) to slightly worse results: the average number of vehicles increases by 4% and the average distance traveled increases by 1%. This is an intuitive finding, since additional restrictions are added to the planning problem.

In the following section, the general restrictions for International Truck Transportation are supplemented by a look at the actual planning situation at a freight forwarding company. This exemplary real-life planning situation is used as indication of how a practical adaptation of the dynamic MNS procedure should look.

5.2 Exemplary Real-Life Planning Situation at a Freight Forwarding Company

The considered freight forwarding company distinguishes national and international transportation tasks and allocates the associated responsibilities to separate divisions. This makes it easier to regard only the international activities which consist of transportation tasks over the entire European territory.

Most of the company's international **requests** belong to the group of *occasional transportation tasks* (tramp transportation, independent of predefined networks), a smaller part of recurring requests can be attributed to the group of *line haul tasks*. The majority of requests are of type Full Truckload between a Pickup and a Delivery location (P $\rightarrow$ D).

In addition, it is also possible that one request may possess several load and unload locations (subsequently labeled as "request bundle"). Such request bundles may be predefined by a given customer order or created by a dispatcher, who merges two or more compatible requests manually. The only logical requirement for request bundles is the precedence constraint of scheduling a Pickup before the associated Delivery. Apart from that, all combinations are allowed: e.g. P1 $\rightarrow$ P2 $\rightarrow$ D1 $\rightarrow$ D2, P1 $\rightarrow$ D1 $\rightarrow$ P2 $\rightarrow$ D2 or P1 $\rightarrow$

P2 → D1 → P3 → D3 → D2. In total, approx. 40% of all orders are part of a bundle. The sequence within a bundle is fixed and shall not be changed in the optimization process.

Further request attributes are: soft time windows for Pickup and Delivery location, geographical coordinates for Pickup and Delivery location, required vehicle type, information about the need for hazardous goods equipment in the vehicle, and duration of load and unload process.

The company employs approximately 1200 vehicles of different type (see Section 5.4 for details). The available **vehicles** are registered in different European countries. All vehicles are equipped with a digital tachograph. Main vehicle attributes are the vehicle's type and the information about whether equipment for transportation of hazardous goods is carried. Both aspects are crucial requirements for a feasible load-to-vehicle assignment. A vehicle is operated by one or two drivers, which may be subject to change. On average, approx. 70% of vehicles are operated in single driver mode and approx. 30% in team driver mode.

The **drivers** come from 19 European countries, with a focus on Eastern Europe. Driver scheduling directly influences the tour planning. This is explained as follows:

- An international driver, who usually spends several weeks en route, may want to get home occasionally for a holiday. So, at the end of his *operating time*, the dispatcher has to find a request with target location preferably near to the driver's home location. If this is not possible, drivers may also be exchanged at several European meeting points alternatively, from where they are brought home by minivans. However, unfavorable far away exchange points may cause higher costs for bringing the driver home and may also result in driver dissatisfaction.
- Furthermore, EC social regulation has to be observed. Especially in the one driver mode, fortnightly maximum driving time and the compensation for a reduced weekly rest period can be quite restrictive. Therefore, it is an incentive for the freight forwarding company to frequently exchange drivers (every two or three-week interval).

  Each time, the *new driver* starts without any outstanding compensation time for weekly rest periods and with an unconsumed travel time account. The *old driver* takes his outstanding rest period near the meeting point or at home (by any means: separated from the vehicle).

  By performing this kind of driver exchange, the vehicle's utilization can be significantly increased.

Subsequently, the **planning process** is described in detail.

The planning process is subdivided (see Figure 5.6). In a first *acquisition step*, an acceptance/rejection decision is made (level 1). This also includes the active solicitation of requests.[14] In a second step, the actual *tour planning* including order-to-vehicle-assignment

[14] In the literature, the acceptance/rejection decision is distinguished from the cost/pricing problem: in the first case, the reward to be obtained is known and in the second case it is unknown. At the considered freight forwarding company both types of problems occur. However, since this aspect is not the focus of this work, we will only speak of an acceptance/rejection decision in the following.

and detailed vehicle scheduling is performed (level 2).

Coordination between both planning steps is reached by pricing: level 2 (tour planning) provides internal prices for transport relations, which, for example, include the probability of getting an *add-on request* in a specific target region. Prices may be changed on a daily basis, depending on the current planning situation. In addition, specific needs for add-on requests may be communicated directly between both planning levels.

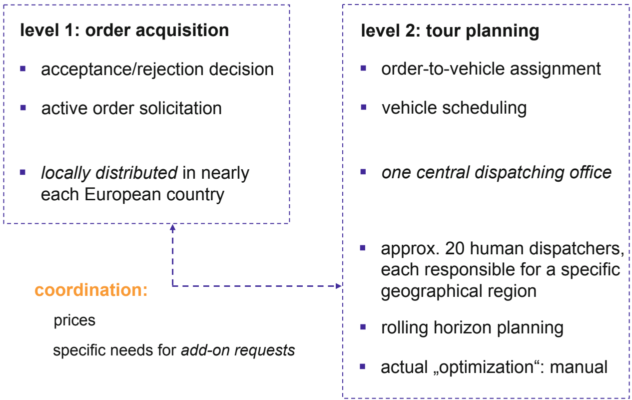


Figure 5.6: Planning process at the cooperating freight forwarding company

Acquisition is not performed at one central geographical place, but locally distributed in nearly every European country. This involves the advantage of being close to the local market with all its specific characteristics and languages. Tour planning, however, is executed at a central dispatching office: here, approximately 20 human dispatchers perform level 2 tasks, in which each dispatcher is reponsible for a specific geographical region coordinating inbound and outbound flows (that means the dispatcher has to find suitable *add-on requests* for incoming vehicles, e.g., with the objective of producing minimal unloaded traveled distance and minimal vehicle idle time).

Currently, “Decision Support” is contributed by only several “Information Systems”: a *database* (handling order data, vehicle data, etc.), a *route planner* (proposing route options with shortest travel distance/shortest travel time), and a *graphical representation of vehicle scheduling*. The actual “optimization” task of assigning orders to vehicles and subsequent vehicle scheduling, however, is performed completely manually in a rolling horizon approach.

While the first part of a day is basically used for monitoring the existing plan and conducting adjustments due to unforeseen events, like traffic jams, vehicle break downs, extended loading times or changes in order specifications, the second part of the day is used for incorporation of new requests into the plan. The specific scheduling for the following day has to be completed by approx. 18:00. EC social regulations are only considered indirectly, e.g. by maximum daily kilometer performances of 540 km - 600 km for a vehicle in single driver mode and of 900 km - 920 km for a vehicle in team driver mode.

The obtained insights at the considered freight forwarding company – planning data and planning process – serve as guidelines for the adjustments that have to be applied to the original MNS procedure of Section 4.1 in order to cover the real-life requirements for International Truck Transportation.

5.3 Adjustment of Multiple Neighborhood Search

This section describes the main *general* and *company specific requirements* that are actually incorporated into MNS in order to handle the real-life planning problem. The resulting adaptations primarily affect the MNS scheduling, whereupon the general planning process is not touched. To allow for a proper representation of the new scheduling activities, the internal data structure is revised. In addition, the program modules *Best Insertion* and *Scheduling* are adapted to the new planning requirements of the extended real-life SLPDPTW.

5.3.1 Selected General Requirements

In a first step, the **general requirements** that have actually been chosen for inclusion in the MNS procedure are presented. These are basically EC 561, EC 2002/15 on working hours and traffic bans (see Table 5.2).

considered requirements
EC 561: basic rules and exceptions
Single Driver Mode
Team Driver Mode
EC 15/2002: working time restrictions
partial: Team Driver Mode
traffic bans
general Sunday traffic ban
0:00 - 24:00 for <u>all</u> countries

Table 5.2: Summary of general real-life restrictions included in MNS

Consideration of *EC 561* is a mandatory task, since the legislator explicitly specifies in *article 10(4)* (of *EC 561*) that “freightforwarders shall ensure that transport time schedules respect this regulation.” In contrast to the actual planning at the cooperating freight forwarding company, where a tour’s admissibility is only estimated, the adapted MNS procedure will produce an explicit scheduling in accordance with the given requirements.

All *basic rules* of EC 561 are incorporated[15]. In addition, nearly all *exceptions* to EC 561 are also integrated, except for the splitting option of the 45-minute break. This is

[15] We should point out that the *fortnightly driving time restriction for single driver mode* is relaxed for the computation of our final results. This is due to driver exchanges in the manual benchmark planning performed at our cooperating freight forwarding company, which lead to (permissible) fortnightly vehicle driving times in single driver mode of over 90 hours. Real-life MNS does not include explicit driver exchange and therefore performs planning with the same driver over the entire five-week planning horizon. The resulting disadvantage is partially compensated by the specified relaxation.

due to the fact that international (long distance) transportation tasks last two days on average, involving only a few loading and unloading activities. Thus, the advantage of scheduling split breaks seems to be negligible. While 93% of all transportation tasks (of the cooperating company) fall in the category of EC 561, approximately 7% involve *AETR regulation*. Since AETR regulations have been proved to be quite similar to EC 561 and are to be adapted to EC 561 soon, EC 561 is applied for all transportation tasks.

Working time restrictions (EC 2002/15) are considered partially: in agreement with the cooperating partner, an extra 30-minute break is scheduled in the team driver mode after 12 hours of driving. Table 5.3 visualizes all "driver-based" aspects for single driver and team driver scheduling that have been integrated into the planning procedure.

Single Driver Mode	**Team Driver Mode**
Daily driving time: - regular: 9 hours - exception: twice a week 10 hours	**Interval driving time:** - regular: 18 hours - exception: 4 extra hours per week, max. 2 extra hours per interval
Weekly driving time: - max. 56 hours a week - fortnightly: 90 hours in total	**Weekly driving time:** - max. 112 hours a week - fortnightly: 180 hours in total
Breaks: - regular: 45 min after 4.5 hours	**Breaks:** - 30 min after a driving period of 12 hours, due to working hour regulation
Daily rest period: - regular: 11 hours within 24-hour interval - exceptions: reduction to 9 hours three times a week (without compensation), splitting option 3h+9h (as regular rest period)	**Daily rest period:** - 9 hours within 30-hour interval
Weekly rest period: - regular: 45 hours - exception: 24 hours with 21-hour compensation	**Weekly rest period:** - regular: 45 hours - exception: 24 hours with 21-hour compensation

Table 5.3: Included restrictions for scheduling single and team driver mode

Furthermore, *traffic bans* are also considered: since the variability of such rules is very high between European countries, a simplified approach is chosen, just assuming a complete Sunday traffic ban (0:00 - 24:00) for all countries.

5.3.2 Selected Requirements based on the Real-Life Planning Situation

In contrast to the original version of MNS, not every order is suitable any longer to be transported by every vehicle. To handle these **assignment restrictions**, the new variables "load_type" and "adr_class" are introduced for both orders and vehicles. An order may only be assigned to a vehicle if both variables are consistent or if there is a possible substitution option. The first variable "load_type" primarily provides information about the required and offered vehicle size (for orders and vehicles, respectively). The second variable "adr_class" indicates whether hazardous goods equipment is needed (order) and whether such equipment is available (vehicle).

Due to the consideration of a **Full Truckload problem**, original order information on weight and volume consumption and vehicle information on weight and volume capacity is skipped. However, the real-life situation is not perfectly consistent to a classic Single Load problem. Some orders arrive as **request bundles**, including transportation of more than one order at the same time. In the strictest sense, this is “Multi Load” transportation. However, due to the fixed sequence within such a request bundle, there are no remaining planning options. Further load consolidation or changes in sequence are not allowed.

Hence, such a request bundle is treated as a single order. For such bundled orders, the new attributes “bundle_no”, “pos_p” and “pos_d” are introduced: “bundle_no” indicates whether an order is part of a bundle and also allows for identification of the other parts of the same bundle; “pos_p” and “pos_d” indicate the positions of the order’s Pickup and Delivery tasks within the request bundle’s sequence.

In addition, there is **no more central depot** from which a tour starts and finally ends. Hence, some information about each vehicle’s initial geographical position is required (“geo_long_initial” and “geo_lat_initial”). A final destination is not defined, since planning with **open tours** is performed. Initial geographical information is supplemented by a specific time of availability (“av_from”) that indicates from which point in time the vehicle is ready for execution of the first planning task.

Furthermore, two new vehicle attributes are introduced which indicate whether a vehicle is operated in **Single or in Team Driver mode:** “driver_one” and “driver_two”. As we have seen in Section 5.1, a different number of available drivers results in completely different requirements for vehicle scheduling. The vehicle attributes “driver_one” and “driver_two” allow for a direct reference to real drivers. In the chosen adaptation, however, an exchange of the initial driver(s) is not explicitly considered.

Tables 5.4 and 5.5 summarize all **request and vehicle attributes** of the adapted MNS version. The first column shows the *internal abbreviation of an attribute*, the second column includes a short *description*, and the third column contains a note on the *internally used data type*.

Finally, it is also worth mentioning that in contrast to the original problem setting, a delayed arrival at a Pickup or Delivery location can no longer be scheduled completely freely. Instead, a **core arrival time** has to be considered. Such a core arrival time prevents delayed arrivals at undesirable times, e.g. at 03:00. The core arrival time specifies in which time interval (different to the order’s original time window) a delayed arrival may occur.

The core interval at our cooperating company was chosen to be [7:00, 20:00]. Hence, if there is a delayed arrival at 21:00, the vehicle has to wait until 7:00 the next day to serve the location. The core interval, however, is extended if the location’s original time window already contains a boundary outside of the core interval. Suppose the Delivery time window [5:00, 9:00]: in such a case, the delayed vehicle would be allowed to service the location already at 5:00 the next day.

request attributes		
req_count	unique request identifyer	∈ Integer
call_in	the time, the request is revealed to the decision maker	∈ Date-Format
load_type	specification, what vehicle is needed for transportation of the request	∈ String
adr_class	indication (0,1), if hazardous material equipment is needed	∈ Binary
bundle_no	number, indicating, if a request is part of a bundle	∈ String
pos_p	indication of the Pickup's position within a bundle	∈ Integer
pos_d	indication of the Delivery's position within a bundle	∈ Integer
EPT	earliest Pickup time	∈ Date-Format
LPT	latest Pickup time	∈ Date-Format
geo_long_Pickup	geographical coordinate: longitude Pickup location (deg° mm'ss")	∈ String
geo_lat_Pickup	geographical coordinate: latitude Pickup location (deg° mm'ss")	∈ String
loadtime	time needed to perform loading procedure at Pickup location (in minutes)	∈ Integer
EDT	earliest Delivery time	∈ Date-Format
LDT	latest Delivery time	∈ Date-Format
geo_long_Delivery	geographical coordinate: longitude Delivery location (deg° mm'ss")	∈ String
geo_lat_Delivery	geographical coordinate: latitude Delivery location (deg° mm'ss")	∈ String
unloadtime	time needed to perform unloading procedure at Delivery location (in minutes)	∈ Integer

Table 5.4: Attributes of a request in the real-life scenario

vehicle attributes		
veh_count	unique vehicle identifyer	∈ Integer
load_type	specification, of a vehicle's load type (relevant for feasible assignment of a request)	∈ String
adr_class	indication (0,1), if vehicle carries hazardous material equipment	∈ Binary
geo_long_initial	geographical coordinate: longitude initial location (deg° mm'ss")	∈ String
geo_lat_initial	geographical coordinate: latitude initial location (deg° mm'ss")	∈ String
av_from	time, a vehicle is available for new transportation tasks	∈ Date-Format
driver_one	number, of a first driver	∈ Integer
driver_two	(optional) number, of a second driver	∈ Integer

Table 5.5: Attributes of a vehicle in the real-life scenario

5.3.3 Adjustment of Internal Data Structures

In the description of the original version of the MNS procedure, explaining details of the data structure which was used for internal representation of the vehicle scheduling was intentionally omitted. In the real-life case, however, the variety of planning options that have to be covered, combined with the general dynamic planning situation, justify a short consideration.

In this work a variably sized LinkedList is chosen for representation of each vehicle's scheduling. This list may include five scheduling elements: *Pickup*, *Delivery*, *Break*, *Wait*, and *Go*. Each element contains time information accurate to the minute. *Pickup* and *Delivery* elements additionally include information on the associated order number and the order's geographical position. Furthermore, two binary indicators ("time fixed" and "vehicle fixed") are stored with each *Pickup* and *Delivery* element, all the other elements only have a single binary indicator ("time fixed"). In a rolling horizon framework, those indicators signal what parts of the planning are open for further changes and what parts are permanently fixed.

Each element's specific meaning and the information stored with the elements is explained in the following.

- **Pickup:** This element is scheduled to indicate the arrival at a Pickup location. It contains information about what request is picked up ("req_count"), geographical coordinates of the Pickup location ("geo_long_Pickup", "geo_lat_Pickup"), actual arrival time (start of service), end of service (start of service + loading time), drive-on time, fixation indicator vehicle, and fixation indicator time.

  P;15;100225;490746;600;660;660;1;1, for example, means that the vehicle arrives at the Pickup location of request number 15 at geographical coordinate (10°02'25", 49°07'46") at system time 600. Service ends at system time 660 and the vehicle immediately drives towards another location at system time 660. Since fixation indicators are both equal to 1, this internal element must not be changed any more.

- **Delivery:** This element is scheduled to indicate the arrival at a Delivery location. It contains information about what request is delivered ("req_count"), geographical coordinates of the Delivery location ("geo_long_Delivery", "geo_lat_Delivery"), actual arrival time (start of service), end of service (start of service + unloading time), drive-on time, fixation indicator vehicle, and fixation indicator time.

  The example *D;15;113240;472209;900;960;1000;1;0* means that the vehicle arrives at the Delivery location of request number 15 at geographical coordinates (11°32'40", 47°22'09") at system time 900. Service ends at system time 960, the vehicle waits 40 minutes and subsequently drives towards another location at system time 1000. Since the first fixation indicator (vehicle fixed) is equal to 1 and the second fixation indicator (time fixed) is equal to 0, the Delivery element must not be exchanged to another vehicle, but it may be rescheduled within the current vehicle's tour.

- **Break:** This element is scheduled to indicate a non-driving period, primarily to satisfy regulation EC 561 on breaks, daily rest periods and weekly rest periods. A "Break" element is not penalized in the objective function. It contains a start time, an end time and a single fixation indicator.

B;700;745;0, for example, stands for a non-driving period, starting at system time 700 and ending at system time 745. Since the fixation indicator is equal to 0, this internal element may be rescheduled by improvement procedures.

- **Wait:** This element is scheduled to indicate a waiting period. Such a period may have several reasons: waiting time to avoid early arrival at a Pickup location, waiting time at a (fixed) Delivery location until the time window opens, or waiting time if the vehicle is idle. A "Wait" element is penalized in the objective function. Even if times of "Waiting" have another intention than "Break", they are certainly considered for compliance with EC social regulations.

  W;800;900;0, for example, stands for a non-driving period, starting at system time 800 and ending at system time 900. Since fixation indicator is equal to 0, this internal element may be changed by improvement procedures.

- **Go:** This element is scheduled to indicate a time of traveling. It is used in particular between "Break" elements. It contains a start time, an end time and a single fixation indicator.

  G;700;970;0, for example, stands for a driving period of 4.5 hours, starting at system time 700, and ending at system time 970. Since fixation indicator is equal to 0, this internal element may be changed by improvement procedures.

Certainly, there are some redundancies in this data structure. The chosen intuitive structure, however, helps to keep a clear perspective on intermediate planning results on command line level and therefore decisively simplifies the debugging process.

5.3.4 Adjustments to the MNS modules Best Insertion and Scheduling

Besides the underlying data structure, there are also major modifications to some program modules of the original MNS version of Section 4.1. The new real-life requirements result in updated versions of the modules *Best Insertion* and *Scheduling*. The applied changes are reported in the following.

The adapted workflow of module **Best Insertion** is visualized in Figure 5.7. Compared to the original Multi Load case, there are some simplifications: the number of possible insertion positions in the new Single Load case is significantly smaller, since Pickup and Delivery may be scheduled only in direct succession. Hereby, one loop is saved. In addition, recurring capacity checks within each tour have been substituted by an initial check of order-to-vehicle compatibility.

The general workflow proceeds as follows: The program module Best Insertion is called by the MNS main program, handing over a specific order number and the current fixation time. Best Insertion runs through all vehicles in an outer loop. If a vehicle passes the compatibility check, the number of open positions in this vehicle's scheduling is determined. Afterwards, a loop is started that successively inserts the new order into each possible position. For this purpose, the *Scheduling submodule* is called. If the associated

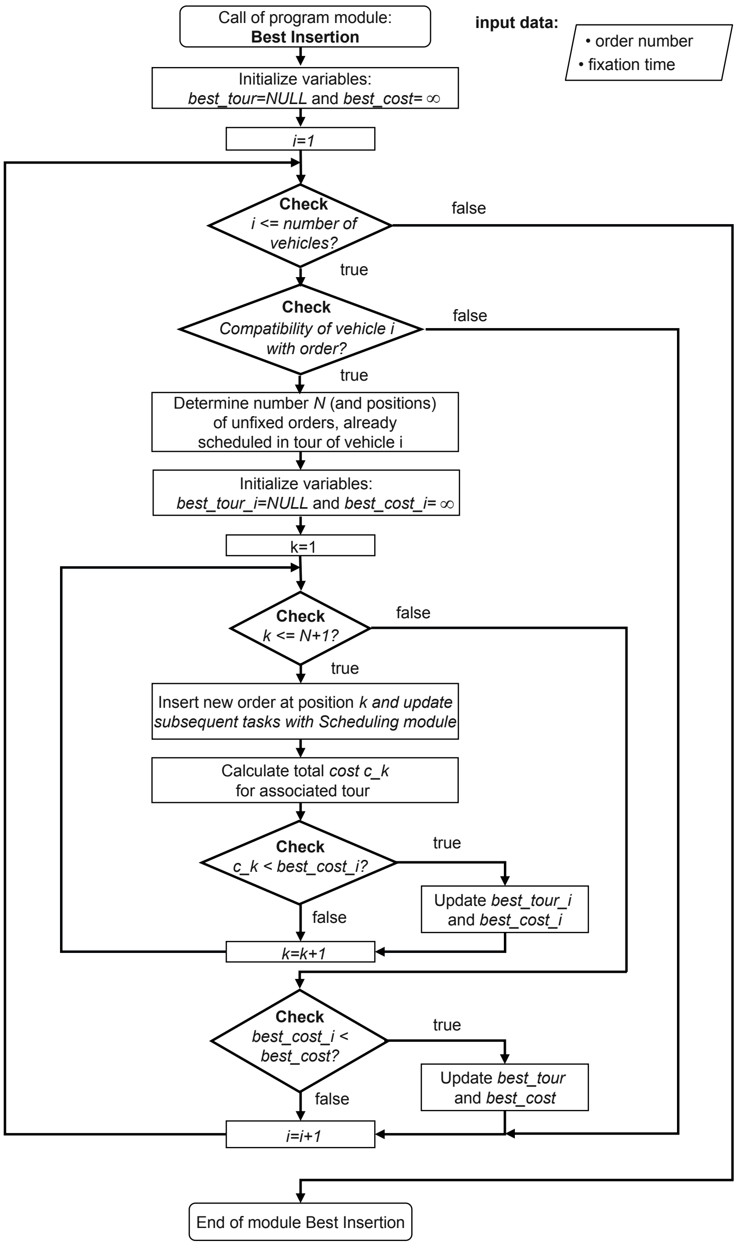


Figure 5.7: Program flow chart: Best Insertion

costs for a new scheduling undercut the currently best scheduling's cost, the vehicle's best scheduling is replaced by the new result. This investigation is repeated for all compatible vehicles. Finally, the insertion position over all vehicles and all associated open insertion positions is chosen, which generates the overall minimum insertion costs.

The workflow of the **Scheduling** module is summarized in Figure 5.8. The module is called by Best Insertion in order to produce a feasible scheduling that includes a new order at a specific position in a specific vehicle's tour. For this purpose, the following input data are handed from Best Insertion to the Scheduling module: request number, fixation time, current vehicle tour, and insertion position. In a preprocessing step, the Scheduling module analyzes, whether the vehicle is equipped with one or two drivers. Accordingly, the scheduling rules are selected (cp. Table 5.3). In addition, some auxiliary variables based on the driver's travel time history are calculated: available travel time until the next break is required, available travel time in the current travel time interval, available travel time in the current week, remaining options for scheduling with exceptions, outstanding compensations, etc.

Then, the earliest possible departure time for the new request's Pickup is calculated. Afterwards, scheduling to the first Pickup is performed using *basic regulation rules*. This is followed by a check whether the resulting arrival time conforms to the associated Pickup time window. If this is true, the Pickup's successor is scheduled. Otherwise, it is differentiated between *too early arrival* and *too late arrival*. In the case of too early arrival, an initial waiting time is scheduled at the Pickup's predecessor, so that a prompt arrival at EPT is ensured. In the case of too late arrival, re-scheduling is performed using exceptional regulation, thus trying to reduce delay.

In the following steps, the Pickup's fixed successors are scheduled: In the basic P $\rightarrow$ D case, this is just the trip to the associated Delivery location. In request bundles, there may be a whole number of fixed successors (Pickups and Deliveries). Scheduling is again executed according to *basic rules*. Then it is checked whether there is a "late arrival". If this is not true, scheduling is accepted and the algorithm turns towards the following successors (if existent). However, if there is a late arrival, re-scheduling is started, using *all possible exceptions* to the basic regulation rules.

In contrast to the Pickup case, prevention of too early arrival is not attempted by switching the waiting time to the predecessor. There would be no advantage: once processing of an order has been started, the execution of the first Pickup and all remaining parts of the respective order are fixed and cannot be exchanged by other tasks. Hence, it is irrelevant where to schedule the waiting time.

Finally, the Scheduling module returns the achieved results to Best Insertion.

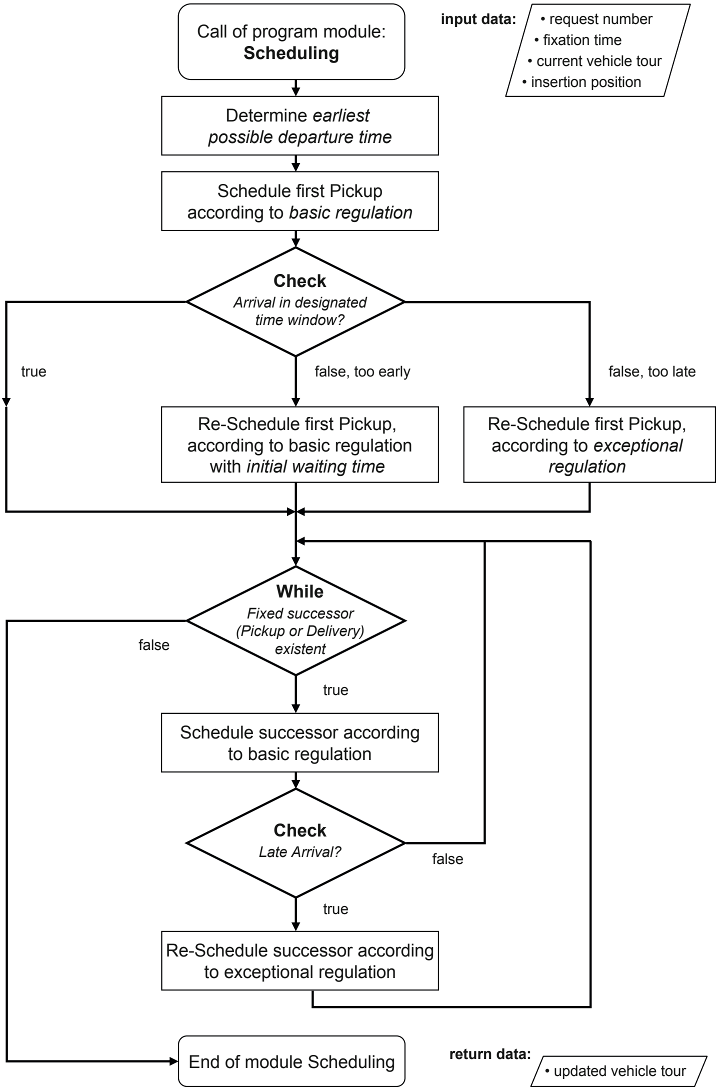


Figure 5.8: Program flow chart: Vehicle Scheduling

5.4 Real-Life Test Data Set: Preprocessing and Analysis

This section describes the preprocessing of a real-life test data set. First, the initially available raw data set is relieved from inappropriate data. Then, the remaining data are adjusted to a consistent notation. This is followed by some plausibility checks. Finally, benchmark values for the objective function criteria are derived.

The real-life raw data set contains order data of approximately five weeks (Monday, 17.08. 2009 until Saturday, 19.09.2009). It includes a total of 23,305 orders and 1,600 vehicles. In addition to the vehicle information, which is included in a separate table, the following order-related information is given:

- basic customer requests (type 1),
- request bundles (type 2),
- order(or bundle)-to-vehicle assignment (type 3), and
- the associated empty trips (type 4).

Type 1 and type 2 information can be interpreted as input data to the planning problem. Type 3 and type 4 information include the actual planning that was performed at the freight forwarding company.

Selection of appropriate data

Due to the fact that not only international transportation tasks and not only occasional transportation tasks without predefined networks are included, the data set has to be revised. For this purpose, each vehicle's actual tour (= planning result) is replicated with the type 3 and type 4 tables. Based on the specific tours, all vehicles only doing national transportation tasks are skipped. The same is applied to vehicles only performing regular line transportation. In addition, vehicles performing requests with origin or destination outside Europe (e.g. Afghanistan, Iran), as well as vehicles with nearly no activity are deleted. Vehicles changing from one driver operation to team driver operation and vice versa also have to be omitted. After this revision, there is a number of 953 remaining vehicles (related to the initial number of vehicles: approx. 60%).

In a next step, the requests that were actually transported with those vehicles remaining are selected. This results in a number of 14,025 requests (related to the initial number of orders: approx. 60%, as well): 900 static, and 13,125 dynamic (degree of dynamism = 93.6%). All the other requests – not transported with one of the remaining vehicles – are eliminated. This results in a total workload of 2,805 requests/week, and an average workload of 2.94 requests/week per available vehicle.

The remaining order and vehicle data, however, are not yet in a form that can be immediately handed to the adapted real-life MNS. Further preprocessing is required concerning notation inconsistencies and plausibility checks.

Adjustment of notation inconsistencies

Because of the use of inconsistent notation for *load_type* in the request and the vehicle database, the according entries have to be harmonized to give clear order-to-vehicle assignment rules to the algorithm. This step results in five *load_type* categories for vehicles (272∗M, 451∗P, 171∗K, 8∗V, and 51∗HZ) and six *load_type* categories for orders (1450∗M, 9499∗P, 1840∗K, 294∗KP, 112∗V, and 830∗HZ). An illustration is given in Figure 5.9.

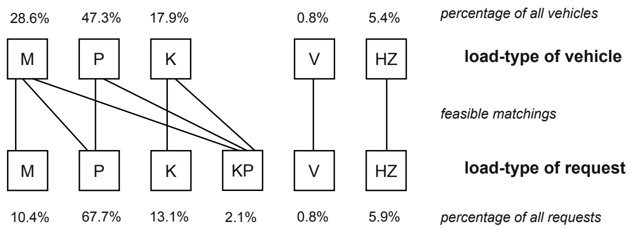


Figure 5.9: Feasible load-to-vehicle assignments

A matching is only possible, if the *load_type* of an order and the *load_type* of a vehicle are identical. This is symbolized by the direct connections between orders and vehicles with the same *load_type*. In the special case of request *load_type* “KP”, the vehicle types “K”, “P” and “M” are allowed for assignment. As a further exception, request *load_type* “P” may also be transported by vehicles with *load_type* “M”.

To give a little more insight into the meaning of these *load_type* shortcuts, the English and also the German truck descriptions shall be listed: “M” (Megatrailer - mega truck), “P” (LKW mit Plane - curtain side truck), “K” (Koffer/Kühler - box truck/refridgerated truck), “HZ” (Hängerzug - road train), and “V” (Vario truck). In addition, some exemplary pictures of the associated trucks and some technical details are shown in Figure 5.10.

Finally, it is also worth mentioning that 461 orders (3.2%) are classified as *hazardous goods*, therefore requiring a vehicle with special equipment for transportation of such goods. Two hundred and thirty-four vehicles (24.5%) carry such *hazardous goods* equipment.

Plausibility checks

After selection of the remaining basic customer requests (type 1), the following plausibility checks and consequential modifications are executed:

(i) IF (EPT < Call-In) THEN {Call-In = EPT - 1 day}

(ii) IF (EPT=LPT=00:00) THEN {EPT=03:00 and LPT=14:00}

(iii) IF (EPT + 1 hour > LPT) THEN {LPT = EPT + 1 hour}

(iv) IF (EPT > EDT) THEN {EDT=EPT}

mega truck (M)

- dimensions: 13.62 m (length)
  2.48 m (breadth)
  3.00 m (height)
- workload: 25 400 kg
- number of pallets: 34+34

curtain side truck (P)

- dimensions: 13.62 m (length)
  2.48 m (breadth)
  2.78 m (height)
- workload: 26 000 kg
- number of pallets: 34

box truck, refridgerated truck (K)

- dimensions: 13.41 m (length)
  2.46 m (breadth)
  2.65 m (height)
- workload: 24 310 kg
- number of pallets: 33

road train (HZ)

- dimensions (per swap-body):
  7.68 m (length)
  2.48 m (breadth)
  3.00 m (height)
- workload: 13 250 kg * 2
- number of pallets: 19+19

vario truck (V)

- dimensions: 5.00 m (length)
  2.45 m (breadth)
  2.40 m (height)
- workload: 2 700 kg

Figure 5.10: Different available vehicle types (Willi Betz Logistik, 2010)

(v) IF (EDT=LDT=00:00) THEN {EDT=07:00 and LDT=18:00}

(vi) IF (EDT + 1 hour > LDT) THEN {LDT = EDT + 1 hour}

The first check (i), ensures that an order's Call-In occurs before the associated Pickup time window opens. Checks (ii) and (v) catch a missing time window input. In such a case, the standard time windows [03:00, 14:00] and [07:00, 18:00] are chosen for Pickup and Delivery, respectively. Checks (iii) and (vi) make sure that all time windows possess a sufficiently long opening time. Check (iv) ensures a correct relative position of Pickup and Delivery time windows.

Furthermore, all times in an interval ranging from [23:00, 03:00] are set to 03:00 of the following day to comply with real-life restrictions. In cases with EDT - LPT > ten days, the time gap is set to exactly 10 days to avoid the need for storage.

Determination of initial vehicle position and availability

Finally, the vehicle data set is completed with the information of *when* and *where* each vehicle is initially available for new transportation tasks. The vehicle attribute "av_from" is chosen as the beginning time of the first vehicle task that was actually performed in the five-week horizon real-life planning (derived from type 3 and type 4 information). Initial geographical coordinates "geo_long_initial" and "geo_lat_initial" are chosen respectively.

In this way, the vehicles available for MNS planning start at the same time and at the same geographical position as the vehicles do in the actual real-life planning. Since MNS also gets the same orders as in the real-life planning, it could theoretically end up producing the same scheduling that human dispatchers have produced in real-life. On the other hand, it also gets the chance to produce a completely different, maybe better, planning.

Derivation of benchmark objective function values

After finishing the preprocessing of the real-life data, some performance indicators of actually performed planning at the freight forwarding company can be derived.

First, *total delay* is determined for all Pickup and Delivery requests. This is performed by calculating the gap between actual arrival time at a Pickup (Delivery) location and LPT (LDT). All delays are added, resulting in a total delay of 2,260.52 days at Pickup locations and of 4,391.44 days at Delivery locations. Hence, a total delay of 6,651.96 days was generated.

Subsequently, the *empty traveled distance* has to be calculated. For the real-life orders and the actual planning, however, there are only geographical coordinates of Pickup and Delivery locations available, but no information on actually driven street distances. Hence, some assumptions are required. We choose to calculate the actual traveled distances with a Euclidean metric. The same metric, of course, is used for the calculation of distances in the MNS procedure in order to ensure comparability.

Basic distance between two geographical coordinates A (x_A, y_A) and B (x_B, y_B) (with x_A, x_B representing geographical longitude of location A and B, respectively; and y_A, y_B representing geographical latitude of location A and B, respectively) is calculated as follows:

$$dist(A, B) = 111.2 \cdot \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 \cdot \cos(x_A) \cdot \cos(x_B)} \cdot 1.3$$

The formula contains an approximation of the air-line distance accounting for the curvature of the earth (cp. Fleischmann, 2010). The resulting distance is multiplied by a street factor of 1.3 in order to include deviations from the idealized air-line distance.

The distance formula is applied to all loaded and all unloaded trips which can be derived from type 3 and type 4 information. In this way, a total traveled distance of 16,535,592 kilometers is calculated. This number consists of 1,858,752 empty kilometers and 14,676,840 loaded kilometers. The associated *empty-to-all ratio* was:

$$\frac{\textit{empty driven kilometers}}{\textit{total number of driven kilometers (empty and loaded)}} = 11.2\%$$

The benchmark values which are used for comparison in Section 5.5 are summarized in Table 5.6. All values are given in "hours". The translation of traveled distance into traveled time is performed with the assumption of a vehicle average speed of 72 km/h (in agreement with our cooperating freight forwarding company). This vehicle speed assumption is also applied to all travel time calculations in the modified MNS procedure.

As explained above, values of *delay* are directly derived from the freight forwarding company's data set. This is also true for *empty and loaded travel time* and *total operating time*. Further assumptions, however, have to be made for the category *loading time*, which includes the times of all loading and unloading processes. Each loading and each unloading process is assumed to last one hour. In the specific case of several loading and/or unloading processes at the same geographical position, however, only the last loading process is assumed to have a duration of one hour. All previous loading processes are scheduled with only a single minute of loading time.

delay	159,647 hours
travel time	229,661 hours
loaded	203,845 hours
unloaded	25,816 hours
loading time	24,000 hours
break/wait	496,471 hours
total operating time	750,132 hours

Table 5.6: Benchmark results for the five-week real-life test data set

In a last step, the *break/wait value* has to be calculated. This is performed indirectly, by subtracting traveling time and loading time from total operating time.

At this point, the preprocessing and the analysis of the real-life data set is complete. The derived benchmark values for the objective function criteria are used in the following section to evaluate the performance of the adapted MNS procedure.

5.5 Computational Results

This section starts with the parameterization of the adapted MNS procedure with regard to *penalty cost*, *anticipation horizon* and application of *improvement neighborhoods*. In addition, some insights into the impact of simulation speed on solution quality are derived (Section 5.5.1). Afterwards, an exemplary MNS generated tour for the five-week real-life test data set is presented. This includes a detailed analysis of the generated scheduling and an investigation of its compliance with the chosen general real-life requirements (Section 5.5.2).

In a next step, some of the computational results are reported, pointing out the solution's dependency on the weighting of the objective function criteria *delay* and *empty travel time* (Section 5.5.3). Finally, the pros and cons of the implementation of a dynamic Fleet Management System for International Truck Transportation are discussed (Section 5.5.4).

5.5.1 Parameterization

Parameterization and all subsequent simulation runs are executed on a quad core PC (Intel Core 2 Quad CPU, 2.83 GHz, 8 GB RAM), with each simulation running on one of the four available cores. All detailed parameterization results can be found in Appendix B. In the beginning, there is the open question of how to choose the **penalty costs for delay, empty travel time and waiting**. From our cooperating freight forwarding company, there is only the general specification of weighting the reduction of empty travel time highest and the reduction of delay second highest. The parameterization is started with a high simulation speed of s = 120, which is successively decreased to a real time simulation (s = 1).

To get a first impression of the impact of the different penalty cost values and of mutual dependency, several penalty cost combinations are investigated with a **high simulation speed of s = 120**. For the five-week real-life data set, this results in a simulation time of approximately seven hours for each cost combination. In total, 49 different penalty cost variations are tested: due to the relatively lower importance of *waiting*, the associated penalty costs are fixed at a value of 1 for all combinations. Penalty costs for *delay* and *empty travel time* are both chosen from the penalty cost set {1, 5, 8, 10, 20, 30, 40}. In the following, the chosen penalty cost pairs are denoted as "a,b", with *a* representing the penalty cost for *empty travel time*, and *b* representing the penalty cost for *delay*. As further parameter settings, the initial improvement time is set to 180 minutes (neighborhoods I:II:III = 1:1:1), for general improvement the neighborhoods I and II are chosen in relation 66:33, the tabu time of neighborhood I is set to 30 minutes, and the anticipation horizon is chosen to be 10 minutes.

The results are summarized in Table 5.7. For each penalty cost combination, the four solution criteria *empty travel time*, *delay*, *break/wait*, and *total operating time* are given in the form of the percentage deviation from the benchmark values derived in Section 5.4. Here, a negative value indicates an improvement of solution quality, while a worsening of solution quality is indicated by a positive value. For the penalty cost combination (30,5), for example, an empty travel time of 26,025 hours (+0.8%), a delay of 217,675 hours (+36.3%), a break/wait time of 489,456 hours (-1.4%), and a total operating time of 743,325 hours (-0.9%) are achieved. In total, the performance of the high simulation speed results is very modest. All results are clearly inferior to the benchmark.

However, some general observations can be made. A higher weighting of the cost for *empty traveling* induces a monotonous improvement in the associated empty travel time. This monotonous behavior can also be found in most of the cases of delay, when the associated penalty costs (for *delay*) are increased. The results indicate that there is antithetic behavior between minimization of empty travel time and minimization of delay. Improvements in one category are accompanied by a worsening in the other category. Therefore, a parameterization has to be found that achieves preferably good planning results for both requirements.

Fortunately, values of *break/wait* and *total operating time* seem to be less sensitive to the penalty cost variations for empty travel time and delay. It can be observed that the best break/wait result is achieved for a cost parameterization (1,1). This is an intuitive finding, since the penalty cost value of break/wait that has been initially fixed to 1, is relatively the highest for this cost combination. In addition, the results of *break/wait* and *total operating time* are also slightly better when "better" results for empty travel time are achieved.

The parameterization of delay and empty travel time penalty costs is continued with an analysis of a slower **simulation speed of s = 5**. Since a slower simulation speed induces longer simulation runs (for s = 5: approximately one week), we choose only a subset of

penalty cost "traveling empty"		penalty cost "delay"													
		1		**5**		**8**		**10**		**20**		**30**		**40**	
1		56.9	26.3	69.5	-12.2	74.5	-9.8	74.3	-5.3	78.3	-2.2	78.9	-3.7	78.5	-7.7
		-7.4	-2.9	-3.5	0.0	-2.8	0.7	-2.5	0.9	-1.9	1.5	-1.8	1.5	-2.1	1.3
5		11.0	47.6	34.8	-12.7	39.2	-13.0	39.7	-14.2	46.3	-5.3	48.0	-9.4	50.4	-3.7
		-6.9	-4.2	-2.5	-0.5	-1.9	0.1	-1.6	0.3	-1.3	0.7	-1.0	1.0	-1.4	0.8
8		0.3	68.0	25.2	-12.0	30.4	-12.2	34.0	-12.2	40.6	-10.1	41.9	-10.8	43.3	-9.3
		-6.7	-4.4	-2.1	-0.5	-1.2	0.2	-1.1	0.4	-1.1	0.7	-0.8	0.9	-0.6	1.1
10		-4.7	84.0	21.1	-9.5	26.5	-11.9	30.7	-15.3	36.3	-13.1	39.3	-8.5	41.9	-7.7
		-6.3	-4.3	-1.9	-0.5	-1.6	-0.1	-1.6	0.0	-0.9	0.6	-0.9	0.8	-0.6	1.0
20		-17.1	164.3	9.6	10.2	16.4	3.7	19.3	-3.9	29.6	-8.3	34.7	-4.4	36.6	-6.4
		-4.8	-3.8	-1.4	-0.6	-1.6	-0.5	-0.8	0.1	-0.7	0.5	-0.3	1.0	-0.2	1.1
30		-22.2	235.9	**0.8**	**36.3**	9.4	13.3	13.6	9.9	23.5	-3.4	29.7	-1.4	34.0	-2.5
		-3.9	-3.3	**-1.4**	**-0.9**	-0.8	-0.2	-0.7	0.0	-0.5	0.5	-0.3	0.8	-0.9	0.6
40		-26.2	316.2	-4.5	55.0	3.5	32.0	8.4	19.3	20.2	3.7	25.9	-1.5	30.7	-1.7
		-4.1	-3.6	-1.2	-1.0	-0.8	-0.4	-0.7	-0.2	-0.4	0.4	-0.6	0.5	-0.4	0.8

percentage deviation from benchmark

empty travel time	delay
break/wait	operating time

penalty cost "waiting" = 1

Table 5.7: Parameterization of penalty costs (sim speed s: = 120, improvement neighborhoods: I-II 66%-33%, anticipation horizon: 10 min)

nine promising parameter combinations from the initial analysis for further investigation: 20,8; 30,10; 30,8; 20,5; 40,8; 30,5; 40,5; 8,1; 10,1. All other settings are kept the same as in the first case.

The results are visualized in Figure 5.11 (black triangles). On the x-axis and the y-axis, the actually achieved objective function values for delay and empty travel time are outlined. For the penalty cost combination (30,10), for example, an empty travel time of 27,424 hours, a delay of 128,188 hours, a break/wait time of 489,660 hours, and a total operating time of 744,927 hours is generated. The reproduction of detailed results for break/wait and total operating time is skipped in this illustration to allow for a clear representation of empty travel time and delay.

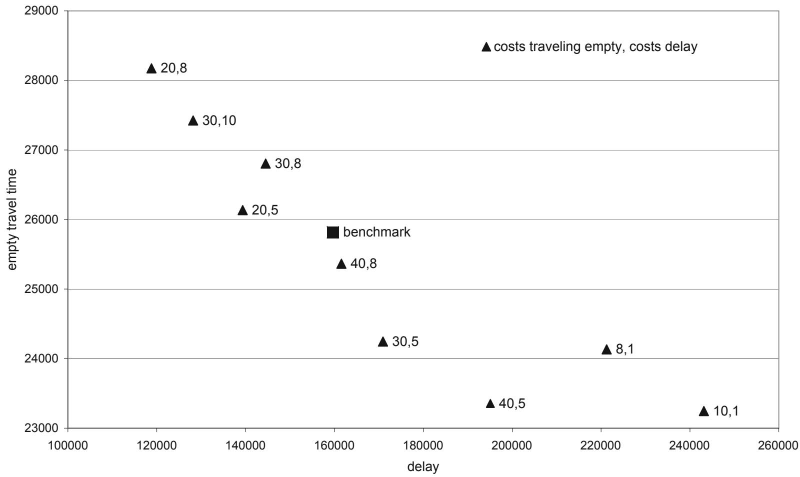


Figure 5.11: Parameterization of penalty costs (sim speed: s = 5, improvement neighborhoods: I:II 66:33, anticipation horizon: 10 min)

Furthermore, the relative location of the benchmark result is included (black square), which allows for a division of the figure into four quadrants relative to the benchmark result. In the upper-left quadrant, delay can be reduced in comparison to the benchmark, but there is an increase in empty travel time. In the lower-right quadrant, empty travel time can be reduced in comparison to the benchmark, but there is a increase in delay. In the upper-right quadrant, there are no relative improvements at all. In the lower-left quadrant, both categories, delay and empty travel time, can be improved. All achieved results fall into the upper-left quadrant (penalty cost combinations: 20,8; 30,10; 30,8; 20,5) and into the lower-right quadrant (penalty cost combinations: 40,8; 30,5; 40,5; 8,1; 10,1). Hence, only one objective function criterion can be improved in each case, but not both at the same time.

Since the reduction in simulation speed from s = 120 to s = 5 has caused significant improvements in solution quality, we (again) select a subset of promising cost parameter settings from the current s = 5 results (30,10; 20,5; 40,8; 30,5; 40,5) for a final analysis with the **simulation speed of s = 1 (real time simulation)**. Consequently, the simulation time for a single real time simulation run is now approximately five weeks.

The s = 1 results are visualized in Figure 5.12. The gray triangles show the results that are generated with a simulation speed of s = 1. In this case, two penalty cost combinations – 40,8 and 30,5 – reach the preferred lower-left quadrant. With the penalty cost values (40,8), an empty travel time of 24,817 hours and a delay of 150,976 hours is achieved. In comparison to the benchmark, this is a significant reduction of 3.9% (empty travel time) and of 5.4% (delay). With the penalty cost values (30,5) empty travel time is reduced even further (-6.4%), but with the drawback of an only slightly reduced delay (-0.1%). Nevertheless, the penalty cost combination (30,5) is chosen for the remaining parameterizations.

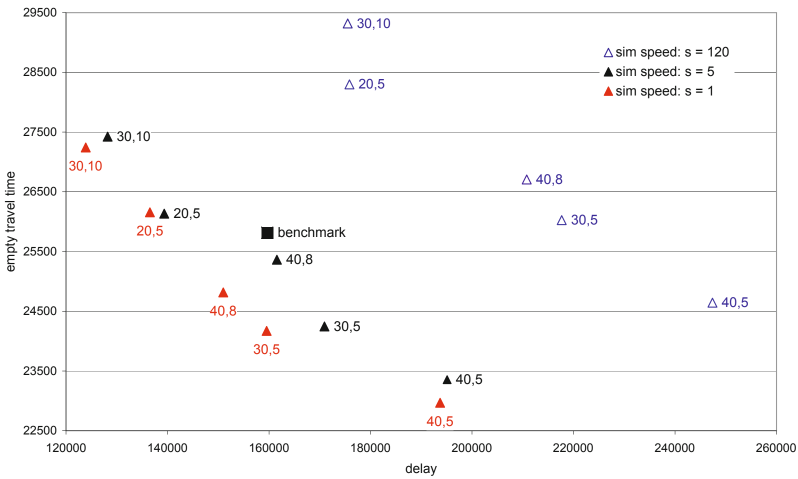


Figure 5.12: Parameterization of penalty costs (sim speed: s = 1, improvement neighborhoods: I:II 66:33, anticipation horizon: 10 min), impact of solution speed

Figure 5.12 also allows for an analysis of the *impact of simulation speed on the solution quality*. For the five penalty cost combinations that are simulated in real time, the results that are achieved with simulation speeds s = 5 (black triangles) and s = 120 (blank triangles) are also given. In this way, the improvement that is achieved by slower simulation times can be tracked:

In the exemplary case of the penalty cost values (40,8), with a simulation speed of s = 120, an empty travel time of 26,707 hours and a delay of 210,785 hours is produced. This is reduced to 25,364 hours of empty travel time (-5.1%) and to 161,562 hours of delay (-23.4%) with simulation speed s = 5. Further improvements are achieved with the real time simulation. Empty travel time is reduced once more by 2.2%, and delay is reduced by 6.6%.

These results definitely indicate an *advantage of the real time simulation*. The extra time that is given to applying the improvement neighborhoods of MNS seems to be beneficial. A drawback, however, is the long calculation time that only allows for a limited number of parameterizations to be tested.

In a second step, the **anticipation horizon** is parameterized, investigating the following values: 5 min, 10 min, 30 min, 60 min, 90 min, and 120 min. As further settings, the penalty costs values (30,5) and the improvement neighborhoods I:II in relation 66:33 are chosen. All other settings are the same as in the previous tests. The simulations are executed twice: with a simulation speed of s = 5 and with a simulation speed of s = 1. Figures 5.13 and 5.14 show the achieved results for the respective simulation speeds. The results are grouped according to the objective function criteria empty travel time, delay, break/wait, and total operating time. For each variation, the results that are achieved with a specific anticipation horizon are given as percentage deviations from the benchmark.

For a simulation speed of s = 5, best performance in empty travel time is achieved for an anticipation horizon of 10 minutes. Interestingly, the 30-minute horizon achieves the worst result of all investigated anticipation horizons. In category delay, the anticipation horizon of 30 minutes, however, shows the best performance and is slightly better than the 10-minute horizon. For longer anticipation horizons (e.g., 90 min or 120 min), solution quality in delay drops significantly. Break/Wait and total operating time show quite insensitive behavior to variations of the anticipation horizon. In total, the 10-minute anticipation horizon shows the best performance for a simulation speed of s = 5.

For a simulation speed of s = 1, the situation changes. In terms of empty travel time reduction, the 10-minute horizon is outperformed by all longer anticipation horizons. For this category, best results are achieved with a horizon of 60 minutes, followed by a horizon of 30 minutes. In category delay, the best result is achieved with a horizon of 30 minutes. Like in the s = 5 simulation, break/wait and total operating time show quite insensitive behavior to variations of the anticipation horizon. In total, the 30-minute anticipation horizon now demonstrates the best performance (for s = 1).

It is interesting that parameterizations with different simulation speeds cause different results. This observation indicates that a proper parameterization for a real time simulation should be also executed in real time, at least if there is enough time. For the calculation of the final results, which is of course performed in real time, the anticipation horizon is therefore chosen to be 30 minutes.

The last parameterization concerns the **allocation of improvement time** to available neighborhood operations. For detection of the best combination, the following variations are analyzed: 100:0, 75:25, 66:33, 50:50, 33:66, 25:75, and 0:100. These numbers can be interpreted as percentage values and serve as a basis of how much computation time is allocated to neighborhoods I and II. As further settings, the penalty cost values (30,5) and an anticipation horizon of 10 minutes are chosen. All other settings are kept the same as in the previous tests. As in the previous case, the simulation of the investigated variations are executed twice: with a simulation speed of s = 5 and with a simulation speed of s = 1.

Figures 5.15 and 5.16 show the achieved results for both simulations, respectively. The results are grouped again according to the objective function criteria empty travel time,

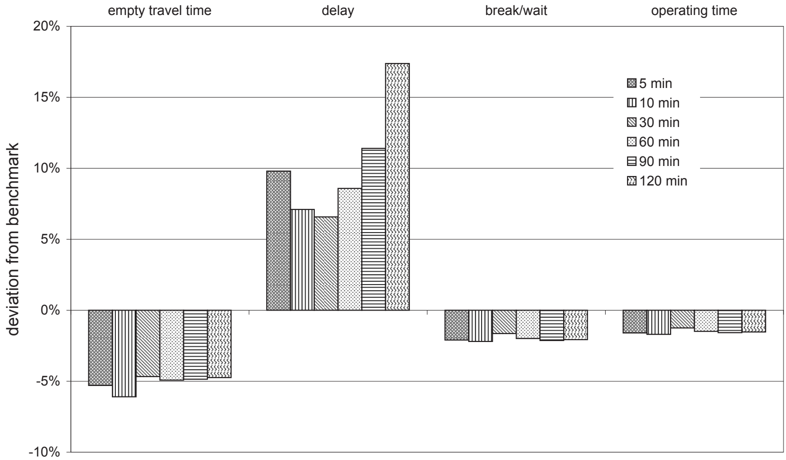


Figure 5.13: Parameterization of anticipation horizon, 5 min up to 120 min (penalty costs: 30,5, sim speed: s = 5, improvement neighborhoods: I:II 66:33)

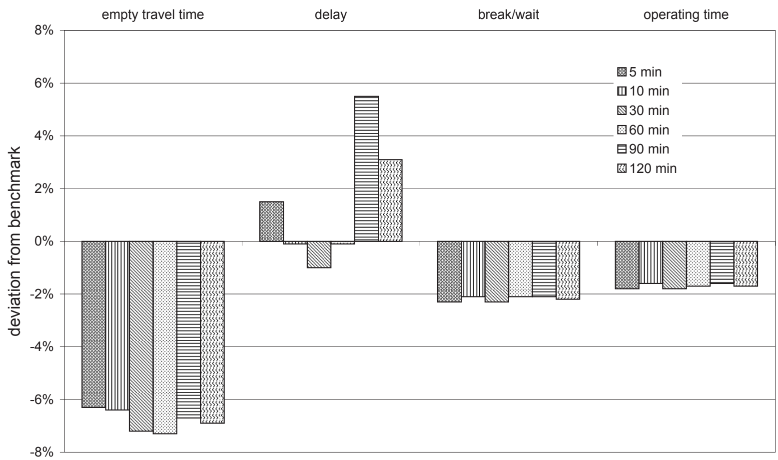


Figure 5.14: Parameterization of anticipation horizon, 5 min up to 120 min (penalty costs: 30,5, sim speed: s = 1, improvement neighborhoods: I:II 66:33)

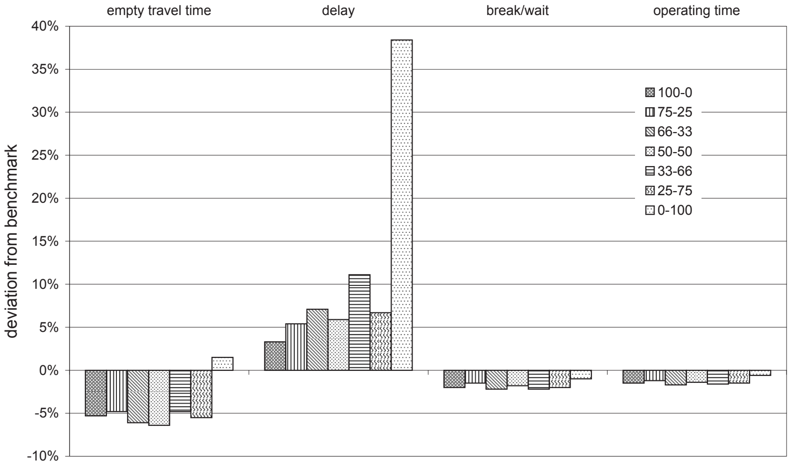


Figure 5.15: Parameterization "allocation of improvement time", neighborhood I : neighborhood II (penalty costs: 30,5, sim speed: s = 5, anticipation horizon: 10 min)

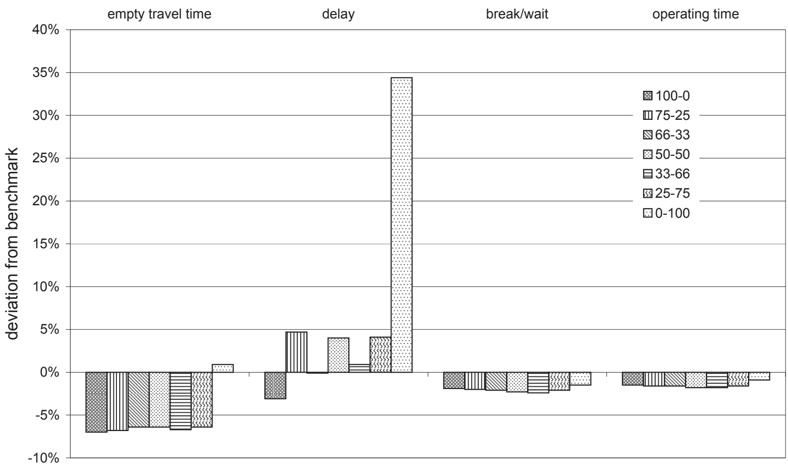


Figure 5.16: Parameterization "allocation of improvement time", neighborhood I : neighborhood II (penalty costs: 30,5, sim speed: s = 1, anticipation horizon: 10 min)

delay, break/wait, and total operating time. For each variation, the results that are achieved with a specific allocation of improvement time are given as percentage deviations from the benchmark.

For a simulation speed of s = 5, the best performance in empty travel time is achieved by the allocation 50:50, which is followed by the allocation 66:33 with the second best result. In terms of delay, the best result is generated by the allocation 100:0. The opposite allocation 0:100 results in the worst level of delay. 50:50, here as well, is slightly better than 66:33. Again, break/wait and total operating time show a quite insensitive reaction to the applied variations. In conclusion, best overall performance is achieved with the allocation 50:50 (for s = 5).

For a simulation speed of s = 1, however, differing results are generated. The best performance in terms of empty travel time, and also delay, is achieved with the allocation 100:0. While the other allocations produce empty travel times at least in the same range as 100:0, the picture changes in delay: here, a 3.1% reduction is generated by 100:0, with the second best allocation 66:33 being stuck only at a 0.1% reduction.

The surprising success of allocation 100:0 can be explained as follows. The real time simulation allocates a five times higher amount of calculation time to improvement procedure II (intraroute exchanges). In some cases, all exchange operations of improvement procedure II are investigated before the allocated time is actually consumed. If such a situation occurs, the excessive time is not used for any other calculation in order to keep the percental allocation of improvement time at the specified levels. In contrast, improvement procedure I has so many exchange operations available that it never "runs out of work". With the allocation 100:0, therefore, the available improvement time can be used completely, which declares its better performance.

At this point, the main parameterizations of penalty costs, anticipation horizon and improvement procedure are finished. Table 5.8 summarizes all internal parameters and penalty cost settings that are finally selected for the application of the real-life MNS on the real-life data set.

Real-Life MNS	
internal parameters	
initial improvement	
duration:	180 min
neighborhoods I:II:III	1:1:1
general improvement:	
neighborhoods I:II	100:0
tabu time:	30 min
anticipation:	30 min
penalty costs	
c_traveling_empty (per min):	30, 40, 40
c_delay (per min):	5, 5, 8
c_wait (per min):	1, 1, 1

Table 5.8: Parameter settings for real-life test data set

The results that are achieved with this "best" parameterization will be presented in Section 5.5.3. However, before we come to the final results, the following section analyzes a typical vehicle tour that is produced with MNS.

5.5.2 Exemplary Real-Life Scheduling

In this subsection an exemplary real-life vehicle tour that is produced by MNS for the five-week real-life data set is selected for detailed investigation. For this purpose, simply the very first available vehicle (in single driver mode) with internal number "00001" is chosen. The tour planning is generated with the parameter settings: penalty costs (40,8), an anticipation horizon of 10 minutes and with the application of improvement neighborhoods I:II in proportion 66:33. All other settings are kept the same as in the previous tests. The simulation is executed in real time (simulation speed s = 1).

Figure 5.17 visualizes the **course of the resulting five-week vehicle tour on a European map**. The vehicle starts its trip in Reutlingen (Germany) and finishes its tour in Dieppe (France). The sequence of locations that are included in the tour is indicated in capital letters from A (Reutlingen, Germany) to S (Dieppe, France). Furthermore, information on Pickup and Delivery locations of every *loaded trip* is given. In the exemplary tour, there are also some request bundles having more than one Pickup and/or Delivery location. Due to the relative geographical proximity of these locations, they are treated as a single location (only one capital letter).

The *first loaded trip* directs the vehicle from Reutlingen (Germany) at point A, to Miskolc (Hungary) at point B. Afterwards, the vehicle has to perform an empty trip from Miskolc (Hungary) at point B to Mosonszolnok (Hungary) at point C. Here, the vehicle gets its *second loaded trip* from Mosonszolnok (Hungary) at point C, to Vienna (Austria) at point D. In Vienna (Austria) at point D, the *third loaded trip* is directly available: Vienna (Austria) at point D, to Mannheim (Germany) at point E. From Mannheim (Germany) at point E, the vehicle has to perform a short empty traveling distance to Heidelberg (Germany) at point F. From Heidelberg (Germany) at point F the *fourth loaded trip* is started, which has its destination in Busalla (Italy) at point G. And so on... The resulting vehicle tour has an empty-to-all ratio of 9.38% and an average travel time per week of 42.6 hours.

Figures 5.18, 5.19, 5.20, and 5.21 show the associated **detailed scheduling** (Excel output file of MNS). The **meaning of the available columns** is explained as follows. In the first column, an *activity log* is included that describes the vehicles respective activity (e.g., "trip to Pickup of order 1307", "break of journey"). In the second column, the *time interval for the respective activity* is given (e.g., "Mo 24.8.2009 8:30 – Mo 24.8.2009 13:00"). Columns three and four include the *geographical coordinates (longitude and latitude)* of Pickup and Delivery locations (in the format "degree,mmss"). This is followed by column four, which is not originally included in the MNS output file. Column four is created, in order to allow for a cross-reference to the European map in Figure 5.17: The capital letters which are used to visualize the vehicle's main Pickup and Delivery locations in Figure 5.17 are included here in the detailed vehicle scheduling for a better traceability of the vehicle's tour.

Column six contains the Pickup or Delivery *time window* when a respective location is reached (e.g., "Tu 18.8.2009 16:00 – Tu 18.8.2009 22:00"). Afterwards, the actually sched-

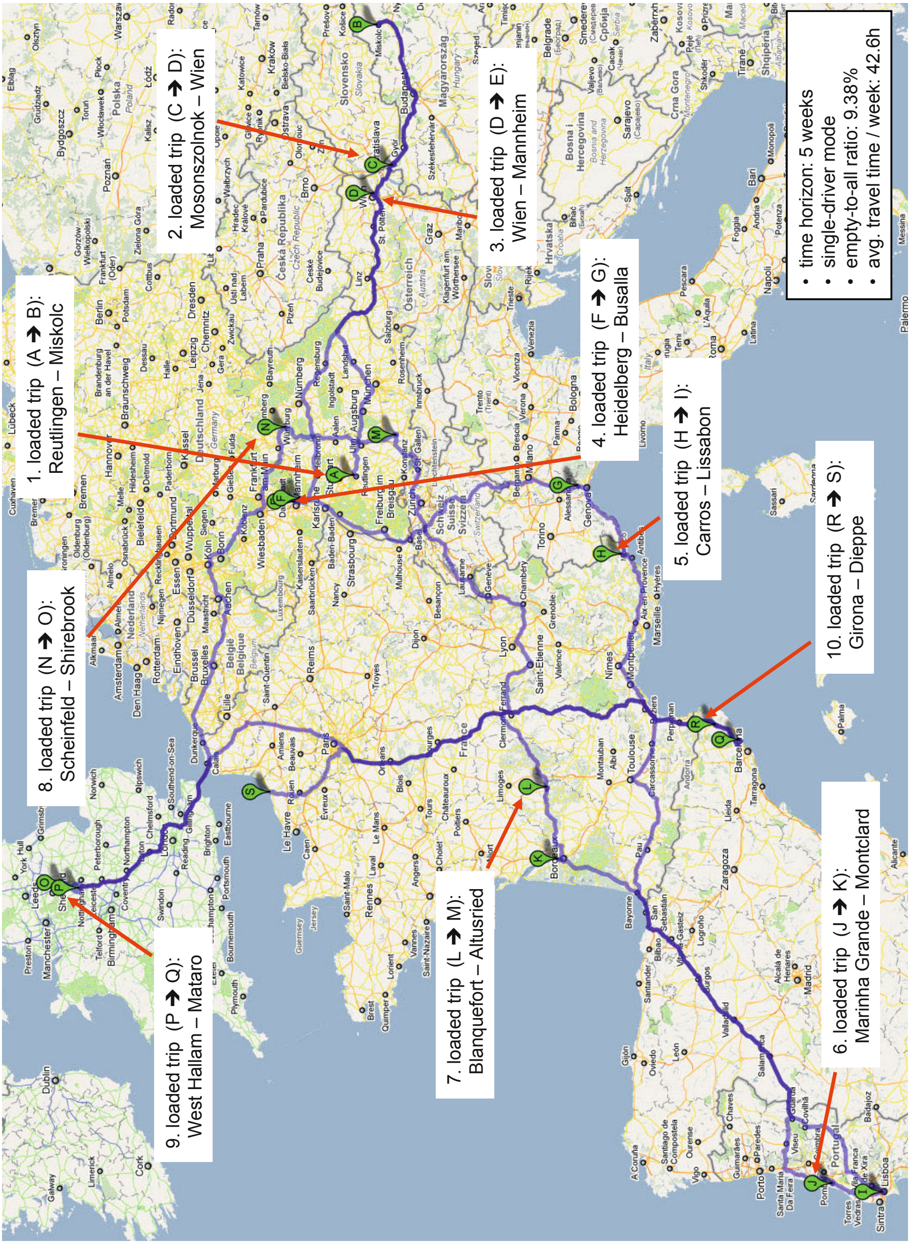


Figure 5.17: Exemplary results: five-week vehicle tour - geographical illustration

uled *arrival* time is given in column seven (e.g., "Tu 18.8.2009 16:00"). Columns eight to eleven include summary values for *break* time, *wait* time, *travel* time, and *load* time for the current activity (in the format "hh:mm"). These values support the inspection of whether all real-life restrictions have actually been complied with. If an arrival at Pickup or Delivery occurs after LPT or LDT, the associated *delay* is given in the last column (in the format "hh:mm").

In the following, a **detailed description of how to read the scheduling entries** is given.

In the beginning *(line 1)*, the vehicle's initial position ("9,1329, 48,3031") is given. Afterwards *(line 2)*, it is indicated that this specific vehicle is not available from the very beginning of the simulation, but first at "Tu 18.8.2009 16:00". In the scheduling of *line 3*, the vehicle is sent to the Pickup of order 1307. Since the associated Pickup location is in very close proximity to the vehicles initial position, no travel time is scheduled. The loading activity is started immediately, and takes one minute *(line 4)*. This is due to the fact that this Pickup is bundled with a second order at the same geographical position. *Line 5* contains the scheduling of the second order "trip to Pickup of order 1338". This Pickup location is also reached immediately. Since it is the last loading activity at the same geographical location, it results in the scheduling of the regular one-hour loading time *(line 6)*.

At "Tu 18.8.2009 17:01", the loading activities are finished and the vehicle starts its first real trip towards Delivery of order 1307 *(line 7)*. The first travel time interval lasts 4:30h from "Tu 18.8.2009 17:01 – Tu 18.8.2009 21:31". Then, a 45-minute break is scheduled from "Tu 18.8.2009 21:31 – Tu 18.8.2009 22:16" *(line 8)*. This is followed by the next 4:30h travel time interval from "Tu 18.8.2009 22:16 – We 19.8.2009 02:46" *(line 9)*. After a further 45-minute break *(line 10)*, an exceptional extra driving hour is scheduled from "We 18.8.2009 03:31 – We 19.8.2009 04:31" *(line 11)*. At this point, the maximum daily interval driving time of 10 hours is reached. Hence, a daily rest period is scheduled: in the present case, a reduced daily rest period of 9 hours *(line 12)*.

At "We 19.8.2009 13:31", the journey is continued with the next 4:30h travel time interval from "We 19.8.2009 13:31 – We 19.8.2009 18:01" *(line 13)*. This is followed, by a 45-minute break *(line 14)*, a further 4:30h travel time interval *(line 15)*, one more 45-minute break *(line 16)*, and a final driving hour from "Th 20.8.2009 00:01 – Th 20.8.2009 01:01" *(line 17)*. Due to this second extension to 10 hours of daily travel time, this week's potential 10-hour travel time extensions are completely utilized. Afterwards, the next reduced daily rest period is scheduled from "Th 20.8.2009 01:01 – Th 20.8.2009 10:01" *(line 18)*. This consumes the second of in total three reduced daily rest periods available per week.

At "Th 20.8.2009 10:01", the vehicle continues its trip for 3:55h and finally reaches the Delivery location of order 1307 at "Th 20.8.2009 13:56" *(line 19)*. This arrival time involves a delay of 15:56h. *Line 20* contains the associated one-hour unloading activity. In a next step, the vehicle travels 0:49h from "Th 20.8.2009 14:56 – Th 20.8.2009 15:45" to the request bundle's second Delivery of order 1338 *(line 21)*. This arrival time is also delayed, in this case by 17:45h. After the unloading activity is finished *(line 22)*, a weekly

	activity log	interval of activity	geo.long.	geo.lat.		time window	arrival	break	wait	travel	load	delay
1	initial position		9,1329	48,3031								
2	vehicle not yet available!	Mo 17.8.2009 00:00 -- Tu 18.8.2009 16:00										
3	trip to Pickup of order 1307	Tu 18.8.2009 16:00 -- Tu 18.8.2009 16:00	9,1324	48,3041	A	Tu 18.8.2009 16:00 -- Tu 18.8.2009 22:00	Tu 18.8.2009 16:00					
4	loading of order 1307	Tu 18.8.2009 16:00 -- Tu 18.8.2009 16:01									:01	
5	trip to Pickup of order 1338	Tu 18.8.2009 16:01 -- Tu 18.8.2009 16:01	9,1324	48,3041		Tu 18.8.2009 16:00 -- Tu 18.8.2009 22:00	Tu 18.8.2009 16:01					
6	loading of order 1338	Tu 18.8.2009 16:01 -- Tu 18.8.2009 17:01									1:00	
7	trip to Delivery of order 1307	Tu 18.8.2009 17:01 -- Tu 18.8.2009 21:31								4:30		
8	break of journey	Tu 18.8.2009 21:31 -- Tu 18.8.2009 22:16						:45				
9	continuation of journey	Tu 18.8.2009 22:16 -- We 19.8.2009 02:46								4:30		
10	break of journey	We 19.8.2009 02:46 -- We 19.8.2009 03:31						:45				
11	continuation of journey	We 19.8.2009 03:31 -- We 19.8.2009 04:31								1:00		
12	break of journey	We 19.8.2009 04:31 -- We 19.8.2009 13:31						9:00				
13	continuation of journey	We 19.8.2009 13:31 -- We 19.8.2009 18:01								4:30		
14	break of journey	We 19.8.2009 18:01 -- We 19.8.2009 18:46						:45				
15	continuation of journey	We 19.8.2009 18:46 -- We 19.8.2009 23:16								4:30		
16	break of journey	We 19.8.2009 23:16 -- Th 20.8.2009 00:01						:45				
17	continuation of journey	Th 20.8.2009 00:01 -- Th 20.8.2009 01:01								1:00		
18	break of journey	Th 20.8.2009 01:01 -- Th 20.8.2009 10:01						9:00				
19	trip to Delivery of order 1307	Th 20.8.2009 10:01 -- Th 20.8.2009 13:56	21,0749	47,5135		We 19.8.2009 06:00 -- We 19.8.2009 22:00	Th 20.8.2009 13:56			3:55		15:56
20	unloading of order 1307	Th 20.8.2009 13:56 -- Th 20.8.2009 14:56									1:00	
21	trip to Delivery of order 1338	Th 20.8.2009 14:56 -- Th 20.8.2009 15:45	20,4725	48,0656	B	We 19.8.2009 06:00 -- We 19.8.2009 22:00	Th 20.8.2009 15:45			:49		17:45
22	unloading of order 1338	Th 20.8.2009 15:45 -- Th 20.8.2009 16:45									1:00	
23	break of journey	Th 20.8.2009 16:45 -- Mo 24.8.2009 08:30						87:45				
24	trip to Pickup of order 3693	Mo 24.8.2009 08:30 -- Mo 24.8.2009 13:00								4:30		
25	break of journey	Mo 24.8.2009 13:00 -- Mo 24.8.2009 13:45						:45				
26	trip to Pickup of order 3693	Mo 24.8.2009 13:45 -- Mo 24.8.2009 16:30	17,1038	47,5104	C	Mo 24.8.2009 15:00 -- Mo 24.8.2009 17:00	Mo 24.8.2009 16:30			2:45		
27	loading of order 3693	Mo 24.8.2009 16:30 -- Mo 24.8.2009 17:30									1:00	
28	trip to Delivery of order 3693	Mo 24.8.2009 17:30 -- Mo 24.8.2009 19:04	16,2731	48,1056	D	Mo 24.8.2009 18:00 -- Mo 24.8.2009 19:00	Mo 24.8.2009 19:04			1:34		:04
29	unloading of order 3693	Mo 24.8.2009 19:04 -- Mo 24.8.2009 20:04									1:00	
30	trip to Pickup of order 3762	Mo 24.8.2009 20:04 -- Mo 24.8.2009 20:04										
31	waiting time	Mo 24.8.2009 20:04 -- Tu 25.8.2009 07:00							10:56			
32	trip to Pickup of order 3762	Tu 25.8.2009 07:00 -- Tu 25.8.2009 07:00	16,2731	48,1056		Mo 24.8.2009 18:00 -- Mo 24.8.2009 19:00	Tu 25.8.2009 07:00					12:00
33	loading of order 3762	Tu 25.8.2009 07:00 -- Tu 25.8.2009 07:01									:01	
34	trip to Pickup of order 3765	Tu 25.8.2009 07:01 -- Tu 25.8.2009 07:01	16,2731	48,1056		Mo 24.8.2009 18:00 -- Mo 24.8.2009 19:00	Tu 25.8.2009 07:01					12:01
35	loading of order 3765	Tu 25.8.2009 07:01 -- Tu 25.8.2009 07:02									:01	
36	trip to Pickup of order 3766	Tu 25.8.2009 07:02 -- Tu 25.8.2009 07:02	16,2731	48,1056		Mo 24.8.2009 18:00 -- Mo 24.8.2009 19:00	Tu 25.8.2009 07:02					12:02
37	loading of order 3766	Tu 25.8.2009 07:02 -- Tu 25.8.2009 08:02									1:00	
38	trip to Delivery of order 3762	Tu 25.8.2009 08:02 -- Tu 25.8.2009 12:32								4:30		
39	break of journey	Tu 25.8.2009 12:32 -- Tu 25.8.2009 13:17						:45				
40	continuation of journey	Tu 25.8.2009 13:17 -- Tu 25.8.2009 17:47								4:30		

Figure 5.18: Exemplary results: five-week vehicle tour - scheduling I

	activity log	interval of activity	geo.long.	geo.lat.		time window	arrival	break	wait	travel	load	delay
41	break of journey	Tu 25.8.2009 17:47 -- Tu 25.8.2009 18:32						:45				
42	continuation of journey	Tu 25.8.2009 18:32 -- Tu 25.8.2009 19:32								1:00		
43	break of journey	Tu 25.8.2009 19:32 -- We 26.8.2009 04:32						9:00				
44	continuation of journey	We 26.8.2009 04:32 -- We 26.8.2009 09:02								4:30		
45	break of journey	We 26.8.2009 09:02 -- We 26.8.2009 09:47						:45				
46	trip to Delivery of order 3762	We 26.8.2009 09:47 -- We 26.8.2009 11:45	8,1630	48,5220		Tu 25.8.2009 08:00 -- Tu 25.8.2009 16:00	We 26.8.2009 11:45			1:58		19:45
47	unloading of order 3762	We 26.8.2009 11:45 -- We 26.8.2009 11:46									:01	
48	trip to Delivery of order 3765	We 26.8.2009 11:46 -- We 26.8.2009 11:50	8,1459	48,5430		Tu 25.8.2009 08:00 -- Tu 25.8.2009 16:00	We 26.8.2009 11:50			:04		19:50
49	unloading of order 3765	We 26.8.2009 11:50 -- We 26.8.2009 12:50									1:00	
50	trip to Delivery of order 3766	We 26.8.2009 12:50 -- We 26.8.2009 14:10	8,3310	49,3039	E	Tu 25.8.2009 08:00 -- Tu 25.8.2009 16:00	We 26.8.2009 14:10			1:20		22:10
51	unloading of order 3766	We 26.8.2009 14:10 -- We 26.8.2009 15:10									1:00	
52	trip to Pickup of order 4241	We 26.8.2009 15:10 -- We 26.8.2009 15:25	8,3816	49,2438	F	We 26.8.2009 08:00 -- We 26.8.2009 09:00	We 26.8.2009 15:25			:15		6:25
53	loading of order 4241	We 26.8.2009 15:25 -- We 26.8.2009 16:25									1:00	
54	trip to Delivery of order 4241	We 26.8.2009 16:25 -- We 26.8.2009 17:18								:53		
55	break of journey	We 26.8.2009 17:18 -- Th 27.8.2009 04:18						11:00				
56	continuation of journey	Th 27.8.2009 04:18 -- Th 27.8.2009 08:48								4:30		
57	break of journey	Th 27.8.2009 08:48 -- Th 27.8.2009 09:33						:45				
58	continuation of journey	Th 27.8.2009 09:33 -- Th 27.8.2009 13:45								4:12		
59	waiting time	Th 27.8.2009 13:45 -- Fr 28.8.2009 08:00							18:15			
60	trip to Delivery of order 4241	Fr 28.8.2009 08:00 -- Fr 28.8.2009 08:00	8,5720	44,3425	G	Fr 28.8.2009 08:00 -- Fr 28.8.2009 09:00	Fr 28.8.2009 08:00					
61	unloading of order 4241	Fr 28.8.2009 08:00 -- Fr 28.8.2009 09:00									1:00	
62	trip to Pickup of order 5710	Fr 28.8.2009 09:00 -- Fr 28.8.2009 12:51	7,1154	43,4632	H	Fr 28.8.2009 08:00 -- Fr 28.8.2009 12:00	Fr 28.8.2009 12:51			3:51		:51
63	loading of order 5710	Fr 28.8.2009 12:51 -- Fr 28.8.2009 13:51									1:00	
64	trip to Delivery of order 5710	Fr 28.8.2009 13:51 -- Fr 28.8.2009 18:21								4:30		
65	break of journey	Fr 28.8.2009 18:21 -- Fr 28.8.2009 19:06						:45				
66	continuation of journey	Fr 28.8.2009 19:06 -- Fr 28.8.2009 20:45								1:39		
67	break of journey	Fr 28.8.2009 20:45 -- Sa 29.8.2009 05:45						9:00				
68	continuation of journey	Sa 29.8.2009 05:45 -- Sa 29.8.2009 10:15								4:30		
69	break of journey	Sa 29.8.2009 10:15 -- Sa 29.8.2009 11:00						:45				
70	continuation of journey	Sa 29.8.2009 11:00 -- Sa 29.8.2009 15:30								4:30		
71	break of journey	Sa 29.8.2009 15:30 -- Mo 31.8.2009 00:00						32:30				
72	continuation of journey	Mo 31.8.2009 00:00 -- Mo 31.8.2009 04:30								4:30		
73	break of journey	Mo 31.8.2009 04:30 -- Mo 31.8.2009 05:15						:45				
74	continuation of journey	Mo 31.8.2009 05:15 -- Mo 31.8.2009 09:45								4:30		
75	break of journey	Mo 31.8.2009 09:45 -- Mo 31.8.2009 10:30						:45				
76	continuation of journey	Mo 31.8.2009 10:30 -- Mo 31.8.2009 11:30								1:00		
77	break of journey	Mo 31.8.2009 11:30 -- Mo 31.8.2009 20:30						9:00				
78	continuation of journey	Mo 31.8.2009 20:30 -- Tu 01.9.2009 01:00								4:30		
79	break of journey	Tu 01.9.2009 01:00 -- Tu 01.9.2009 01:45						:45				
80	continuation of journey	Tu 01.9.2009 01:45 -- Tu 01.9.2009 06:15								4:30		
81	break of journey	Tu 01.9.2009 06:15 -- Tu 01.9.2009 07:00						:45				
82	trip to Delivery of order 5710	Tu 01.9.2009 07:00 -- Tu 01.9.2009 07:09	-9,0907	38,4359	I	Tu 01.9.2009 07:00 -- We 02.9.2009 07:00	Tu 01.9.2009 07:09			:09		

Figure 5.19: Exemplary results: five-week vehicle tour - scheduling II

	activity log	interval of activity	geo.long.	geo.lat.		time window	arrival	break	wait	travel	load	delay
83	unloading of order 5710	Tu 01.9.2009 07:09 -- Tu 01.9.2009 08:09									1:00	
84	trip to Pickup of order 6511	Tu 01.9.2009 08:09 -- Tu 01.9.2009 09:00								:51		
85	break of journey	Tu 01.9.2009 09:00 -- Tu 01.9.2009 18:00						9:00				
86	trip to Pickup of order 6511	Tu 01.9.2009 18:00 -- Tu 01.9.2009 19:12	-8,5558	39,4456	J	Tu 01.9.2009 09:00 -- Tu 01.9.2009 17:30	Tu 01.9.2009 19:12			1:12		1:42
87	loading of order 6511	Tu 01.9.2009 19:12 -- Tu 01.9.2009 20:12									1:00	
88	trip to Delivery of order 6511	Tu 01.9.2009 20:12 -- We 02.9.2009 00:42								4:30		
89	break of journey	We 02.9.2009 00:42 -- We 02.9.2009 01:27						:45				
90	continuation of journey	We 02.9.2009 01:27 -- We 02.9.2009 04:45								3:18		
91	break of journey	We 02.9.2009 04:45 -- We 02.9.2009 15:45						11:00				
92	continuation of journey	We 02.9.2009 15:45 -- We 02.9.2009 20:15								4:30		
93	break of journey	We 02.9.2009 20:15 -- We 02.9.2009 21:00						:45				
94	continuation of journey	We 02.9.2009 21:00 -- Th 03.9.2009 01:30								4:30		
95	break of journey	Th 03.9.2009 01:30 -- Th 03.9.2009 12:30						11:00				
96	continuation of journey	Th 03.9.2009 12:30 -- Th 03.9.2009 17:00								4:30		
97	break of journey	Th 03.9.2009 17:00 -- Th 03.9.2009 17:45						:45				
98	continuation of journey	Th 03.9.2009 17:45 -- Th 03.9.2009 18:12								:27		
99	waiting time	Th 03.9.2009 18:12 -- Mo 07.9.2009 08:00							85:48			
100	trip to Delivery of order 6511	Mo 07.9.2009 08:00 -- Mo 07.9.2009 08:00	-0,3655	44,5538	K	Mo 07.9.2009 08:00 -- Mo 07.9.2009 14:30	Mo 07.9.2009 08:00					
101	unloading of order 6511	Mo 07.9.2009 08:00 -- Mo 07.9.2009 09:00									1:00	
102	trip to Pickup of order 9021	Mo 07.9.2009 09:00 -- Mo 07.9.2009 10:16								1:16		
103	waiting time	Mo 07.9.2009 10:16 -- Mo 07.9.2009 14:00							3:44			
104	trip to Pickup of order 9021	Mo 07.9.2009 14:00 -- Mo 07.9.2009 14:00	1,1321	45,0753	L	Mo 07.9.2009 14:00 -- Mo 07.9.2009 15:00	Mo 07.9.2009 14:00					
105	loading of order 9021	Mo 07.9.2009 14:00 -- Mo 07.9.2009 15:00									1:00	
106	trip to Delivery of order 9021	Mo 07.9.2009 15:00 -- Mo 07.9.2009 19:30								4:30		
107	break of journey	Mo 07.9.2009 19:30 -- Mo 07.9.2009 20:15						:45				
108	continuation of journey	Mo 07.9.2009 20:15 -- Mo 07.9.2009 23:00								2:45		
109	break of journey	Mo 07.9.2009 23:00 -- Tu 08.9.2009 08:00						9:00				
110	continuation of journey	Tu 08.9.2009 08:00 -- Tu 08.9.2009 12:30								4:30		
111	break of journey	Tu 08.9.2009 12:30 -- Tu 08.9.2009 13:15						:45				
112	continuation of journey	Tu 08.9.2009 13:15 -- Tu 08.9.2009 17:45								4:30		
113	break of journey	Tu 08.9.2009 17:45 -- We 09.9.2009 04:45						11:00				
114	trip to Delivery of order 9021	We 09.9.2009 04:45 -- We 09.9.2009 07:24	10,1550	47,4747	M	We 09.9.2009 07:00 -- We 09.9.2009 12:00	We 09.9.2009 07:24			2:39		
115	unloading of order 9021	We 09.9.2009 07:24 -- We 09.9.2009 08:24									1:00	
116	trip to Pickup of order 10286	We 09.9.2009 08:24 -- We 09.9.2009 12:05	10,2722	49,3955	N	We 09.9.2009 08:00 -- We 09.9.2009 13:00	We 09.9.2009 12:05			3:41		
117	loading of order 10286	We 09.9.2009 12:05 -- We 09.9.2009 13:05									1:00	
118	trip to Delivery of order 10286	We 09.9.2009 13:05 -- We 09.9.2009 16:45								3:40		
119	break of journey	We 09.9.2009 16:45 -- Th 10.9.2009 01:45						9:00				
120	continuation of journey	Th 10.9.2009 01:45 -- Th 10.9.2009 06:15								4:30		
121	break of journey	Th 10.9.2009 06:15 -- Th 10.9.2009 07:00						:45				
122	continuation of journey	Th 10.9.2009 07:00 -- Th 10.9.2009 11:30								4:30		
123	break of journey	Th 10.9.2009 11:30 -- Th 10.9.2009 12:15						:45				

Figure 5.20: Exemplary results: five-week vehicle tour - scheduling III

	activity log	interval of activity	geo.long.	geo.lat.		time window	arrival	break	wait	travel	load	delay
124	continuation of journey	Th 10.9.2009 12:15 -- Th 10.9.2009 13:15								1:00		
125	break of journey	Th 10.9.2009 13:15 -- Th 10.9.2009 22:15						9:00				
126	continuation of journey	Th 10.9.2009 22:15 -- Fr 11.9.2009 02:45								4:30		
127	break of journey	Fr 11.9.2009 02:45 -- Fr 11.9.2009 03:30						:45				
128	continuation of journey	Fr 11.9.2009 03:30 -- Fr 11.9.2009 08:00								4:30		
129	break of journey	Fr 11.9.2009 08:00 -- Fr 11.9.2009 17:00						9:00				
130	trip to Delivery of order 10286	Fr 11.9.2009 17:00 -- Fr 11.9.2009 18:47	-1,1328	53,1136	O	Fr 11.9.2009 08:00 -- Fr 11.9.2009 09:00	Fr 11.9.2009 18:47			1:47		9:47
131	unloading of order 10286	Fr 11.9.2009 18:47 -- Fr 11.9.2009 19:47									1:00	
132	break of journey	Fr 11.9.2009 19:47 -- Mo 14.9.2009 15:30						67:43				
133	trip to Pickup of order 11721	Mo 14.9.2009 15:30 -- Mo 14.9.2009 16:00	-1,2136	52,5811	P	Mo 14.9.2009 08:00 -- Mo 14.9.2009 16:00	Mo 14.9.2009 16:00			:30		
134	loading of order 11721	Mo 14.9.2009 16:00 -- Mo 14.9.2009 17:00									1:00	
135	trip to Delivery of order 11721	Mo 14.9.2009 17:00 -- Mo 14.9.2009 21:30								4:30		
136	break of journey	Mo 14.9.2009 21:30 -- Mo 14.9.2009 22:15						:45				
137	continuation of journey	Mo 14.9.2009 22:15 -- Tu 15.9.2009 02:15								4:00		
138	break of journey	Tu 15.9.2009 02:15 -- Tu 15.9.2009 13:15						11:00				
139	continuation of journey	Tu 15.9.2009 13:15 -- Tu 15.9.2009 17:45								4:30		
140	break of journey	Tu 15.9.2009 17:45 -- Tu 15.9.2009 18:30						:45				
141	continuation of journey	Tu 15.9.2009 18:30 -- Tu 15.9.2009 23:00								4:30		
142	break of journey	Tu 15.9.2009 23:00 -- We 16.9.2009 10:00						11:00				
143	continuation of journey	We 16.9.2009 10:00 -- We 16.9.2009 14:30								4:30		
144	break of journey	We 16.9.2009 14:30 -- We 16.9.2009 15:15						:45				
145	continuation of journey	We 16.9.2009 15:15 -- We 16.9.2009 17:25								2:10		
146	waiting time	We 16.9.2009 17:25 -- Th 17.9.2009 08:00							14:35			
147	trip to Delivery of order 11721	Th 17.9.2009 08:00 -- Th 17.9.2009 08:00	2,2606	41,3205	Q	Th 17.9.2009 08:00 -- Th 17.9.2009 12:00	Th 17.9.2009 08:00					
148	unloading of order 11721	Th 17.9.2009 08:00 -- Th 17.9.2009 09:00									1:00	
149	trip to Pickup of order 10781	Th 17.9.2009 09:00 -- Th 17.9.2009 10:12	2,4949	41,5933	R	Th 17.9.2009 08:00 -- Th 17.9.2009 10:00	Th 17.9.2009 10:12			1:12		:12
150	loading of order 10781	Th 17.9.2009 10:12 -- Th 17.9.2009 11:12									1:00	
151	trip to Delivery of order 10781	Th 17.9.2009 11:12 -- Th 17.9.2009 15:42								4:30		
152	break of journey	Th 17.9.2009 15:42 -- Th 17.9.2009 16:27						:45				
153	continuation of journey	Th 17.9.2009 16:27 -- Th 17.9.2009 19:45								3:18		
154	break of journey	Th 17.9.2009 19:45 -- Fr 18.9.2009 06:45						11:00				
155	continuation of journey	Fr 18.9.2009 06:45 -- Fr 18.9.2009 11:15								4:30		
156	break of journey	Fr 18.9.2009 11:15 -- Fr 18.9.2009 12:00						:45				
157	continuation of journey	Fr 18.9.2009 12:00 -- Fr 18.9.2009 15:58								3:58		
158	waiting time	Fr 18.9.2009 15:58 -- Mo 21.9.2009 08:00							64:02			
159	trip to Delivery of order 10781	Mo 21.9.2009 08:00 -- Mo 21.9.2009 08:00	1,0557	49,5539	S	Mo 21.9.2009 08:00 -- Mo 21.9.2009 09:00	Mo 21.9.2009 08:00					
160	unloading of order 10781	Mo 21.9.2009 08:00 -- Mo 21.9.2009 09:00									1:00	

-------	-------	-------	-------	-------
375:58	197:20	213:38	22:04	150:30
break	**wait**	**travel**	**load**	**delay**
	empty:	20:03		
	loaded:	193:35		

Figure 5.21: Exemplary results: five-week vehicle tour - scheduling IV

rest period from “Th 20.8.2009 16:45 – Mo 24.8.2009 8:30”, lasting 87:45h, is started.

The subsequent scheduling instructions are of the same type as in the first 22 lines. Therefore, the detailed description is finished at this point. In a next step, **compliance with the general restrictions for International Truck Transportation** is considered.

To simplify the analysis of compliance, the cells of columns 8 to 11 vary in color.

- The *yellow sections* represent the activities performed in a daily driving time interval. They include travel time, break time and load time.
- The *orange sections* contain daily rest periods, and
- the *blue sections* include weekly rest periods.

Exemplarily, the week from *line 100* to *line 132* (Mo 07.9.2009 08:00 until Mo 14.9.2009 15:30) is considered:

- Compliance with *daily driving time restrictions*: There are six travel time intervals with associated total travel times of 8:31h, 9:00h, 10:00h, 10:00h, 9:00h, and 1:47h. Twice (as allowed as a maximum) the total travel time was extended to 10 hours, the remaining *total travel times* stay at or under *a maximum of 9 hours* per daily driving interval.
- Compliance with *weekly driving time*: With a total of 48:18h, the maximum weekly driving time of 56 hours is respected.
- Compliance with *breaks*: All *traveling activities* have a *maximum duration of 4:30h* and are interrupted by “breaks of journey” of at least 0:45h (loading activities are also counted as non-driving periods).
- Compliance with *daily rest period*: The week contains five *daily rest periods*: four of reduced 9-hour length and one of regular 11-hour length. The interested reader may notice that there seems to be one more reduced 9-hour daily rest period, as it is allowed. This, however, can be attributed to the waiting time of 3:44h at *line 103*. This waiting time may be added to the subsequent 9-hour daily rest period (3h+9h splitting option); therefore, the associated 9 hours are counted as a regular 11-hour daily rest period. Hence, there are only three reduced daily rest periods in this week, which conforms to the restrictions.

  Another interesting aspect occurs in *line 108*: the vehicle starts a daily rest period, even though it has only reached a total traveling time of 8:31h in the respective travel time interval. The maximum travel time restriction would allow for 29 additional minutes. In this case, however, the restriction “daily rest period has to be taken within 24 hours after the last daily rest period (*24-hour rule*)” comes into effect. At “Mo 07.9.2009 23:00” the end of the last daily rest period (here: weekly rest period) is exactly 15 hours ago. Thus, it is the last possible time to start a daily rest period (which in this case even has to be a reduced 9-hour one) to fulfill the 24-hour rule.
- Compliance with *weekly rest period*: In the scheduling, there are five weekly rest periods with the lengths of 87:45h, 32:30h, 85:48h, 67:43h, and 64:02h. Four weekly

rest periods reach the minimum regular duration of 45 hours, the second weekly rest period *(line 71)* is reduced to 32:30h and needs a 21-hour compensation. This compensation is accomplished directly with the subsequent weekly rest period, where 85:48h of weekly rest period are scheduled. Hereby, the required 45h+21h=66h are more than fulfilled.

- Compliance with *Sunday traffic ban*: No activities are scheduled on Sundays.

The investigation of the exemplary five-week vehicle tour has provided some more insights into how a solution of the MNS procedure looks. The next subsection presents the final results that have actually been achieved with the best MNS parameter settings.

5.5.3 Final Results

The overall solution quality of the investigated real-life problem is clearly dependent on several aspects, especially empty travel time and delay, but also waiting time and total operating time. Unfortunately, an *antithetic behavior* was detected for the objective function criteria *empty travel time* and *delay*. There is no explicit specification how to relatively weight both factors. There is only the preference to weight reduction in empty travel time higher than reduction in delay and reduction in delay higher than reduction in waiting time.

Therefore, it is not possible to present "one" best result. Instead, it makes sense to present a choice of good solutions with different weighting of the preferences. The results are generated with the best parameter settings of Section 5.5.1 for the three promising penalty cost combinations: (40,8), (30,5) and (40,5). Detailed results for the objective function criteria empty travel time, delay, break/wait, and overtime are outlined in Table 5.9. The percentage deviations from the manual planning benchmark are visualized in Figure 5.22.

In the first case, a solution is presented that allocates improvements quite equally between empty travel time and delay. This solution is generated by the penalty cost setting (40,8): empty travel time is reduced by 3.9%, delay by 6.0% and break/wait by 1.6%. The empty-to-all ratio which was 11.2% in the manual planning benchmark is reduced to a level of 10.8%. Since there is a stronger reduction in delay as in empty travel time, some additional "improvement" may be shifted from the reduction of delay to the reduction of empty travel time which is – according to the given preferences – the primary focus.

penalty costs	**empty traveling**	**delay**	**break/wait**	**operating time**
40,8	24808	150122	488489	741140
30,5	23936	156915	485434	737213
40,5	22750	186175	485486	736079

Table 5.9: Final results for best parameter and penalty cost settings (in hours)

This is achieved in a second scenario with penalty cost setting (30,5): here, a reduction of empty travel time of 7.3% is generated. Simultaneously, delay is improved by 1.7% and break/wait by 2.2%. The empty-to-all ratio decreases to a level of 10.5%.

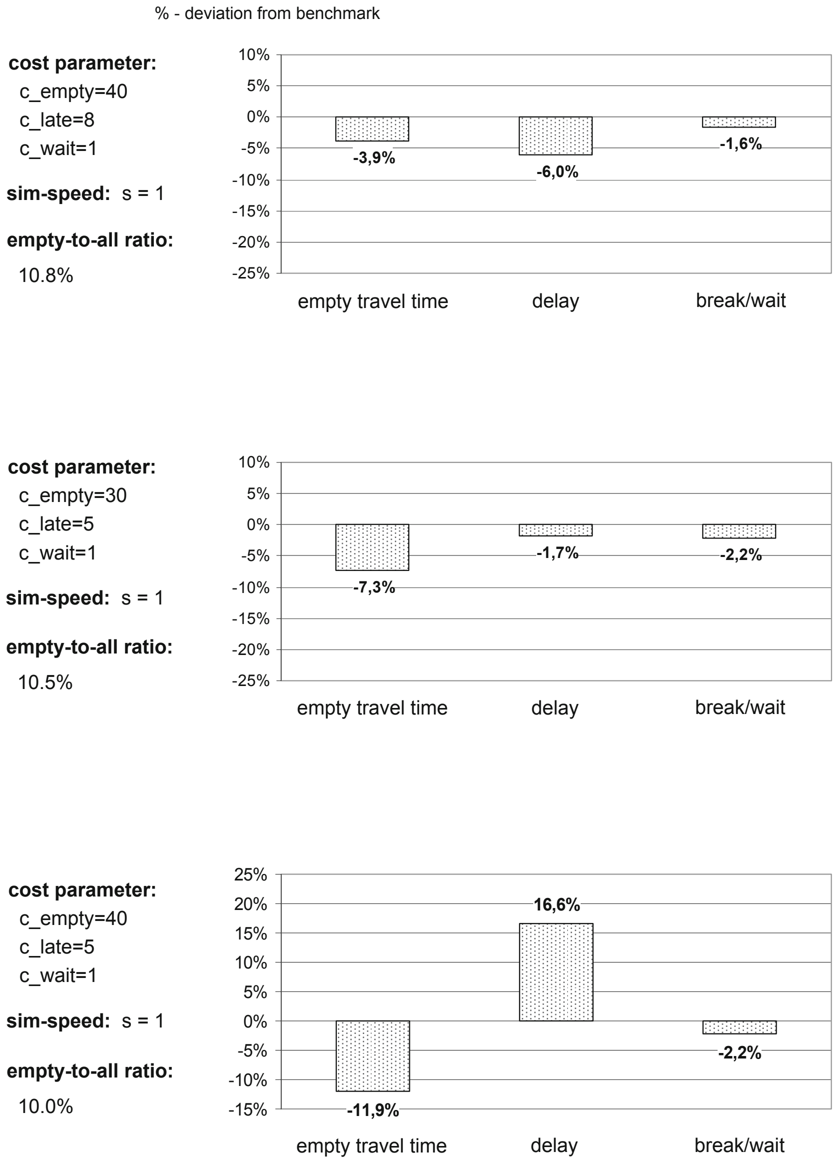


Figure 5.22: Final results for cost parameters (40,8), (30,5) and (40,5)

In a third penalty cost setting (40,5), we try to achieve even further improvements of empty travel time. And indeed, an improvement of 11.9% in terms of empty travel time compared to the manual planning benchmark is produced. The empty-to-all ratio decreases to a level of only 10.0%. However, this improvement comes along with a significant worsening in category delay (16.6%), while break/wait stays at a constant level in comparison to the previous setting (-2.2%). Due to the worsening in delay, this penalty cost setting seems to be only recommendable in situations with very strong preferences for reduction of empty travel time (in relation to reduction of delay).

In total, the second solution with penalty cost setting (30,5) seems to fulfill the required preferences for a "good solution" in the most suitable way: improvements are achieved in comparison to the manual planning benchmark for all objective function criteria, with highest improvements in category empty travel time and second highest improvements in category delay. In the following, we will refer to this second penalty cost setting and the associated results.

5.5.4 Discussion

The final results show that the application of a computer-based real-life planning system is capable of producing planning results with significant reductions in empty travel time and in delay. In the following, the *pros and cons* of an implementation of such a planning system are discussed. An overview is given in Table 5.10.

In a first step, the **possible benefits** are summarized.

- In the five-week test horizon, *1,880 hours of empty travel time are saved* compared to the manual planning benchmark (scenario with penalty cost setting: 30,5). With the assumed average speed of 72 km/h, this equates to 135,360 kilometers. Projected to a whole year with 52 weeks, a saving of approx. 1.4 million empty kilometers would be generated.

- Furthermore, there is an *increase in customer service*: total delay is reduced by 1.7% compared to the manual planning benchmark (scenario with penalty cost setting: 30,5). This corresponds to a weekly reduction of delay of approx. 546 hours.

- In addition to the saving of empty kilometers and to the improvement in service quality, the introduction of a computer-based Decision Support System for the dispatching of an international truck fleet will also cause a *reduction in manual planning effort*. Such a dynamic Decision Support System, however, is not capable of replacing a human dispatcher, it just supports human dispatchers with planning proposals. Some unsystematical data errors or unexpected planning situations will always require the final approval of a human. Nevertheless, the time that a dispatcher needs to perform the planning tasks considered in this work may be reduced. This, for example, allows for a higher number of vehicles to be supervised per dispatcher or for the taking on of other productive work in the freed up time.

- Independently of the quantifiable savings, the implementation of such a project initiates a general improvement process (e.g., for the input data quality and consistency), which may also result in general positive feedback for other parts and planning tasks of the freight forwarding company.

costs		benefits
implementation costs	**operating costs**	**reduction in empty traveled distance** of 7.3%
- guarantee of general and on time data availability	- staff, to keep the system running and for maintainance	five-week horizon: 1,880 h · 72 km/h = 135,360 km
- measures, securing input data quality and consistency	- energy, hardware	→ year-long horizon (approx.): 1.4 million km
- acquisition/development of planning software	- ...	
- acquisition of sufficient hardware resources	**imponderabilities**	**reduction in delay** of 1.7%
- management of data interfaces	- remaining real-life restrictions, not considered	**reduction in manual planning effort**
- training of the dispatchers	- user incompliance	- time for other productive work
- ...	- ...	**general improvement process**
		- positive feedback for other tasks

Table 5.10: Implementation of a dynamic Fleet Management System: costs and benefits

In a second step, the **costs and risks** that have to be set against the potential benefits are outlined: initial implementation cost to acquire the new planning system and to get it running; recurring operating and maintainance costs; and also imponderabilities (risks), which may decrease the extent of actually generated savings.

Implementation costs:

- In the beginning, it has to be ensured that all information that is needed for the planning process is digitally available (*general data availability*). Possibly, the existing information system has to be backed up with additional information. Hereby, an on time data collection process is crucial (*on time data availability*). These preliminary aspects may cause first introduction costs.

- Furthermore, *measures for securing the input data quality and consistency* are needed. This is because a computer based planning system is not capable of finding unsystematic errors in the input data by itself. Undetected data errors may render the planning results partially useless. Such measures may include the installation of automatic checks of input data and also the raising of quality awareness of the people who manually enter the input data (perhaps with a gratification system for error free data handling).

- Introduction costs, furthermore, include the *costs for acquisition/development of planning software*. A freight forwarding company will not usually have the resources to build up a Decision Support System on its own. Hence, a planning solution should be bought from a professional software provider. This provider should have sufficient experience in the freight forwarding sector and also a skilled workforce that allows for customer specific adaptations and prompt service in case of difficulties. However, to avoid paying all the generated savings to the software company, the freight forwarder should have own employees available who understand the planning program and who are able to perform program adaptions themselves.

- *Appropriate hardware equipment* is needed for the planning software. It was shown in this study that even a conventional PC is able to perform the planning of a large problem instance. Thus, the hardware costs should only be moderate.

- Some financial effort is also necessary to establish real time links between all existing databases and the new planning system (*management of data interfaces*). With regard to this, it could be advantageous to introduce a dynamic planning system from the software company providing the associated database and information systems.

- Finally, there are introduction costs for the *training of the dispatchers*. This is not only necessary in order to enable people to use the dynamic planning system, but also to create general acceptance of the new planning system.

Operating and maintainance costs:

- In addition to the introduction costs, there are also some *recurring operating and maintainance costs* for the planning system that have to be considered. These costs specifically occur for the company's own staff which keeps the planning system running and performs necessary program adaptations. Furthermore, permanent energy costs or costs for the replacement of hardware resources must be accounted for.

Possible imponderabilities (risks):

- There may be the problem of user in-compliance. As explained in Section 3.5, the usage of a computer based planning system may be significantly reduced if too many (correct) computer suggestions are manually overwritten by human planners. The only way to cope with this problem is the endeavor to create general acceptance of the software by the human dispatchers.

- Furthermore, it should be mentioned that despite the adapted MNS procedure is including many real-life restrictions, there are some aspects remaining that have not been covered: e.g., explicit planning of driver exchange or further sources of dynamism. The additional consideration of these aspects in a real-life planning, may result in a less significant reduction in empty traveled distance.

After estimation of all savings and cost values, one approach of assessing the advantageousness of an investment in a dynamic planning system could be the calculation of the resulting net present value. Such a detailed quantification, however, is very company specific and therefore shall not be a part of this work.

Chapter 6

Conclusion and Outlook

In this chapter, the methodology, achievements and main findings of this thesis are summarized (Section 6.1). This is followed by some recommendations for further research in the area of Dynamic Fleet Management (Section 6.2).

6.1 Conclusion

This work has investigated a dynamic real-life planning situation that has mostly been neglected in the existing literature: Dynamic Fleet Management for International Truck Transportation with occasional transportation tasks.

The goal of the study was outlined in the first chapter:

> *To design a Dynamic Fleet Management System for International Truck Transportation focusing on occasional transportation tasks that is capable of improving the planning process at a freight forwarding company in terms of empty traveled distance and service quality, hereby taking into account all important European real-life requirements (EC social regulations, working time and traffic bans).*

A number of research questions were posed to guide us in reaching the goal of the study. With the reconsideration of these research questions, the methodology, achievements and findings of this study shall be summarized.

What are the specific characteristics of dynamic planning problems?

This question is investigated in Section 2.1: the term "dynamic" and various sources of dynamism are defined. The *main differences between classical static planning and dynamic planning* are presented. Afterwards, the degree of information availability and the possible reactivity in dynamic planning situations are discussed. This is followed by the investigation of potential measures for the "degree of dynamism", a discussion of appropriate simulation techniques for dynamic algorithms, and five possible ways of performance evaluation.

Where do dynamic planning situations occur in real-life?

In Section 2.2 the *most important dynamic real-life applications* (Pickup Tour for Courier

Steffen Schorpp, *Dynamic Fleet Management for International Truck Transportation,*
DOI 10.1007/978-3-8349-6675-9_6,

Services, Traveling Repairman Problem, Taxi/Dial-A-Ride services, Express Mail Delivery, and International Truck Transportation with occasional transportation tasks) are introduced and connected to the associated theoretical problem definitions. Furthermore, a classification scheme for these dynamic real-life problems is proposed using the characterizing attributes “depot bound/depot free” and “local area/wide area”.

What is the state of the art in the literature on Dynamic Fleet Management?

Since the *literature on dynamic wide area applications is scarce*, the literature review in Chapter 3 focuses on *dynamic vehicle routing in general*. It includes exemplary dynamic real-life publications and algorithm orientated papers. The algorithm orientated publications are divided into three groups, depending on the knowledge of the future: the first two groups do not have any knowledge of the future and therefore only perform “myopic” planning; in contrast to the first group, the second group tries to anticipate the future anyway. Stochastic information about the future is considered only in third group publications, which make explicit use of it.

The investigated algorithms are based on various concepts: local search approaches, various metaheuristics (Tabu Search, Evolutionary Approaches, Variable Neighborhood Search, Ant Colony, Second Objective Function), the heuristic application of exact procedures, rule-based approaches, and multi-agent systems. Most concepts apply classical Best Insertion techniques. The literature review is finally complemented by a look at the most popular dynamic test instances and by publications considering the acceptance of dynamic planning applications in real-life.

What dynamic solution approaches are suitable for a Dynamic Fleet Management System?

In Chapter 4 *two dynamic planning approaches* based on completely different planning ideas *are developed*: an *Assignment based procedure* and an I*nsertion based procedure with Multiple Neighborhood Search*. First stage, both approaches are designed for a local area MLPDPTW and not for the final real-life planning problem. The available test instances for this local area problem are used, to perform extensive tests and to evaluate the strengths and weaknesses of both procedures.

For a self-generated set of test instances, the Assignment based approach produces better results. For some test instances from the literature, however, better performance is observed with Insertion based MNS. *Both procedures are suitable* for the adaptation to the actual real-life planning problem, nevertheless, the *Insertion based MNS procedure is finally chosen*. This is due to its higher robustness, the smaller parameterization effort and less overall complexity.

What general requirements come along with International Truck Transportation in Europe?

In Section 5.1 four important categories of general requirements are analyzed: *Social Regulation EC 561/2006, Social Regulation AETR, Directive EC 2002/15 on working hours, and general driving bans*. The first three aspects are driver related, while the fourth aspect is of general type. For the social regulations, we discuss the requirements for single and team driver mode in detail (daily driving time, weekly and fortnightly driving time, breaks, daily and weekly rest periods). Hereby, we do not only focus on basic rule scheduling, but also elaborate all possible exceptions. We notice that scheduling with exceptions enables better planning results: the use of extended driving time and reduced rest periods, for example, allows for better compliance with the time window restrictions of an urgent order.

The working time requirements of Directive EC 2002/15 do not only include driving time, but "all times during which a driver cannot dispose freely of his time." The respective requirements for weekly working time, breaks and night work are outlined. In terms of general driving bans, a very heterogeneous situation is found, with nearly every European country applying its own restrictions. The European countries can be roughly differentiated into three groups: with general traffic bans, with partial traffic bans, and without traffic bans. As the most common feature, various types of Sunday traffic bans can be identified.

What specific requirements are necessary to cover the planning situation at the cooperating freight forwarding company?

In Section 5.2 the planning requirements of our cooperating freight forwarding company are outlined. An important aspect is the *restricted order-to-vehicle assignment*, which only allows for a matching if the load type of orders and vehicles is identical. This is complemented by some orders' need for a vehicle with hazardous goods equipment. Some vehicles are available in *single driver mode* (70%), some in *team driver mode* (30%).

Orders primarily occur as classical Full-Truckload requests with one Pickup and one Delivery location. However, there may also be *request bundles* with several prespecified load and unload locations whose sequence cannot be changed by the planning procedure. Further real-life requirements concern the scheduling of delayed arrivals and the exchange of drivers.

What simulation speed should be used to evaluate the real-life planning procedure's performance?

It turns out that an appropriate simulation speed is of *decisive impact on solution quality.* Results which are significantly better in all objective function criteria can be generated with slower simulation speeds. This impact is exemplarily demonstrated in Section 5.5.1: with a decrease of simulation speed from s = 120, over s = 5, to s = 1, the average solution quality is remarkably increased. This better performance can be attributed to the improvement components of the MNS real-life planning procedure, which are capable of investigating a larger scope of the solution space if there is more simulation time available. The unequivocally *best performance* is achieved if a *real time simulation* (with a simulation speed of s = 1) is applied.

The simulation speed is *also an important aspect for the parameterization*: the best parameters with a simulation speed of s = 5, are not necessarily identical to the best parameters with a simulation speed of s = 1. This finding seems to be a good reason to also perform the parameterization steps in real time. However, there is the drawback of the long simulation runs for real time simulations that only allow for a limited number of parameter variations to be tested.

How much potential savings can be generated with the application of a computer-based dynamic planning system for International Freight Transportation? Is it reasonable to implement such a Decision Support System?

In Section 5.4 we analyse a five-week real-life test data set from our cooperating freight forwarding company and derive benchmark values from the actually performed manual planning. Afterwards, we apply our adapted MNS real-life procedure to the five-week test data set and compare the computer-based results with the benchmark.

Generally, we discover an *antithetic behavior* of the objectives *reduction of empty travel time* and *reduction in delay*. Therefore, we explore three penalty cost settings (30,5; 40,5; 40,8) that – for a simulation speed of s = 1 – generate promising results for both objective function criteria at the same time.

With the penalty cost setting (40,8), we achieve a *reduction in empty travel time of 3.9%* and a *reduction in delay of 6.0%.* With the penalty cost setting (30,5), we achieve a reduction in empty travel time of 7.3% and a reduction in delay of 1.7%. The best results in terms of empty travel time (reduction of 11.9%) are achieved with penalty costs (40,5); however, in this case there is a significant worsening of delay (16.6%). Further objective function criteria, like break/wait and total operating time, are slightly improved in all three cases.

Whether these savings can justify the implementation of a Decision Support System is discussed in Section 5.5.4. We contrast possible benefits and costs connected with such an implementation. For our cooperating company, for example, we estimate that the introduction of a computer-based planning system would result in a *yearly saving of approx. 1.4 million empty traveled kilometers.* In addition, the manual planning effort could be reduced and a general improvement process could be initiated. On the other hand, there are various introduction and operating costs for a dynamic planning system, as well as possible imponderabilities (like user incompliance).

The comparison of pros and cons mostly contains qualitative aspects, since a detailed monetary quantification would require too much company-specific information. Nevertheless, we assume that savings of the magnitude of 1.4 million empty kilometers per year should make the investment in a computer-based planning system a beneficial decision.

6.2 Recommendations for Further Research

The investigated planning procedure covers many real-life restrictions. However, there are **further** extending options, especially in terms of **possible sources of dynamism**. With dynamically occurring orders, we have only considered the most important dynamic aspect. Future works could also include cancelation or modification of already known customer orders, changes in vehicle travel time and complete vehicle breakdown.

Another extension could be the **inclusion of stochastic information of the future**. Section 3.3.3 has presented some promising ideas for incorporation of such extra information into dynamic planning procedures. However, an important question in this context concerns how this stochastic information could actually be gathered in real-life and also how reliable the gathered data would be.

It also materialized that there is a very close connection between vehicle routing and driver scheduling in wide area Fleet Management. In this work, we took the information on the vehicles' crews as externally given and applied the resulting restrictions to our tour planning. We did not treat driver exchange and the number of drivers per vehicle as decision variables. However, the overall planning quality (concerning the total costs of tour and driver scheduling) may be decisively improved, if **coordinated tour and driver planning** were performed:

- Firstly, a driver has to get home from time to time, which generates extra costs for the freight forwarding company. If it is possible in the tour planning, to assign an order to the driver's vehicle with a destination near to the driver's home location or close to another favourable exchange point, the costs of exchanging the driver could be reduced.

- The second, even more important aspect, deals with the direct impact of a vehicle's crew (number of available drivers; time interval a vehicle is operated by the same driver(s)) on vehicle routing restrictions (EC social regulations).

  A vehicle in *team driver mode* can be operated for a much longer time per week (112 hours maximum/first week, instead of 56 hours maximum/first week) than a vehicle in single driver mode. The resulting better vehicle utilization allows for more orders to be transported and for better compliance with time windows (faster treatment of urgent orders). In coordinated tour and driver planning, the decision to run a vehicle in one or team driver mode could be handled flexibly: one driver mode could be chosen in weeks with a small number of available requests and team driver mode in weeks with many orders; a crew of two drivers could be assigned preferably to vehicles carrying urgent orders. Of course, for this strategy, double personnel costs (team driver mode) and also costs for frequent driver exchange have to be taken into account: If the extra vehicle utilization necessary to justify these extra expenses cannot be generated, it may be better to waive the team driver option. This would be a question to be decided by coordinated tour and driver planning.

  When drivers are *exchanged frequently*, a higher overall vehicle utilization with more transported requests and better compliance with time windows can be achieved. A "fresh driver", whose driving time account is unconsumed and who has no outstanding rest period compensations, is - in tendency - capable of performing a much longer weekly driving time compared to a driver who has spent several weeks on the vehicle and whose driving time is possibly subject to more stringent restrictions (especially limitations due to maximum fortnightly driving time and outstanding rest period compensation). Of course, a strategy with frequent driver changes also generates higher costs for the exchanging procedure (empty kilometers to the exchange point, getting the old driver home and the new driver to the vehicle, overnight accommodation for rest periods away from the vehicle, etc.). The detection of the optimal points in time to perform driver exchanges should be a result of coordinated tour and driver planning.

As we have seen in Section 5.2, the planning problem of the cooperating freight forwarding company consists of two levels: order acquisition (acceptance/rejection decision;

cost/pricing problem) and tour planning. In this work, we have concentrated on tour planning aspects and have excluded the acquisition. However, with our insertion based MNS procedure, it would be a simple undertaking to evaluate possible new orders on the basis of the current schedule. The associated incremental costs for the new order's insertion could be reported within seconds to the responsible human dispatcher, thus, providing **decision support for the acquisition process**.

In future works, a **revenue management system** could be developed which makes use of these dynamically calculated incremental insertion costs. Such a system should also consider the change in future profits or opportunity costs associated with servicing a new potential order. First works on pricing decisions in dynamic fleet management were published by Figliozzi et al. (2007) and Topaloglu and Powell (2007).

Another potential extension to the tour planning problem is the **grouping of customer orders into priority classes**, e.g., A and B. Customers who are willing to pay a little more for an on time service (in compliance with the specified time windows) could choose the tariff of priority level A, while customers with less restrictive demands on punctuality could choose the cheaper tariff of priority level B. In this way, the overall profit of the freight forwarding company may be increased. The service quality, in terms of time window compliance, could be simply differentiated by different treatment of delays in the objective function – with higher costs to penalize delays of priority A orders and smaller costs to penalize delays of priority B orders.

In terms of the procedure's performance, it would be interesting to analyze the impact of yet a further **increase in computational power**. In Section 5.5.1 the approximate quintuplication of the number of calculation steps (from a simulation speed of s = 5 to a simulation speed of s = 1) effected an increase in solution quality in the dimension of 2.2% in empty travel time and of 6.6% in delay (for penalty costs settings (40,8), cp. Figure 5.12). Would yet a further increase of computational power by a factor of five result in similar improvements? An answer could be given, for example, with a parallelized architecture with several processors. For the required problem decomposition, the improvement neighborhoods of MNS with exchange operations between two vehicle tours (neighborhood I) and within one vehicle tour (neighborhood II) seem to be particularly suitable.

Bibliography

Angelelli, E., **Mansini, R.**, **Speranza, M.** (**2004**): A Real-Time Vehicle Routing Model for a Courier Service Problem. In: B. Fleischmann, A. Klose (Eds.), *Distribution Logistics: Advanced Solutions to Practical Problems*, pp. 87–104. Springer Publishing, Berlin.

Attanasio, A., **Cordeau, J.**, **Ghiani, G.**, **Laporte, G.** (**2004**): Parallel Tabu Search Heuristic for the Dynamic Multi-Vehicle Dial-A-Ride Problem. Parallel Computing, Vol. 30, Iss. 3, pp. 377–387.

Beasly, J. (**1990**): OR-Library: Distributing Test Problems by Electronic Mail. Journal of the Operational Research Society, Vol. 41, Iss. 11, pp. 1069–1072.

Beaudry, A., **Laporte, G.**, **Melo, T.**, **Nickel, S.** (**2010**): Dynamic Transportation of Patients in Hospitals. OR Spectrum, Vol. 32, Iss. 1, pp. 77–107.

Bell, W., **Dalberto, L.**, **Fisher, M.**, **Greenfield, A.**, **Jaikumar, R.**, **Kedia, P.**, **Mack, R.**, **Prutzman, P.** (**1983**): Improving the Distribution of Industrial Gases with an Online Computerized Routing and Scheduling Optimizer. Interfaces, Vol. 13, Iss. 6, pp. 4–23.

Bent, R., **van Hentenryck, P.** (**2004**): Scenario-Based Planning for Partially Dynamic Vehicle Routing with Stochastic Customers. Operations Research, Vol. 52, Iss. 6, pp. 977–987.

Bock, S. (**2004**): Echtzeitfähige Steuerung von Speditionsnetzwerken: Nutzung moderner Informations- und Kommunikationstechnologien zur effizienten Durchführung von Transporten: Habilitationsschrift Universität Paderborn. Deutscher Universitätsverlag, Wiesbaden.

Bock, S. (**2010**): Real-Time Control of Freight Forwarder Transportation Networks by Integrating Multimodal Transport Chains. European Journal of Operational Research, Vol. 200, Iss. 3, pp. 733–746.

Branchini, R., **Armentano, V.**, **Lokketangen, A.** (**2009**): Adaptive Granular Local Search Heuristic for a Dynamic Vehicle Routing Problem. Computers and Operations Research, Vol. 36, Iss. 11, pp. 2955–2968.

Branke, J., **Middendorf, M.**, **Noeth, G.**, **Dessouky, M.** (**2005**): Waiting Strategies for Dynamic Vehicle Routing. Transportation Science, Vol. 39, Iss. 3, pp. 298–312.

Brown, G., **Graves, G.** (**1981**): Real-Time Dispatch Of Petroleum Tank Trucks. Management Science, Vol. 27, Iss. 1, pp. 19–32.

Steffen Schorpp, *Dynamic Fleet Management for International Truck Transportation,*
DOI 10.1007/978-3-8349-6675-9,

Bundesverband Güterkraftverkehr Logistik und Entsorgung (BGL) e.V. (**2010**): English-German translation of logistic terms. Interview from 31.08.2010.

Caramia, M., **Italiano, G.**, **Oriolo, G.**, **Pacifici, A.**, **Perugia, A.** (**2002**): Routing a Fleet of Vehicles for Dynamic combined Pickup and Delivery Services. In: P. Chamoni, R. Leisten, A. Martin (Eds.), *Operations Research Proceedings, Duisburg 2001*, pp. 3–8. Springer Publishing, Berlin.

Chen, H., **Hsueh, C.**, **Chang, M.** (**2006**): The Real-Time Time-Dependent Vehicle Routing Problem. Transportation Research Part E, Vol. 42, Iss. 5, pp. 383–408.

Chen, Z., **Xu, H.** (**2006**): Dynamic Column Generation for Dynamic Vehicle Routing with Time Windows. Transportation Science, Vol. 40, Iss. 1, pp. 74–88.

Cheung, B., **Choy, K.**, **Li, C.**, **Shi, J.**, **Tang, J.** (**2008**): Dynamic Routing Model and Solution Methods for Fleet Management with Mobile Technologies. International Journal of Production Economics, Vol. 113, Iss. 2, pp. 694–705.

Christophides, N., **Beasley, J.** (**1984**): The Period Routing Problem. Networks, Vol. 14, Iss. 2, pp. 237–256.

Commercial Regulatory Authority (**2010**): Regulations for International Truck Transportation - Interpretation and Handling. Mr. Lieb (Gewerbeaufsichtsamt Oberbayern), Interview from 05.02.2010.

Commerzbank Research (**2010**): Branchenbericht Transport/Logistik - November 2009. Mr. Labitzke, Interview from 24.06.2010.

Cordeau, J., **Laporte, G.** (**2003**): A Tabu Search Heuristic for the Static Multi-vehicle Dial-a-Ride Problem. Transportation Research Part B, Vol. 37, Iss. 6, pp. 579–594.

Cordeau, J., **Laporte, G.**, **Potvin, J.**, **Savelsbergh, M.** (**2007**): Transportation on Demand: Logistics. In: C. Barnhart, G. Laporte (Eds.), *Handbooks in Operations Research and Management Science: Logistics*, Bd. 14, pp. 429–466. North-Holland Publishing, Amsterdam.

Dantzig, G., **Ramser, J.** (**1959**): The Truck Dispatching Problem. Management Science, Vol. 6, Iss. 1, pp. 80–91.

Dorigo, M. (**1992**): Optimization, Learning and Natural Algorithms. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano (Italy).

Du, T., **Li, E.**, **Chou, D.** (**2005**): Dynamic Vehicle Routing for Online B2C Delivery. Omega, Vol. 33, Iss. 1, pp. 33–45.

Ebben, M., **van der Heijden, M.**, **van Harten, A.** (**2005**): Dynamic Transport Scheduling under Multiple Resource Constraints. European Journal of Operational Research, Vol. 167, Iss. 2, pp. 320–335.

Ernst, A., **Horn, M.**, **Krishnamoorthy, M.**, **Kilby, P.**, **Degenhardt, P.**, **Moran, M.** (**2007**): Static and Dynamic Order Scheduling for Recreational Rental Vehicles at Tourism Holdings Limited. Interfaces, Vol. 37, Iss. 4, pp. 334–341.

European Union (**2002**): Directive 2002/15/EC of the European Parliament and of the Council of 11 March 2002 on the Organisation of the Working Time of Persons Performing Mobile Road Transport Activities. L 80/35.
http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2002:080:0035:0039:EN:PDF, last viewed: 14.02.2010.

European Union (**2006a**): Directive 2006/22/EC of the European Parliament and of the Council of 15 March 2006 on Minimum Conditions for the Implementation of Council Regulations (EEC) No 3820/85 and (EEC) No 3821/85 concerning Social Legislation relating to Road Transport Activities. L 102/35.
http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2006:102:0035:0043:EN:PDF, last viewed: 10.02.2010.

European Union (**2006b**): Regulation (EC) No 561/2006 of the European Parliament and of the Council on the Harmonisation of Certain Social Legislation Relating to Road Transport. L 102/1.
http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2006:102:0001:0013:EN:PDF, last viewed: 22.02.2010.

European Union (**2009**): Regulation (EC) No 1071/2009 of the European Parliament and of the Council on common rules for access to the international road haulage market. L 300/72.
http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2009:300:0072:0087:EN:PDF, last viewed: 29.06.2010.

Eurostat (**2009**): Panorama of Transport 2009.
http://epp.eurostat.ec.europa.eu/cache/ITY_OFFPUB/KS-DA-09-001/EN/KS-DA-09-001-EN.PDF, last viewed: 25.06.2010.

Fabri, A. (**2008**): Behandlung des Dynamischen Pickup and Delivery Vehicle Routing Problems mit Zeitfenstern und Ladebedingungen mittels spezieller Statusgraphen: Dissertation Universität Dortmund.
https://eldorado2.tu-dortmund.de/bitstream/2003/25107/2/Fabri_Behandlung_des_dPDVRPTW.pdf, last viewed: 14.04.2010.

Fabri, A., **Recht, P.** (**2006**): On Dynamic Pickup and Delivery Vehicle Routing with several Time Windows and Waiting Times. Transportation Research Part B, Vol. 40, Iss. 4, pp. 335–350.

Falk, J. (**1995**): Ein Multi-Agentensystem zur Transportplanung und -steuerung bei Speditionen mit Trampverkehr: Entwicklung und Vergleich mit zentralisierten Methoden und menschlichen Disponenten: Dissertation Universität Erlangen-Nürnberg. Infix Publishing, Sankt Augustin.

Federal Ministry for Labor and Social Affairs, Germany (**2009**): Arbeitszeitgesetz from 22.09.2009.
http://bundesrecht.juris.de/bundesrecht/arbzg/gesamt.pdf, last viewed: 26.01.2010.

Figliozzi, M., **Mahmassani, H.**, **Jaillet, P.** (**2007**): Pricing in Dynamic Vehicle Routing Problems. Transportation Science, Vol. 41, Iss. 3, pp. 302–318.

Fisher, M., **Jaikumar, R.**, **van Wassenhove, L.** (**1981**): A Generalized Assignment Heuristic for Vehicle Routing. Networks, Vol. 11, Iss. 2, pp. 109–124.

Fleischmann, B. (**2010**): Vorlesung Internationale Logistik - Kapitel 2: Grundlagen der Standortplanung. Lehrstuhl für Produktion und Logistik, Universität Augsburg.

Fleischmann, B., **Gnutzmann, S.**, **Sandvoß, E.** (**2004**): Dynamic Vehicle Routing Based on Online Traffic Information. Transportation Science, Vol. 38, Iss. 4, pp. 420–433.

Gendreau, M., **Guertin, F.**, **Potvin, J.**, **Seguin, R.** (**2006**): Neighborhood Search Heuristics for a Dynamic Vehicle Dispatching Problem with Pickups and Deliveries. Transportation Research Part C, Vol. 14, Iss. 3, pp. 157–174.

Gendreau, M., **Guertin, F.**, **Potvin, J.**, **Taillard, E.** (**1999**): Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching. Transportation Science, Vol. 33, Iss. 4, pp. 381–390.

Gendreau, M., **Potvin, J.** (**1998**): Dynamic Vehicle Routing and Dispatching. In: T. Crainic, G. Laporte (Eds.), *Fleet Management and Logistics*, pp. 115–126. Springer Publishing, New York.

Ghiani, G., **Guerriero, F.**, **Laporte, G.**, **Musmanno, R.** (**2003**): Real-Time Vehicle Routing: Solution Concepts, Algorithms and Parallel Computing Strategies. European Journal of Operational Research, Vol. 151, Iss. 1, pp. 1–11.

Ghiani, G., **Manni, E.**, **Quaranta, A.**, **Triki, C.** (**2009**): Anticipatory Algorithms for Same-Day Courier Dispatching. Transportation Research Part E, Vol. 45, Iss. 1, pp. 96–106.

Glover, F. (**1989**): Tabu Search - Part I. ORSA Journal on Computing, Vol. 1, Iss. 3, pp. 190–206.

Goel, A. (**2007**): Fleet Telematics: Real-Time Management and Planning of Commercial Vehicle Operations: PhD thesis, University of Leipzig. Springer Publishing, New York.

Goel, A. (**2009**): Vehicle Scheduling and Routing with Drivers' Working Hours. Transportation Science, Vol. 43, Iss. 1, pp. 17–26.

Guntsch, M., **Middendorf, M.** (**2002**): Applying Population based ACO to Dynamic Optimization Problems. In: M. Dorigo, M. Sampels, G. Di Caro (Eds.), *ANTS 2002: Lecture Notes in Computer Science*, pp. 111–122. Springer Publishing, Berlin.

Haghani, A., **Jung, S.** (**2005**): A Dynamic Vehicle Routing Problem with Time-Dependent Travel Times. Computers and Operations Research, Vol. 32, Iss. 11, pp. 2959–2986.

Hanshar, F., **Ombuki-Berman, B.** (**2007**): Dynamic Vehicle Routing using Genetic Algorithms. Applied Intelligence, Vol. 27, Iss. 1, pp. 89–99.

Helay, P., **Moll, R.** (**1995**): A new Extension of Local Search applied to the Dial-A-Ride Problem. European Journal of Operational Research, Vol. 83, Iss. 1, pp. 83–104.

Holland, J. (**1975**): Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. University of Michigan Press, Ann Arbor.

Hvattum, L., **Lokketangen, A.**, **Laporte, G.** (**2006**): Solving a Dynamic and Stochastic Vehicle Routing Problem with a Sample Scenario Hedging Heuristic. Transportation Science, Vol. 40, Iss. 4, pp. 421–438.

Hvattum, L., **Lokketangen, A.**, **Laporte, G.** (**2007**): A Branch-and-Regret Heuristic for Stochastic and Dynamic Vehicle Routing Problems. Networks, Vol. 49, Iss. 4, pp. 330–340.

Ichoua, S., **Gendreau, M.**, **Potvin, J.** (**2000**): Diversion Issues in Real-Time Vehicle Dispatching. Transportation Science, Vol. 34, Iss. 4, pp. 426–438.

Ichoua, S., **Gendreau, M.**, **Potvin, J.** (**2006**): Exploiting Knowledge About Future Demands for Real-Time Vehicle Dispatching. Transportation Science, Vol. 40, Iss. 2, pp. 211–225.

Ivankina, E. (**2004**): Erprobung von Heuristiken für die dynamische Tourenplanung: Diploma thesis. Lehrstuhl für Produktion und Logistik, University of Augsburg.

Jaillet, P., **Wagner, M.** (**2006**): Online Routing Problems: Value of Advanced Information as Improved Competitive Ratios. Transportation Science, Vol. 40, Iss. 2, pp. 200–210.

Jonker, R., **Volgenant, A.** (**1987**): A Shortest Augmenting Path Algorithm for Dense and Sparse Linear Assignment Problems. Computing, Vol. 38, Iss. 4, pp. 325–340.

Karlin, A., **Manasse, L.**, **Rudolph, L.**, **Sleator, D.** (**1988**): Competitive Snoopy Caching. Algorithmica, Vol. 3, pp. 79–119.

Kilby, P., **Prosser, P.**, **Shaw, P.** (**1998**): Dynamic VRPs: A Study of Scenarios. Technical Report, University of Strathclyde (UK).
http://www.cs.st-andrews.ac.uk/~apes/reports/apes-06-1998.ps.gz, last viewed: 20. 04.2010.

Kim, Y., **Mahmassani, H.**, **Jaillet, P.** (**2002**): Dynamic Truckload Truck Routing and Scheduling in Oversaturated Demand Situations. Transportation Research Record, Vol. 1783, pp. 66–71.

Kim, Y., **Mahmassani, H.**, **Jaillet, P.** (**2004**): Dynamic Truckload Routing, Scheduling, and Load Acceptance for Large Fleet Operation with Priority Demands. Transportation Research Record, Vol. 1882, pp. 120–128.

Kok, L., **Meyer, C.**, **Kopfer, H.**, **Schutten, J.** (**2009**): A Dynamic Programming Heuristic for the Vehicle Routing Problem with Time Windows and the European Community Social Legislation. Working Paper, University of Twente, Enschede (The Netherlands).

Kraftfahrtbundesamt (**2010**): Musterabbildung Fahrerkarte für EG Kontrollgerät.
http://www.kba.de/cln_005/nn_124592/DE/ZentraleRegister/EGKontrollgeraet/ Karten/karten__node.html?__nnn=true, last viewed: 03.08.2010.

Krumke, S. (**2001**): Online Optimization - Competitive Analysis and Beyond: Habilitation thesis. Konrad-Zuse-Zentrum für Informationstechnik (ZIB), Berlin.

Krumke, S., **Rambau, J.**, **Torres, L.** (**2002**): Real-Time Dispatching of Guided and Unguided Automobile Service Units With Soft Time Windows. Konrad-Zuse-Zentrum für Informationstechnik (ZIB), Berlin.

Larsen, A. (**2000**): The Dynamic Vehicle Routing Problem: PhD thesis Technical University of Denmark. Department of Mathematical Modelling, Lyngby, Denmark.

Larsen, A., **Madsen, O.**, **Solomon, M.** (**2002**): Partially Dynamic Vehicle Routing - Models and Algorithms. Journal of the Operational Research Society, Vol. 53, Iss. 6, pp. 637–646.

Larsen, A., **Madsen, O.**, **Solomon, M.** (**2004**): The A Priori Dynamic Travelling Salesman Problem with Time Windows. Transportation Science, Vol. 38, Iss. 4, pp. 459–472.

Larsen, A., **Madsen, O.**, **Solomon, M.** (**2008**): Recent Developments in Dynamic Vehicle Routing Systems. In: B. Golden, S. Raghavan, E. Wasil (Eds.), *The Vehicle Routing Problem - Latest Advances and New Challenges*, pp. 199–218. Springer Publishing, New York.

Li, H., **Lim, A.** (**2003**): A Metaheuristic For The Pickup And Delivery Problem With Time Windows. International Journal on Artificial Intelligence Tools, Vol. 12, Iss. 2, pp. 173–186.

Li, J., **Mirchandani, P.**, **Borenstein, J.** (**2009**): Real-Time Vehicle Rerouting Problems with Time Windows. European Journal of Operational Research, Vol. 194, Iss. 3, pp. 711–727.

Liao, T. (**2004**): Tabu Search Algorithm for Dynamic Vehicle Routing Problems under Real-Time Information. Transportation Research Record, Vol. 1882, pp. 140–149.

Lund, K., **Madsen, O.**, **Rygaard, J.** (**1996**): Vehicle Routing Problems with varying Degrees of Dynamism: Working Paper Technical University of Denmark. Department of Mathematical Modelling, Lyngby, Denmark.

Magalhaes, J., **Sousa, J.** (**2006**): Dynamic VRP in Pharmaceutical Distribution - A Case Study. Central European Journal of Operations Research, Vol. 14, Iss. 2, pp. 177–192.

Mahmassani, H., **Kim, Y.**, **Jaillet, P.** (**2000**): Local Optimization Approaches to Solve Dynamic Commercial Fleet Management Problems. Transportation Research Record, Vol. 1733, pp. 71–79.

Mes, M., **van der Heijden, M.**, **Schuur, P.** (**2010**): Look-Ahead Strategies for Dynamic Pickup and Delivery Problems. OR Spectrum, Vol. 32, Iss. 2, pp. 395–421.

Mes, M., **van der Heijden, M.**, **van Harten, A.** (**2007**): Comparison of Agent-Based Scheduling to Look-Ahead Heuristics for Real-Time Transportation Problems. European Journal of Operational Research, Vol. 181, Iss. 1, pp. 59–75.

Ministry of Social Affairs, Baden-Württemberg (**2010**): Regulations for International Truck Transportation - Interpretation and Handling. Mr. Morath (Sozialministerium Baden-Württemberg), Interview from 17.02.2010.

Mitrovic-Minic, S., **Krishnamurti, R.**, **Laporte, G.** (**2004**): Double-Horizon Based Heuristics for the Dynamic Pickup and Delivery Problem with Time Windows. Transportation Research Part B, Vol. 38, Iss. 8, pp. 669–685.

Mitrovic-Minic, S., **Laporte, G.** (**2004**): Waiting Strategies for the Dynamic Pickup and Delivery Problems with Time Windows. Transportation Research Part B, Vol. 38, Iss. 7, pp. 635–655.

Mladenovic, N., **Hansen, P.** (**1997**): Variable Neighborhood Search. Computers and Operations Research, Vol. 24, Iss. 11, pp. 1097–1100.

Montemanni, R., **Gambardella, L.**, **Rizzoli, A.**, **Donati, A.** (**2005**): Ant Colony System for a Dynamic Vehicle Routing Problem. Journal of Combinatorial Optimization, Vol. 10, Iss. 4, pp. 327–343.

Nanry, W., **Barnes, W.** (**2000**): Solving the Pickup and Delivery Problem with Time Windows using Reactive Tabu Search. Transportation Research Part B, Vol. 34, Iss. 2, pp. 107–121.

Okhrin, I., **Richter, K.** (**2008**): The Real-Time Vehicle Routing Problem. In: J. Kalcsics, S. Nickel (Eds.), *Operations Research Proceedings, Saarbrücken 2007*, pp. 141–146. Springer Publishing, Berlin.

Osman, I. (**1993**): Metastrategy Simulated Annealing And Tabu Search Algorithms For The Vehicle Routing Problem. Annals of Operations Research, Vol. 41, Iss. 4, pp. 421–451.

Pankratz, G. (**2002**): Speditionelle Transportdisposition: Modell- und Verfahrensentwicklung unter Berücksichtigung von Dynamik und Fremdvergabe: Dissertation Fern-Universität Hagen. Deutscher Universitätsverlag, Wiesbaden.

Pankratz, G. (**2005**): Dynamic Vehicle Routing by Means of a Genetic Algorithm. International Journal of Physical Distribution and Logistics Management, Vol. 35, Iss. 5, pp. 362–382.

Pfohl, H., **Schäfer, C.** (**1998**): Analyse des Beschaffungsverhaltens von Industrie- und Handelsunternehmen zur Aufdeckung von Zeitpuffern im Beschaffungsentscheidungsprozess - Ergebnisse einer Unternehmensbefragung. Working Paper, Technical University of Darmstadt (Germany).

Potvin, J., **Xu, Y.**, **Benyahia, I.** (**2006**): Vehicle Routing and Scheduling with Dynamic Travel Times. Computers and Operations Research, Vol. 33, Iss. 4, pp. 1129–1137.

Powell, W. (**1996**): A Stochastic Formulation Of The Dynamic Assignment Problem With An Application To Truckload Motor Carriers. Transportation Science, Vol. 30, Iss. 3, pp. 195–219.

Powell, W., **Marar, A.**, **Gelfand, J.**, **Bowers, S.** (**2002**): Implementing Real-Time Optimization Models: A Case Application From The Motor Carrier Industry. Operations Research, Vol. 50, Iss. 4, pp. 571–581.

Powell, W., **Snow, W.**, **Cheung, R.** (**2000a**): Adaptive Labeling Algorithms For The Dynamic Assignment Problem. Transportation Science, Vol. 34, Iss. 1, pp. 50–66.

Powell, W., **Towns, M.**, **Marar, A.** (**2000b**): On the Value of Optimal Myopic Solutions for Dynamic Routing and Scheduling Problems in the Presence of User Noncompliance. Transportation Science, Vol. 34, Iss. 1, pp. 67–85.

Psaraftis, H. (**1988**): Dynamic Vehicle Routing Problems. In: B. Golden, A. Assad (Eds.), *Vehicle Routing: Methods and Studies*, pp. 223–248. North-Holland Publishing, Amsterdam.

Psaraftis, H. (**1995**): Dynamic Vehicle Routing: Status and Prospects. Annals of Operations Research, Vol. 61, Iss. 1, pp. 143–164.

Pureza, V., **Laporte, G.** (**2008**): Waiting and Buffering Strategies for the Dynamic Pickup and Delivery Problem with Time Windows. Infor, Vol. 46, Iss. 3, pp. 165–176.

Regan, A., **Mahmassani, H.**, **Jaillet, P.** (**1995**): Improving Efficiency Of Commercial Vehicle Operations Using Real-Time Information: Potential Uses and Assignment Strategies. Transportation Research Record, Vol. 1493, pp. 188–198.

Regan, A., **Mahmassani, H.**, **Jaillet, P.** (**1996**): Dynamic Decision Making for Commercial Fleet Operations using Real-Time Information. Transportation Research Record, Vol. 1537, pp. 91–97.

Regan, A., **Mahmassani, H.**, **Jaillet, P.** (**1998**): Evaluation Of Dynamic Fleet Management Systems: Simulation Framework. Transportation Research Record, Vol. 1645, pp. 176–184.

Rochat, Y., **Taillard, E.** (**1995**): Probabilistic Diversification and Intensification in Local Search for Vehicle Routing. Journal of Heuristics, Vol. 1, Iss. 1, pp. 147–167.

Sandvoß, E. (**2002**): Dynamische Tourenplanung auf der Basis von Online-Verkehrsinformationen: Dissertation Universität Augsburg. Pro Business Publishing, Berlin.

Savelsbergh, M., **Sol, M.** (**1998**): DRIVE: Dynamic Routing of Independent Vehicles. Operations Research, Vol. 46, Iss. 4, pp. 474–490.

Schumann, R., **Timmermann, T.**, **Timm, I.** (**2009**): Transportation Planning in Dynamic Environments. In: B. Fleischmann, K. Borgwardt, R. Klein, A. Tuma (Eds.), *Operations Research Proceedings Augsburg 2008*, pp. 319–324. Springer Publishing, Berlin.

Shieh, H., **May, M.** (**1998**): Online Vehicle Routing with Time Windows: Optimization-Based Heuristics Approach for Freight Demands Requested in Real-Time. Transportation Research Record, Vol. 1617, pp. 171–178.

Sleator, D., **Tarjan, R.**, **Horowitz, E.** (**1985**): Amortized Efficiency of List Update and Paging Rules. Communications of the ACM, Vol. 28, Iss. 2, pp. 202–208.

Solomon, M. (**1987**): Algorithms For The Vehicle Routing And Scheduling Problems With Time Window Constraints. Operations Research, Vol. 35, Iss. 2, pp. 254–265.

Spivey, M., **Powell, W.** (**2004**): The Dynamic Assignment Problem. Transportation Science, Vol. 38, Iss. 4, pp. 399–419.

Statistisches Bundesamt Deutschland (**2010**): Strukturerhebung im Dienstleistungsbereich Fachserie 9 Reihe 1 - Verkehr und Nachrichtenübermittlung - 2001 bis 2007.
https://www-ec.destatis.de/csp/shop/sfg/bpm.html.cms.cBroker.cls?cmspath=struktur,vollanzeige.csp&ID=1013185, last viewed: 25.06.2010.

Staub, H., **Kluge, V.**, **Helm, J.**, **Canaris, C.** (**2004**): Handelsgesetzbuch - Großkommentar §§425 - 452. De Gruyter Publishing, Berlin.

Stumpf, P. (**1998**): Tourenplanung im speditionellen Güterfernverkehr: Dissertation Universität Augsburg. GVB Publishing, Nürnberg.

Taillard, E. (**1994**): Parallel Iterative Search Methods for Vehicle Routing Problems. Networks, Vol. 23, Iss. 8, pp. 661–673.

Taillard, E., **Badeau, P.**, **Gendreau, M.**, **Guertin, F.**, **Potvin, J.** (**1997**): A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows. Transportation Science, Vol. 31, Iss. 2, pp. 170–186.

Tang, H., **Hu, M.** (**2005**): Dynamic Vehicle Routing Problem with Multiple Objectives. Transportation Research Record, Vol. 1923, pp. 199–207.

Teodorovic, D., **Radivojevic, G.** (**2000**): A Fuzzy Logic Approach to Dynamic Dial-A-Ride Problem. Fuzzy Sets and Systems, Vol. 116, Iss. 1, pp. 23–33.

Tjokroamidjojo, D., **Kutanoglu, E.**, **Taylor, G.** (**2006**): Quantifying the Value of Advance Load Information in Truckload Trucking. Transportation Research Part E, Vol. 42, Iss. 4, pp. 340–357.

Topaloglu, H., **Powell, B.** (**2007**): Incorporating Pricing Decisions into the Stochastic Dynamic Fleet Management Problem. Transportation Science, Vol. 41, Iss. 3, pp. 281–301.

United Nations Economic Commission for Europe (**2006**): AETR - European Agreement Concerning the Work of Crews of Vehicles Engaged in International Road Transport.
http://www.unece.org/trans/main/sc1/sc1aetr.html, last viewed: 15.01.2010.

van Hemert, J., **La Poutre, J.** (**2004**): Dynamic Routing Problems with Fruitful Regions: Models and Evolutionary Computation. In: X. Yao, E. Burke, J. Lozano (Eds.), *Parallel Problem Solving from Nature*, pp. 690–699. Springer Publishing, Berlin.

Vogel, U. (**2010**): Fahrer Jahrbuch 2010. Huss Publishing, München.

Willi Betz Logistik (**2010**): Pictures of different Vehicle Types. Internal Source, 10.05.2010.

Wilson, N., **Colvin, N.** (**1977**): Computer Control for the Rochester Dial-a-Ride System. Technical Report TR-77-22.

Wilson, N., **Sussmann, J.**, **Wong, H.**, **Higonnet, T.** (**1971**): Scheduling Algorithms for Dial-a-Ride Systems. Urban Systems Laboratory Report TR-70-13.

Wilson, N., **Weissberg, H.** (**1976**): Advanced Dial-a-Ride Algorithms Research Project: Final Report. Technical Report TR-76-20.

Xiang, Z., **Chu, C.**, **Chen, H.** (**2008**): The Study of a Dynamic Dial-a-Ride Problem under Time-Dependent and Stochastic Environments. European Journal of Operational Research, Vol. 185, Iss. 2, pp. 534–551.

Yang, J., **Jaillet, P.**, **Mahmassani, H.** (**1999**): On-Line Algorithms for Truck Fleet Assignment and Scheduling under Real-Time Information. Transportation Research Record, Vol. 1667, pp. 107–113.

Yang, J., **Jaillet, P.**, **Mahmassani, H.** (**2004**): Real-Time Multivehicle Truckload Pickup and Delivery Problems. Transportation Science, Vol. 38, Iss. 2, pp. 135–147.

Appendix

Steffen Schorpp, *Dynamic Fleet Management for International Truck Transportation,*
DOI 10.1007/978-3-8349-6675-9,

A Pseudocode of MNS Improvement Neighborhoods

```
Refresh Tabu List according to actual system time
Calculate decreasing cost ranking for als vehicle tours

WHILE (simulation_time < t_fixtime) {
      Identify feasible pair of vehicles for neighborhood operation, preferably a combination of a
      ''cheap'' and an ''expensive'' vehicle tour, taking into account the Tabu list
        -> vehicle_a and vehicle_b
      Determine the exchangeable requests for vehicle_a and vehicle_b
        -> exchangeable_req_a and exchangeable_req_b
      Calculate the initial cost for the tours of vehicle_a and vehicle_b
        -> initial_cost
      Set best_cost = 999 999 999; best_tourroute_a = NULL; best_tourroute_b = NULL

      FOR (i=0; i<exchangeable_req_a; i++) {
            FOR (j=0; j<exchangeable_req_b; j++) {
                  IF (request i is compatible with vehicle B && request j is compatible with vehicle A) {
                        Extract requests i and j from the tours of vehicle_a and vehicle_b, respectively
                        Apply Best-Reinsertion for request i into tour B
                        Apply Best-Reinsertion for request j into tour A
                        Calculate the new cost for the tours of vehicle_a and vehicle_b
                          -> new_cost
                        IF (new_cost < best_cost) {
                              best_cost = new_cost
                              best_tourroute_a = new tour of vehicle A
                              best_tourroute_b = new tour of vehicle B
                        }
                  }
            }
      }

      FOR (i=0; i<exchangeable_req_a; i++) {
            IF (request i is compatible with vehicle B) {
                  Extract request i from the tour of vehicle_a
                  Apply Best-Reinsertion for request i into tour B
                  Calculate the new cost for the tours of vehicle_a and vehicle_b
                    -> new_cost
                  IF (new_cost < best_cost) {
                        best_cost = new_cost
                        best_tourroute_a = new tour of vehicle A
                        best_tourroute_b = new tour of vehicle B
                  }
            }
      }

      FOR (j=0; j<exchangeable_req_b; j++) {
            IF (request j is compatible with vehicle A) {
                  Extract request j from the tour of vehicle_b
                  Apply Best-Reinsertion for request j into tour A
                  Calculate the new cost for the tours of vehicle_a and vehicle_b
                    -> new_cost
                  IF (new_cost < best_cost) {
                        best_cost = new_cost
                        best_tourroute_a = new tour of vehicle A
                        best_tourroute_b = new tour of vehicle B
                  }
            }
      }

      IF (best_cost < initial_cost) {
            Set tour of vehicle A = best_tourroute_a
            Set tour of vehicle B = best_tourroute_b
      }
}
```

Table 1: Pseudocode: λ-1 interchange (neighborhood I)

Steffen Schorpp, *Dynamic Fleet Management for International Truck Transportation,*
DOI 10.1007/978-3-8349-6675-9,

```
Calculate decreasing cost ranking for all vehicle tours

WHILE (simulation_time < t_fixtime) {
      Choose a vehicle, according to decreasing cost ranking
        -> current_vehicle
      Determine the exchangeable requests for current_vehicle
        -> exchangeable_requests
      Calculate the initial cost for the tour of current_vehicle
        -> initial_cost
      Extract the exchangeable requests from the vehicle's tour
        -> extracted_tour
      Generate all sequence permutations for possible re-insertion of the exchangeable requests
        -> permutations
      Set best_cost = initial_cost; best_tourroute = initial_tourroute

      FOR (i=0; i<permutations; i++) {
            Apply Best-Reinsertion of exchangeable_requests into extracted_tour
            in the sequence of permutation i
              -> new_tour
            Calculate the cost for new_tour of current_vehicle
              -> new_cost
            IF (new_cost < best_cost) {
                  best_cost = new_cost
                  best_tourroute = new_tour
            }
      }

      Set tour of current_vehicle = best_tourroute
}
```

Table 2: Pseudocode: intraroute optimal sequence (neighborhood II)

```
Calculate the current plan's objective value
  -> initial_objective
Duplicate the current plan for back-up
  -> initial_solution
Calculate decreasing cost ranking for all vehicle tours
Initialize a list all_exchangeable_requests

WHILE (simulation_time < t_fixtime) {
      FOR (i=0; i<number_of_vehicles; i++) {
            Choose most expensive vehicle, according to decreasing cost ranking
              -> current_vehicle
            Determine exchangeable requests for current_vehicle
              -> exchangeable_requests
            Append exchangeable_requests to the list all_exchangeable_requests
            Extract exchangeable_requests from current_vehicle's tour
      }
      WHILE (size of all_exchangeable_requests > 0) {
            Remove first request of all_exchangeable_requests
              -> first_request
            Apply Best-Reinsertion for first_request over all vehicle tours
      }
      BREAK
}

IF (Re-Insertion of all extracted requests was successful) {
      Calculate the new plan's objective value
        -> new_objective
      IF (new_objective > initial_objective) {
            Reconstruct initial solution
      }
}
ELSE {
      Reconstruct initial_solution
}
```

Table 3: Pseudocode: complete solution rebuild (neighborhood III)

B Parameterization - Detailed Results

Appendix B contains the detailed planning results of the parameterization process in Section 5.5.1. The adapted MNS procedure is applied to the real-life test scenario.

	b	
a	empty travel time	delay
	break/wait	operating time

penalty cost "traveling empty"	penalty cost "delay" 1		5		8		10	
1	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-
8	24133	221255	-	-	-	-	-	-
	465770	717746	-	-	-	-	-	-
10	23246	243138	-	-	-	-	-	-
	465497	716586	-	-	-	-	-	-
20	20281	350737	26135	139345	28174	118806	-	-
	467961	716086	484150	738128	487316	743333	-	-
30	-	-	24246	170913	26805	144508	27424	128188
	-	-	485496	737585	489000	743648	489660	744927
40	-	-	23355	195087	25364	161562	26278	143374
	-	-	485167	736366	488883	742091	490328	744449

Table 4: Parameterization of penalty costs (simulation speed s = 5, improvement neighborhoods I:II 66:33, anticipation horizon 10 min) - Detailed Results (in hours)

penalty cost "traveling empty"	penalty cost "delay" 1		5		8		10	
1	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-
20	-	-	26157	136544	-	-	-	-
	-	-	486986	740986	-	-	-	-
30	-	-	24171	159541	26319	135305	27242	123891
	-	-	485833	737847	486831	740994	488387	743472
40	-	-	22969	193705	24817	150976	-	-
	-	-	484724	735536	487621	740281	-	-

Table 5: Parameterization of penalty costs (simulation speed s = 1, improvement neighborhoods I:II 66:33, anticipation horizon 10 min) - Detailed Results (in hours)

Steffen Schorpp, *Dynamic Fleet Management for International Truck Transportation,*
DOI 10.1007/978-3-8349-6675-9,

		empty traveling	delay	break/wait	operating time
anticip. horiz.	5 min	24450	175323	485956	738249
	10 min	24246	170913	485496	737585
	30 min	24610	170142	488326	740780
	60 min	24545	173351	486564	738953
	90 min	24561	177846	485891	738295
	120 min	24591	187401	486199	738633

Table 6: Parameterization of anticipation horizon (simulation speed s = 5, improvement neighborhoods I:II 66:33, penalty costs (30,5)) - Detailed Results (in hours)

		empty traveling	delay	break/wait	operating time
anticip. horiz.	5 min	24193	162010	484880	736916
	10 min	24171	159541	485833	737847
	30 min	23966	158126	485019	736828
	60 min	23937	159509	485865	737646
	90 min	24076	168449	486213	738133
	120 min	24042	164628	485491	737377

Table 7: Parameterization of anticipation horizon (simulation speed s = 1, improvement neighborhoods I:II 66:33, penalty costs (30,5)) - Detailed Results (in hours)

		empty traveling	delay	break/wait	operating time
% I- % II	100-0	24455	164946	486402	738700
	75-25	24581	168246	488889	741313
	66-33	24246	170913	485496	737585
	50-50	24165	169027	487310	739317
	33-66	24579	177323	485439	737862
	25-75	24407	170408	486550	738800
	0-100	26193	220953	491455	745491

Table 8: Parameterization "allocation of improvement time" (simulation speed s = 5, anticipation horizon 10 min, penalty costs (30,5)) - Detailed Results (in hours)

		empty traveling	delay	break/wait	operating time
% I- % II	100-0	24008	154654	487019	738870
	75-25	24071	167129	486340	738254
	66-33	24171	159541	485833	737847
	50-50	24164	166108	484947	736955
	33-66	24081	161118	484747	736671
	25-75	24152	166242	485971	737966
	0-100	26055	214509	489198	743096

Table 9: Parameterization "allocation of improvement time" (simulation speed s = 1, anticipation horizon 10 min, penalty costs (30,5)) - Detailed Results (in hours)