International Series in Operations Research & Management Science

Volume 177

Series Editor

Frederick S. Hillier
Stanford University, CA, USA

Special Editorial Consultant

Camille C. Price
Stephen F. Austin, State University, TX, USA

For further volumes:
http://www.springer.com/series/6161

Jan Fabian Ehmke

Integration of Information and Optimization Models for Routing in City Logistics

Springer

Jan Fabian Ehmke
Carl-Friedrich Gauss Department
of Economics
University of Braunschweig
Mühlenpfordtstraße 23
38106 Braunschweig
Germany

ISSN 0884-8289
ISBN 978-1-4614-3627-0 ISBN 978-1-4614-3628-7 (eBook)
DOI 10.1007/978-1-4614-3628-7
Springer New York Heidelberg Dordrecht London

Library of Congress Control Number: 2012935233

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Foreword

Today, logistics service providers face a demand for increasing temporal accuracy with respect to deliveries. Examples are just in time supply of material for assembly belt production, and attended home delivery of groceries. For industry as well as for private households, tight time windows for delivery are typically stipulated by contract. Consequently, performing efficient operations while keeping assured time windows are major challenges for vehicle routing.

Especially in metropolitan areas dense traffic at certain times of the day impedes the compliance with time windows on pre-calculated routes. Besides hard to predict spontaneous traffic jams, more regular traffic patterns occur, which typically result from flows in- or outbound of the city center. Integrating these regular traffic patterns into vehicle routing can significantly increase service quality and can decrease costs of customer delivery.

Considering daytime-dependent travel times for routing is truly an emerging field with only little work done so far. In particular, recent papers have mostly addressed isolated topics, often in the context of a case study. The reluctance of researchers to address this field may stem from the necessary interplay of engineering, information systems, and operations research techniques.

Data come first: observations of a huge number of motorists are presupposed over a sufficiently long period. Second, the individual recordings have to be transformed into general measures by means of data aggregation. Third, a way to determine shortest paths in a time-varying network efficiently has to be developed. Additionally, routing algorithms have to be transferred into the domain of time-varying travel times. Finally, application cases are to be defined to eventually assess benefits of routing with dynamic travel times.

The main advantage of this book is the interdisciplinary integration from data acquisition to vehicle routing. Jan Ehmke takes up recent approaches and integrates them into an overall structure, always with a strong bias towards a possible implementation. To this end, this book lays the foundation for the development of more elaborated methodological support for city logistics of the future.

February 2012 Dirk Christian Mattfeld

Contents

Abbreviations

ATIS	Advanced Traveler Information System
BESTUFS	BEST Urban Freight Solutions
CART	Classification And Regression Tree
CD	Compact Disc
CIVITAS	City VITAlity Sustainability
CPU	Central Processing Unit
CRISP	Cross Industry Standard Process
CS	Customer Scenario
DBI	Davies-Bouldin Index
DLR	German Aerospace Center (Deutsches Zentrum für Luft- und Raumfahrt)
DM	Data Mining
DP	Dynamic Programming
EDA	Exploratory Data Analysis
EM	Expectation Maximization (algorithm)
FA	Floating Car Data Average
FCD	Floating Car Data
FH	Floating Car Data Hourly Average
FIFO	First-In First-Out principle
FW	Floating Car Data Weighted Average
GDF	Geographic Data File standard
GIS	Geographic Information System
GPS	Global Positioning System
HCM	Highway Capacity Manual
ID3	Iterative Dichotomiser 3
IH	Insertion Heuristic
ISO	International Organization for Standardization
KDD	Knowledge Discovery in Databases
km/h	Kilometers per hour
KML	Keyhole Markup Language
KNIME	KoNstanz Information MinEr

NN	Nearest Neighbor heuristic
NP	Nondeterministic-Polynomial
OD	Origin-Destination pair
OECD	Organization for Economic Cooperation and Development
OGC	Open Geospatial Consortium
OR	Operations Research
RAD	Relative Absolute Difference
RAM	Random-Access Memory
RD	Roadmap Distance
RT	Roadmap Travel Time
SAV	SAVings heuristic
SOM	Self-Organizing Map
SPP	Shortest Path Problem
SQL	Structured Query Language
TDIH	Time-Dependent Insertion heuristic
TDNN	Time-Dependent Nearest Neighbor heuristic
TDNNDP	Time-Dependent Nearest Neighbor based on Dynamic Programming heuristic
TDIHDP	Time-Dependent Insertion based on Dynamic Programming heuristic
TDSAV	Time-Dependent SAVings heuristic
TDSPP	Time-Dependent Shortest Path Problem
TDTSP	Time-Dependent Traveling Salesman Problem
TDVRP	Time-Dependent Vehicle Routing Problem
TDVRPTW	Time-Dependent Vehicle Routing Problem with Time Windows
TS	Tabu Search
TSP	Traveling Salesman Problem
UCC	Urban Consolidation Center
UK	United Kingdom
US	United States (of America)
VRP	Vehicle Routing Problem
VRPTW	Vehicle Routing Problem with Time Windows
XML	Extensible Markup Language

Symbols

(i, j)	Edge from node i to node j
$a_{i,j}(t)$	Arrival time at node j when departing at time t from node i
a	A target node
a_j	Arrival time at node j
b	A time bin
b_i	Demand of customer i
$c_{i,j}$	Static costs (distances, travel times, …) assigned to edge (i, j)
$c_{i,j}(t)$	Time-dependent costs (distances, travel times, …) assigned to edge (i, j), depending on time t of departure at i
c_i	Centroid of cluster C_i
$d_{i,j}$	Static costs (distances, travel times, …) for a path between i and j
$d_{i,j}(t)$	Time-dependent costs (distances, travel times, …) between i and j, depending on time t of departure at i
$\delta_{i,j}^k$	Interval of linearization for link (i, j) at boundary k
$\delta(C_i, C_j)$	Separation between clusters i and j
e_i	Earliest begin of service for customer i
g_i	Service time for customer i
l	A link
l_i	Latest begin of service for customer i
o_i	Medoid of cluster C_i
s	A source node
$s_{i,j}$	Savings for adjacent nodes i and j
$s_{i,j}^k$	Slope for linearization of link (i, j) at rear boundary of time bin k
$\tau_{i,j}(t)$	Travel time from node i to node j when departing at time t from node i
v_b	Speed in time bin b
v_b^l	Median of speed measurements for link l in time bin b
$s(C_i)$	Function denoting the variation within cluster C_i
$w(k)$	Function for evaluation of cluster quality depending on the number of clusters k
z_i	Boundary at the rear end of time bin i
A	A set of target nodes

B	The total number of time bins
C	A cluster
D	A distance matrix
$D(t)$	A time-dependent distance matrix
$D[u]$	An array of tentative distances for node u
E	A set of edges
G	A graph
H	The maximum number of elements retained
K	A set of vehicles
L	A set of tours
P	A tour
P^{arr}	Arrival time of a tour P
Q	The maximum capacity of a vehicle
S	A set of source nodes
T	A set of time bins
T_i	Departure time at node i
V	A set of nodes

Part I
Problem Description

Chapter 2
City Logistics

City logistics service providers are expected to offer high quality, reasonably priced delivery services in the environment of congested urban areas. The role of city logistics service providers has become more and more important in recent years, since just-in-time concepts have found their way into complex supply chains. This is reflected by challenging restrictions for delivery in terms of tight delivery time windows. Furthermore, city logistics service providers are often the only physical and legal activity perceived by the customer, leading to increasing importance of reliability and service quality of delivery. Online retail, for example, makes consumers believe that goods are available at all times in almost no time at almost any costs, but delivery concepts are actually very demanding.

In this chapter, challenges for city logistics service providers are highlighted, especially with regard to increasing traffic volumes in urban areas and increasing complexity of supply chains (Sect. 2.1). Planning of reliable delivery tours asks for a more sophisticated planning approach, which can be derived from city logistics concepts. To this end, services of city logistics service providers are analyzed in relation with optimization of urban freight transportation systems (Sect. 2.2). City logistics concepts follow an integrated approach, aligning commercial activities with requirements of different stakeholders. Corresponding methodology is exemplified in Sect. 2.3, aiming at the modeling of the urban freight transportation system as a whole. Subsequently, the perspective is constricted to planning systems for city logistics service providers. Strategic, tactical and operational planning are distinguished, and functionality required for advanced planning systems in city logistics routing is defined (Sect. 2.4).

2.1 Challenges

The twenty-first century is going to become a century of urbanization, since growing cities facilitate more attractive opportunities for employment, education, cultural, and sport activities (Taniguchi et al. 2008). In 2008, for the first time,

J. F. Ehmke, *Integration of Information and Optimization Models for Routing in City Logistics*, International Series in Operations Research & Management Science 177,
DOI: 10.1007/978-1-4614-3628-7_2,

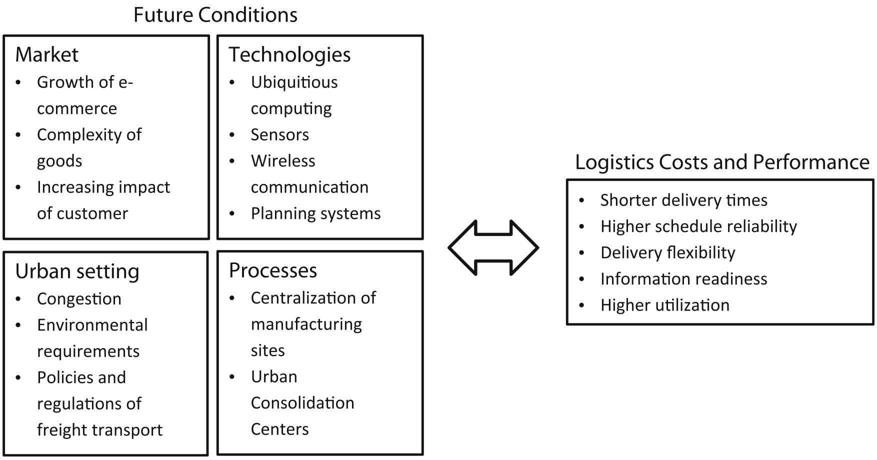


Fig. 2.1 Challenges for logistics service providers (adapted from Hülsmann and Windt 2007)

more people lived in cities than in rural areas worldwide (United Nations Population Fund 2007). An even further migration into cities up to the year 2025 is forecasted, when about 4.5 billion people are expected to live in urban areas, which corresponds to an increase of about 50% compared to the number of people living in cities in 2010 (Statistisches Bundesamt Deutschland 2010). Freight transportation is essential for the development of cities and the supply of their residents. Increasing cities depend on efficient and sustainable freight transportation systems to ensure their attractiveness, economic power, and quality of life.

Logistics service providers operate in the environment of emerging cities. They are exposed to a variety of challenges resulting from the future development of markets, increasing environmental requirements, new technologies, and evolution of complex supply chains. An overview on potential challenges is depicted in Fig. 2.1. Here, future conditions are faced by increasing importance of costs and logistics performance, resulting in, for example, shorter delivery times, higher schedule reliability, and flexibility. Customers expect that the quality of services will rise continuously.

In the following, selected challenges are discussed in more detail. Increasing congestion complicates planning procedures of city logistics service providers, since urban traffic infrastructure is limited, and evolution of supply chains requires the more reliable realization of delivery tours.

2.1.1 Evolution of Supply Chains

City logistics service providers undertake the local distribution of goods which have been consolidated in a shipping terminal. The corresponding logistics network consists of two transportation legs (cf. Fig. 2.2). On the first leg, freight

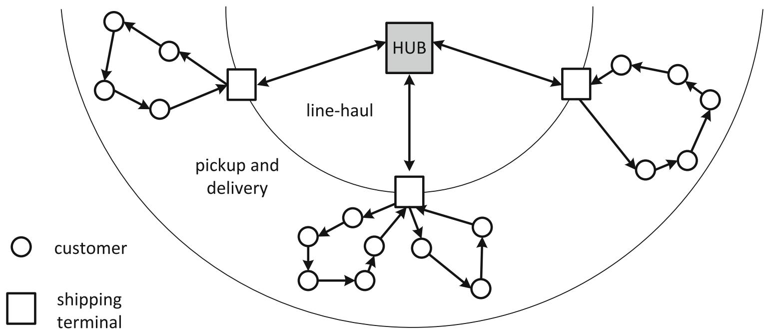


Fig. 2.2 Last-mile delivery in a hub-and-spoke network (adapted from Zäpfel and Wasner 2002)

is transported by large trucks to shipping terminals, where transshipment occurs. Line-haul transports refer to long distance transportation between the sending depots and the hub and between the hub and the receiving depots, respectively. On the second leg, city logistics service providers pick up the goods and deliver them to customers in terms of last-mile delivery. Last-mile delivery involves transportation over short distances with smaller trucks, and is carried out by the receiving depots in their regions. Corresponding hub-and-spoke networks may be operated by a single carrier or by a cooperation of several carriers. For more details on the design of a hub-and-spoke network, see Zäpfel and Wasner (2002).

Increasing importance of city logistics service providers arises from significant changes and developments in the ways in which freight operations are carried out nowadays. Following recent publications by Ruesch and Petz (2008); Crainic et al. (2009a, b), the following trends can be stated:

- Distribution concepts have changed considerably. There is a significant degree of centralization in manufacturing sites, stock keeping points, and retailing, leading to increasing demand for transportation.
- Current production and distribution practices are based on low inventories and timely deliveries. Changing stock keeping patterns and corresponding delivery patterns lead to more frequent, smaller deliveries undertaken by small freight vehicles.
- Supply chain structures have changed substantially, especially for larger companies. Many companies have restructured their supply chain by taking control over large parts and organizing deliveries to their branches themselves.
- Due to ongoing success of e-commerce businesses, distributors as well as retailers are eliminated from the supply chain. Thus, the importance of logistics service providers is increasing, since they undertake the physical distribution of goods, including issuing of a consignment certificate.

In sum, city logistics service providers are embedded in complex supply chains, requiring the fulfillment of demanding customer promises such as tight delivery time windows in the environment of congested urban areas. This emerges for commercial customers (e.g., for timely deliveries in just-in-time production) as well as for consumers (e.g., in online retail or e-commerce activities). The number of complex delivery operations increases parallel to the increasing utilization of road infrastructure, demanding for sophisticated support by planning systems.

2.1.2 Increasing (Freight) Traffic

Increasing traffic within limited city space leads to negative effects in terms of emissions and congestion. Here, city logistics service providers compete against other road users for the scarce traffic space, which cannot be extended unlimitedly. Nonetheless, congestion is usually not considered in city logistics routing. Defiance of varying infrastructure utilization may lead to lower service quality and higher costs of delivery.

Figure 2.3 exemplifies growing infrastructure utilization in Germany, especially within conurbations. Strength and color of roads denote an increase (orange) or decrease (blue) in traffic volumes up to the year 2020. Particularly in the area of conurbations, a huge increase of traffic volumes is expected. This is depicted by a mental orange “C”, reaching from Berlin in the eastern part via Hamburg in the north and the Ruhr district in the west to south Germany. For the area of Munich, a growth of 41% of vehicle miles traveled is expected comparing 2006 with 2020, for example. In the eastern part, which is characterized by rather low industrialization and decreasing population, traffic volumes are expected to decrease.

Corresponding to an increase of overall traffic volumes, the number of freight vehicles moving into and within cities is expected to grow at a steady rate (Crainic et al. 2009b). In European conurbations, more than 80% of today’s road freight trips are of distances below 80 km and can be defined as urban or urban regional transport (Ruesch and Petz 2008). Thus, the generation of efficient and customer-oriented delivery tours will become more and more challenging, forcing city logistics service providers to anticipate congestion in their logistics planning processes.

In the following, city logistics concepts are introduced, focusing on the interaction of urban traffic and transportation systems. This is an important perspective for the improvement of common planning systems.

2.2 Solution Concepts

The need for efficient and environmentally acceptable urban transportation schemes is conjoined by the idea of city logistics. City logistics concepts facilitate integrated solutions for the fundamental dilemma of urban freight transportation: on the one hand, urban freight transportation is fundamental to serve industrial and

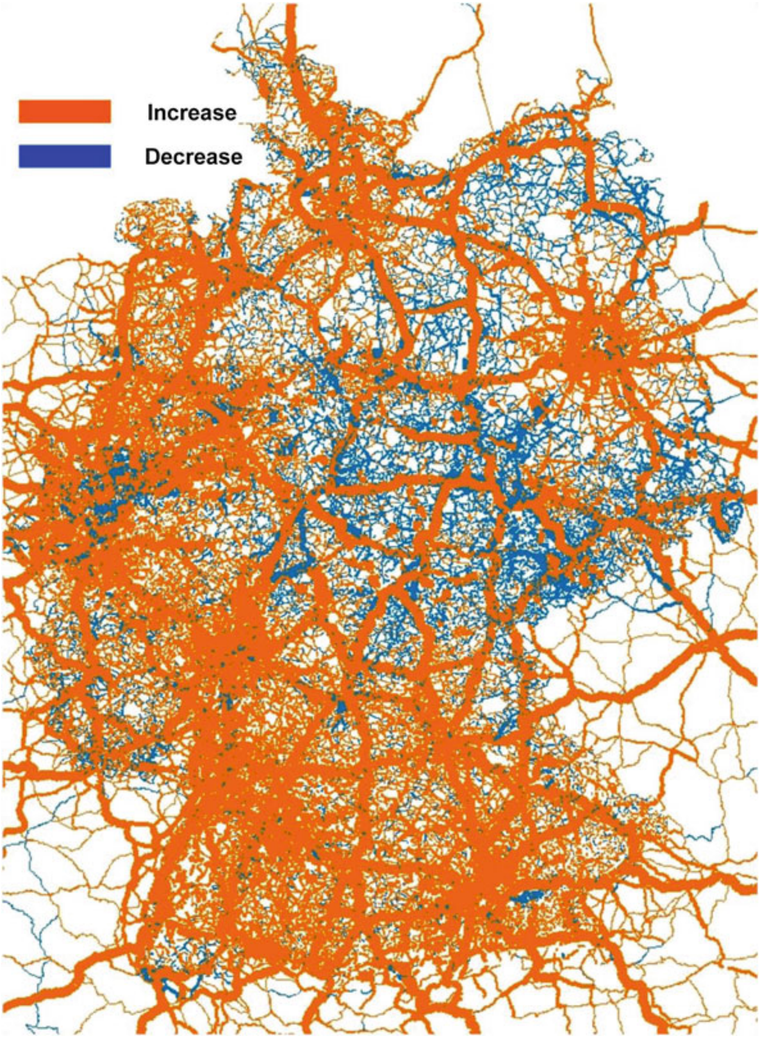


Fig. 2.3 Forecasted traffic volumes in Germany by 2020 (Pape 2006), with kind permission of acatech—Deutsche Akademie der Technikwissenschaften e.V. © 2006 Fraunhofer IRB Verlag, Stuttgart

trade activities in urban areas, ensuring their competitiveness; on the other hand, negative impacts of freight transportation should be limited.

Taniguchi et al. (1999) define city logistics as "the process for totally optimizing the logistics and transport activities by private companies in urban areas while considering the traffic environment, the traffic congestion and energy consumption within the framework of a market economy." Crainic (2008) focuses on the optimization and the utilization of transportation resources that contribute to the realization of profits as well as to environmental nuisances. He hence defines city logistics as the aim of “reducing and controlling the number, dimensions, and characteristics of freight vehicles operating within city limits, improving the efficiency of freight movements, and reducing the number of empty vehicle kilometers.”

City logistics concepts aim at the optimization of urban transportation systems as a whole. They explicitly consider the congestion of traffic network infrastructure, which is neglected by planning systems for routing so far. In the following, core parts of city logistics concepts are introduced and applied to routing for city logistics service providers. First, the alignment of different stakeholders’ goals in urban transportation systems is presented. Then, integrated planning and realization of urban freight transportation by Urban Consolidation Centers (UCC) is discussed. Both aspects are at the heart of many city logistics concepts. They embed the planning activities of city logistics service providers into urban transportation systems as a whole.

2.2.1 Perspective of Different Stakeholders

City logistics concepts aim at the integration of different perspectives of particular stakeholders. According to Taniguchi et al. (2008), the most important stakeholders are as follows:

- *Shippers* send goods to other companies or persons and receive goods from them. They tend to maximize their levels of service in terms of costs and reliability of transport.
- *City logistics service providers* deliver goods to customers. They aim at the minimization of their costs by more efficient pickup and delivery tours, and they are expected to provide a high level of service at low costs. To achieve a high level of service, freight vehicles are loaded inefficiently. They often have to wait near the location of customers when they arrive earlier than the designated time.
- *Residents* are the people who live, work, and shop in the city. They suffer from nuisances resulting from urban freight movements near their residential and retail areas. However, residents also benefit from efficient and reliable delivery.
- *City administrators* attempt to enhance the economic development of the city. They are interested in the reduction of congestion and environmental nuisances as well as in increasing safety of road traffic. To this end, they consider urban transportation systems as a whole to resolve conflicts between the other stakeholders.

Activities of city logistics service providers depend on the interaction of stakeholders presented above. On the one hand, city administrators affect planning procedures by setting complex restrictions for the realization of delivery tours, for example, certain time slots that permit or prohibit the entrance of freight vehicles in pedestrian areas. On the other hand, city administrators collect mass data from the operation of traffic information systems, which are a valuable

Fig. 2.4 Transshipment in UCCs (Allen et al. 2007), with kind permission of ptv AG, Karlsruhe, Germany

source for the more efficient and reliable planning of pickup and delivery tours (cf. Chap. 5). Residents and shippers correspond to customers of city logistics service providers. They expect an economic and reliable delivery service. The interaction of residents and city logistics providers is exemplified by online retail applications (cf. Chap. 3).

2.2.2 Urban Consolidation Centers

A major problem tackled by city logistics initiatives is the inefficient utilization of freight vehicles in urban areas, which contributes significantly to congestion and environmental nuisances such as emissions and noise. A more efficient utilization of freight vehicles can be achieved by consolidation of freight in "city distribution centers" or "urban consolidation centers" (Allen et al. 2007; Crainic et al. 2009b). Increasing efficiency, though, is accompanied by increasing complexity of the corresponding supply chain.

A UCC is a logistics facility that is situated relatively close to the area that it serves, for example, a city center, an entire town, or a specific site (Browne et al. 2005). It canalizes shipments of different companies in terms of an integrated logistics system. UCCs offer storage, sorting, consolidation, and deconsolidation facilities as well as a number of related services such as accounting, legal counsel, and brokerage. An exemplary UCC is depicted in Fig. 2.4, showing large trucks and city freighters as well as corresponding transshipment processes in the UCC of Padova, Italy. Consolidation of deliveries may lead to a decrease of kilometers driven, for example, of approximately 30% for the UCC of Stockholm, Sweden (Neveling 2007). However, increasing efficiency due to transshipment may be counteracted by increasing efforts for more complex cooperation and planning procedures. Transshipment demands for integration of different companies' information systems as well as for the more reliable realization of pickup and delivery tours, considering tight pickup, and delivery time windows.

Crainic et al. (2009b) extend the idea of UCCs by a number of "satellite platforms" that are relatively close to the city center. Freight arrives at an "external zone," where it is consolidated into urban trucks. Each urban truck delivers to one or several satellite platforms. Here, freight is transshipped into environment-friendly vehicles adapted to pickup and delivery in crowded inner city areas. Satellite platforms offer no storage facilities, requiring complex real-time coordination, control, and scheduling of urban trucks and city freighters. In the city of Amsterdam, such a system is in operation ("City Cargo," www.citycargo.nl): freight is consolidated at warehouses on the outskirts of the city and transshipped to specially configured trams. Trams move them to satellite platforms inside the city, where they are picked up by smaller electric freight vehicles.

UCCs are at the heart of many city logistics concepts and initiatives (Janssen and Oldenburger 1991; Ruske 1994; van Duin 1997; Köhler 1997; Köhler 2000; Thompson and Taniguchi 2001; Browne et al. 2005; Taniguchi et al. 2008). Although UCCs are expected to increase the efficiency of urban freight transportation, resulting complexity prevents success of realization in practice. The introduction of a UCC as additional point in the supply chain expects strict compliance of logistics service providers to pickup and delivery time windows, since storage space of UCCs is limited or—in the case of satellite platforms—even not available, and physical properties of goods might require immediate handling. Furthermore, the outsourcing of last-mile delivery to third-party logistics providers may induce suspicion of suppliers, since they lose their direct interface to customers. This is crucial especially for online retail applications, where last-mile delivery is often the only physical and legal activity being perceived by the consumer (see Chap. 3). Though, city logistics initiatives still promote the usage of UCCs, often supported by funding of local city authorities.

2.2.3 City Logistics Initiatives

Urban policies for freight transportation are investigated by a number of public initiatives. The *OECD Working Group on Urban Freight Logistics* focuses on solutions to minimize pollution, noise, and congestion caused by freight transportation, establishing best practices through a review of innovative approaches in OECD cities (OECD 2003). The projects *BESTUFS I and II* summarize best practices of urban freight solutions from a European perspective (www.bestufs.net, Allen et al. 2007). *Trendsetter* describes 54 projects aiming at the improvement of mobility, quality of life, quality of air, reduction of noise, and congestion; five European cities participate in the implementation of innovative city logistics concepts (www.trendsetter-europe.org). The *CIVITAS Initiative* promotes city logistics schemes in terms of sustainable, clean, and efficient urban transportation measures. From 2008 to 2012, 25 European cities take part in five pilot schemes (www.civitas-initiative.org).

The *Institute for City Logistics* (www.citylogistics.org) canalizes research activities on all aspects arising from and around urban freight transportation. Since its foundation in Kyoto, Japan, in 1999, a number of international conferences on city logistics have been organized, resulting in textbooks providing state-of-the-art city logistics concepts and implementations. For a recent overview it is referred to Taniguchi and Thompson (2006); Taniguchi et al. (2008).

City logistics concepts aim at the improvement of urban freight transportation by integrated analysis of transportation infrastructure, transportation resources, and political and economic environment. They induce increasing cooperation between the different stakeholders, resulting in a demand for more reliable delivery services. Especially for UCC operation, advanced planning systems are required which integrate information about customer orders and information about the expected state of urban traffic networks.

2.3 Modeling

City logistics environments may be assessed by comprehensive models describing urban infrastructure, transportation resources, and the impact and behavior of the different stakeholders. In this section, a systems approach on city logistics is presented. The focus is especially on the role of input data for city logistics models. The systems approach is applied to planning procedures of city logistics service providers, enforcing the provision and integration of time-dependent travel times.

Taniguchi et al. (2008) present an overview on the elements of city logistics systems and their relationships. In Fig. 2.5, solid arrows denote the well-known approach of modeling and solution of analytical problems that can be described by models. This comprises the definition of the problem accompanied by objectives and criteria, followed by determination, evaluation, selection, and implementation of a solution. Since city logistics concepts aim at the overall optimization of urban transportation systems, detailed information about transportation resources, constraints, and alternatives are considered for problem solution. Dashed arrows denote the corresponding data collection loop, which enforces the collection and consideration of empirical data in the improvement of urban transportation systems. While this approach is fundamental to the analysis of city logistics problems, it is not acknowledged by planning systems for routing in city logistics, for example, it is not properly exploited by the majority of planning systems up to now.

According to Taniguchi et al. (2008), possible instantiations of the particular components may be as follows:

- Typical city logistics *problems* regard traffic congestion, fleet planning, and fleet management as well as environmental nuisances of urban traffic.
- Corresponding *objectives* are the reduction of operational costs, the increase of efficiency and the reduction of environmental nuisances.

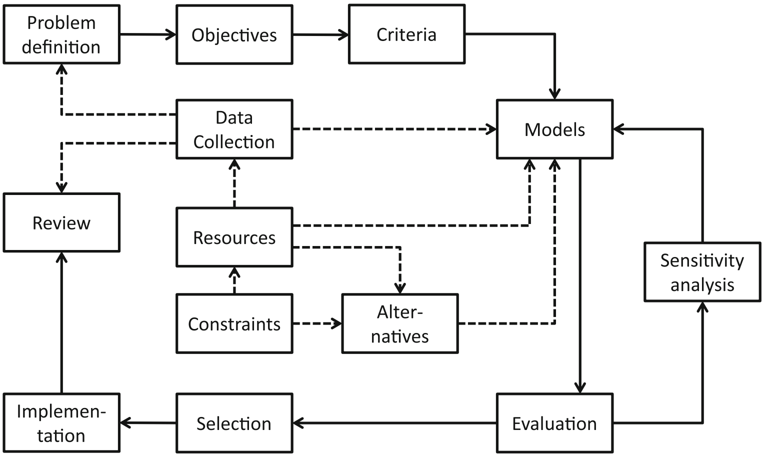


Fig. 2.5 Modeling and analysis of city logistics problems (adapted from Taniguchi et al. 2008)

- *Criteria* may be the number of used freight vehicles, load factors, average speed, or vehicle kilometers traveled.
- *Resources* concern transportation and communication infrastructure, for example, roads, terminals, mobile communications, and freight vehicles.
- *Alternatives* are the options having the potential to solve the problem, for example, route guidance, modern planning systems, or electronic tolling. Financial, legal, social, or political issues limit the range of alternatives that can be considered.
- *Data collection* is required to establish a rational basis for analysis, for example, based on operational data or sensor data such as FCD.
- *Models* provide a simplified representation of the urban freight transportation system. They allow the effects of various changes in the system to be estimated without actually changing the system.
- *Evaluation* involves the methodical comparison of alternatives based on economic, social, financial, energy consumption, and environmental reasons.
- *Sensitivity analysis* investigates the variability of predicted effects of the alternatives with respect to the assumptions made within a model.
- *Selection*, *implementation*, and *review* concern the selection of the best alternative, the implementation of new operating and organizational procedures as well as the investigation if the initial problem has been solved and if the objectives have been attained.

To extend a common planning system for routing in city logistics, the systems approach is applied as follows:

- Efficient and customer-oriented delivery of goods is crucial for the development of viable freight services (*problem definition*). The *objective* is to reduce

operational costs and to improve service quality by taking into account congestion of urban road infrastructure. Reliability of pickup and delivery tours, overall travel times and overall distances, as well as the number of used vehicles define *criteria* for the evaluation of a planning system for routing in city logistics.
- A core point of investigation is the provision and analysis of *information models and optimization models.* Optimization models are formed according to the given criteria of transportation resource optimization. Data collection supplies comprehensive input data for optimization models in terms of information models, which are derived from telematics-based *data collection* by FCD, for example. Information models embed information about typical states of transportation *resources*, for example, the city road network. For routing, this information is represented by extended digital roadmaps, which may lead to more efficient tour *alternatives* due to consideration of time-dependent infrastructure utilization.
- *Evaluation, selection,* and *implementation* of models is done by enhanced planning systems, which consider several variants of planning data sets and optimization procedures. Computational experiments allow for the comparison of efforts for model building and resulting quality of delivery tours with regard to defined criteria.

The presented systems approach aims at the improvement of urban freight transportation by consideration of the different stakeholders' objectives. It provides a big picture of an urban freight transportation system and its numerous components and relationships. In the following, the focus is on the perspective of city logistics service providers and the optimization of pickup and delivery tours within the city logistics framework. Therefore, the integrated collection and provision of detailed traffic data as suggested by the systems approach is examined, which is not properly exploited by common planning systems up to now. The conception of a city logistics planning framework will explicitly focus on the reasonable interaction of data collection and problem solution.

2.4 Planning Systems

In this section, different levels of planning tasks for city logistics service providers are distinguished. Then, a common planning system is extended by advanced routing functionality.

2.4.1 Levels of Planning

Planning systems for logistics service providers may support different levels of planning activities. According to Roy 2001, strategic, tactical, operational, and real-time levels can be distinguished. Individual levels differ by the impact they have on future activities:

- Decisions within the *strategic level* concern a large part of the organization. They have a major financial impact and typically comprise the design of the transportation system, for example, the size and mix of freight vehicles and equipment or the type and mix of transportation services. Corresponding decision problems are poorly structured, complex, and of high risk and uncertainty. They constrain the activities and decisions made at subsequent levels.
- *Tactical planning* deals with short or medium-term activities. Tactical decisions concern the efficient and effective use of transportation infrastructure and the alignment of operations according to strategic objectives. Here, logistics service providers deal with the acquisition and replacement of their equipment, long-term driver to vehicle assignments, and cost and performance analysis. Decisions made at the tactical level limit the activities of operational and real-time management level.
- Decisions of *operational management* concern short term, day-to-day operations. Operational planning is characterized by a short planning horizon and decision problems of detailed problem structure. Here, logistics service providers plan current and next day activities. They should anticipate future developments such as congestion or expected transportation requests. Routing for city logistics service providers is an operational task: a known set of transportation requests is assigned to a given set of transportation resources. Automated planning systems support the corresponding tasks.
- Within *real-time level*, execution of operational decisions is controlled, for example, real-time decisions react on discrepancies between planned and actual state of the transportation system. Activities at the real-time level depend on the decisions made at the higher levels. They are sensitive to the quality and the reliability of operational planning. Deficient anticipation of real-time conditions in operational planning may lead to costly replanning within real-time level.

Since the focus of this work is on routing in city logistics, an advanced planning system for the support of operational planning of city logistics providers is developed. The corresponding architecture is presented in the following.

2.4.2 Architecture of a Planning System

Operational planning of logistics service providers aims at the optimal assignment of transportation requests to transportation resources. Partyka and Hall (2010) give an overview on recent planning systems and their functionality. *Transportation requests* correspond to customer orders, which comprise information about physical properties (e.g., length and weight of a package), geographical properties (e.g., location of pickup and delivery) and logical properties (e.g., customer time windows). *Transportation resources* denote physical properties of the fleet

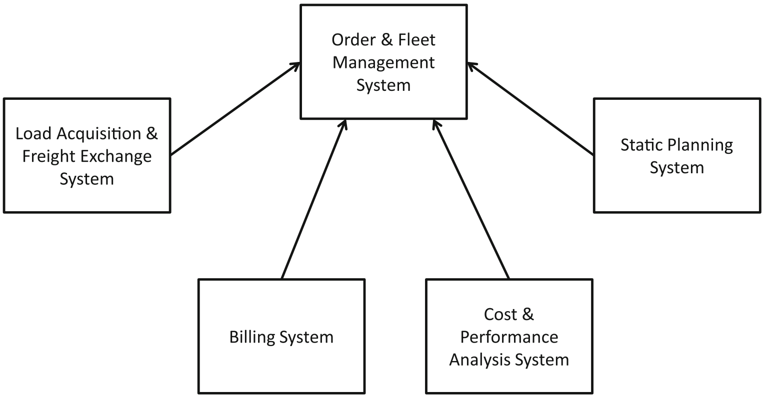


Fig. 2.6 A common planning system for operation of commercial vehicles (Goel 2008)

(e.g., maximum capacity of a freight vehicle), the location of a vehicle (e.g., to which depot it is assigned), and logical properties (e.g., the maximum of drivers' working hours). Goel (2008) presents a comprehensive elaboration of these properties for commercial vehicle operations.

As a result of operational planning, customer orders are assigned to freight vehicles by determination of the most efficient visiting order of customers and routes of vehicles. Corresponding pickup and delivery tours are characterized by a large number of stops within a relatively small geographical area. For instance, warehouse delivery tours in urban areas have a length of about 105 km per day and vehicle in the US (Chatterjee and Cohen 2004).

A common planning system supporting operational planning of a logistics service provider is depicted in Fig. 2.6. The *Order and Fleet Management System* is the central component containing all information about transportation requests and transportation resources. Shippers may enter new transportation requests via an interface to the *Load Acquisition and Freight Exchange System*. The *Billing System* prepares the invoice after order completion. The *Cost and Performance Analysis System* evaluates and aggregates operational data for the support of tactical and strategic decisions. The *Static Planning System* optimally assigns transportation requests to transportation resources by routing functionality, supporting operational planning as described above.

Goel (2008) proposes to extend such a common planning system by a sophisticated real-time telematics component, which automatically adjusts scheduled delivery tours to actual traffic conditions to reduce the planning gap between operational planning and realization. For city logistics routing, this approach is not sufficient, since pressure on city logistics service providers has increased over the past years and time-varying utilization of infrastructure has an enormous impact on the efficient and reliable usage of transportation resources. Expected traffic states should be anticipated as early as in the operational planning phase in order to reduce the planning gap between the operational level and the real-time level. Thus, the focus of this work is on the elaboration of the Static Planning System component by processing and integration of time-dependent travel times.

A recent application example enforcing the planning of more reliable delivery tours is presented in the next chapter.

Chapter 3
Attended Home Delivery

Online shopping has become more and more popular in recent years. Many traditional retailers offer an Internet sales channel nowadays. In this context, city logistics service providers undertake the physical delivery of purchased goods to consumers by means of home deliveries. Meanwhile, their role has become crucial in underlying supply chains, because influence and former functions of distributors diminish.

In the following, characteristics of last-mile deliveries in urban areas are investigated in detail. The last-mile is currently regarded as one of the most expensive, least efficient, and most polluting sections of the entire supply chain (Gevaers et al. 2010), since home deliveries often require the attendance of the consumer, which results in a high rate of delivery failures and empty trips. This leads to negative impacts on costs, efficiency, and environmental performance of delivery tours. Attended home deliveries demand for more sophisticated planning of delivery tours to reliably meet consumer time windows.

Section 3.1 presents a brief overview on business models and reasons for the recent success of online retail. In Sect. 3.2, types of last-mile deliveries are distinguished. Increasing requirements resulting from attended home deliveries are considered in tactical and operational planning of logistics service providers (Sect. 3.3). Subsequently, implications for the conception of an advanced planning system are drawn, derived from the presented city logistics approach and characteristics of attended home deliveries (Sect. 3.4).

3.1 Online Retail

More and more e-commerce businesses compete against each other regarding price and service quality. The choice of products in online shops has diversified since online book shops such as amazon.com have emerged. Even fresh groceries can be

J. F. Ehmke, *Integration of Information and Optimization Models for Routing in City Logistics*, International Series in Operations Research & Management Science 177,
DOI: 10.1007/978-1-4614-3628-7_3,

Fig. 3.1 Delivery of fresh groceries (Allen et al. 2007), with kind permission of ptv AG, Karlsruhe, Germany

ordered online nowadays (cf. Fig. 3.1). Digital outlets of brick and mortar shops complement their distribution channels by online shops. Recent forecasts show that online retail will continue to grow steadily: in the US, online retail sales are expected to increase from $155 billion in 2009 to $250 in 2014. In Western Europe, an even higher yearly compound growth rate of 11% from €68 billion to €114.5 billion is predicted (Schonfeld 2010).

The success of online retail has promoted business models of direct delivery to consumers' homes in recent years ("home shopping"). From a consumer's point of view, home shopping is beneficial due to a greater product choice, better price comparison, the ability to obtain goods not sold locally, or simply convenience (Allen et al. 2007). Home shopping may be useful for bulky and heavy goods, supporting people with mobility problems; it may result in time savings and is available 24/7.

From a logistics point of view, the solution for efficient delivery of home shopping goods is very demanding. On the one hand, efficient and reliable logistics enable economic success of online shops and are thus crucial for the economic success of corresponding business models, since shipping costs are one of the biggest concerns for online customers (Allen et al. 2007). On the other hand, planning of last-mile delivery is very complex. Here, the "not-at-home problem" occurs due to many goods requiring the attendance of the customer during delivery. Furthermore, customers expect delivery as immediate as the online ordering—in case of attended home deliveries, within a tight delivery time window. Thus, logistics service providers have to realize delivery promises by means of tight and reliable delivery time windows.

Examples for online retail applications with regard to home shopping can recently be discovered at e-grocers like Peapod, Caddy–Home, and Le Shop:

- *Peapod* is one of the largest Internet grocers in the US, serving over 11 million households (www.peapod.com). They offer attended home delivery services for more than 10,000 products, including fresh groceries such as cheeses and milk. Delivery occurs by their own fleet; delivery fees are in the range of $6.95–9.95.

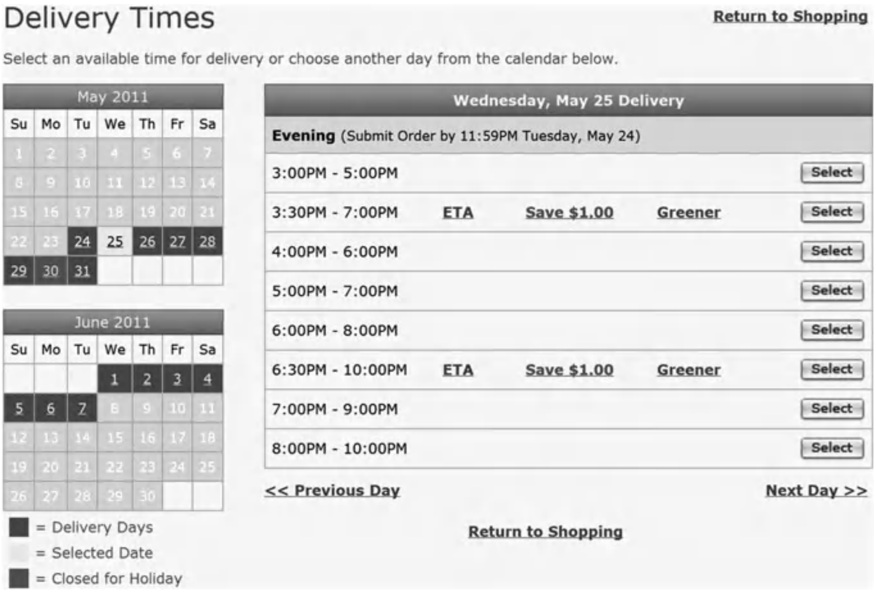


Fig. 3.2 Time slot selection during ordering (www.peapod.com)

For planning of delivery tours, zip code specific characteristics are considered (e.g., population density, Internet penetration, and historical demand data). A discount is allowed if longer customer time slots are chosen in the ordering process (cf. Fig. 3.2). Customers select the desired time slot from a day specific time slot list. Certain time slots may be closed at some point due to capacity considerations (Agatz et al. 2008).

- *Caddy–Home* is the online supermarket of Delhaize, one of Belgium's main supermarket chains. Customers may order products by phone, fax, or via Internet. Delivery is charged at a flat rate of around €7. The day and time of delivery may be chosen by the consumer at the time of order. Caddy–Home immediately checks the feasibility of the desired time slot (www.caddyhome.be).
- *LeShop* is the leading Swiss online supermarket. The company counts 41,500 customers and achieved a turnover of €58 million in 2007. They offer over 9,000 products including fresh fruits and dairy food. There is a flat fee of €8 per delivery, which is conducted by the Swiss postparcel service. If the customer is not at home at the time of delivery, goods are placed according to the customer's specifications, at the front door, with a particular neighbor, or at the post office where they can be collected later (www.leshop.ch).

A comprehensive overview on further businesses dealing with last-mile delivery is given by Allen et al. (2007). Based on their empirical analysis, the following success factors for last-mile solutions can be summarized:

- The provision of a fast, reliable, and flexible delivery service at a reasonable price is the key to the success of online retail applications. Customer time windows play an important role for the service quality of such businesses.
- Attended home delivery will continue to dominate until technical developments allow greater use of unattended delivery systems (e.g., for food deliveries).
- To offer low cost services, retailers and city logistics providers must seek to reduce their peak hour throughputs by means of innovative service pricing, shared operations, and technically acceptable unattended delivery systems.

In sum, crossing the last-mile leads to an enormous increase in convenience for consumers, but also creates a huge challenge for city logistics service providers. Last-mile delivery enables the success of online businesses, especially with regard to perceived service quality and resulting costs of delivery. However, fees are often far from reflecting actual costs of delivery. Following the Internet hype, many e-grocers could not achieve customer volumes needed to make their distribution models viable (e.g., for the failure of Webvan, see Lunce et al. 2006). Suppliers nowadays follow different strategies to ensure a viable service, for example, by dynamic delivery fees or advanced planning systems that are able to feedback on expected realization costs in order to evaluate the potential costs of a new order.

The most challenging type of delivery—attended home delivery—is discussed in the next section in more detail.

3.2 Types of Last-Mile Delivery

Online retail applications require a cost-efficient and consumer-oriented variant of last-mile delivery. The last-mile may be defined as the final leg in a business to consumer delivery service, where the “consignment is delivered to the recipient, either at the recipient’s home or at a collection point” (Gevaers et al. 2010). From an e-commerce point of view, last-mile or home deliveries are the logistics element of the “fulfillment process within consumer e-commerce transactions, other remote purchases from mail order, direct selling and television shopping companies, and deliveries from retail outlets” (Allen et al. 2007). Fulfillment comprises the physical distribution of goods as well as the legal confirmation of the delivery. Most deliveries are of parcels and small packages (e.g., books, CDs, clothing), large items (e.g., furniture, home, or other large electrical appliances), and food.

The starting point of goods is a retailer’s warehouse or a fulfillment center, followed by different supply chain options by which the goods reach the consumer (cf. Fig. 3.3). Correspondingly to hub-and-spoke networks introduced in Sect. 2.1, goods are moved to a shipping terminal, a regional distribution center or/and a

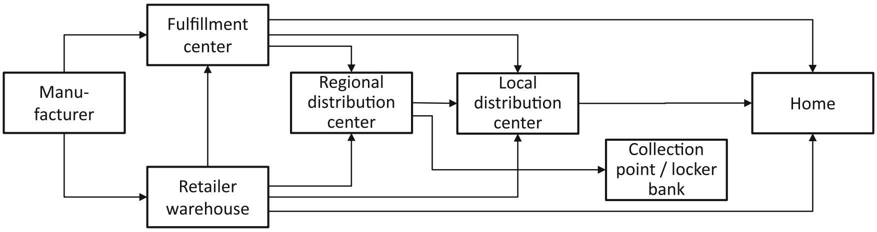


Fig. 3.3 Common architectures of last-mile delivery (adapted from Allen et al. 2007)

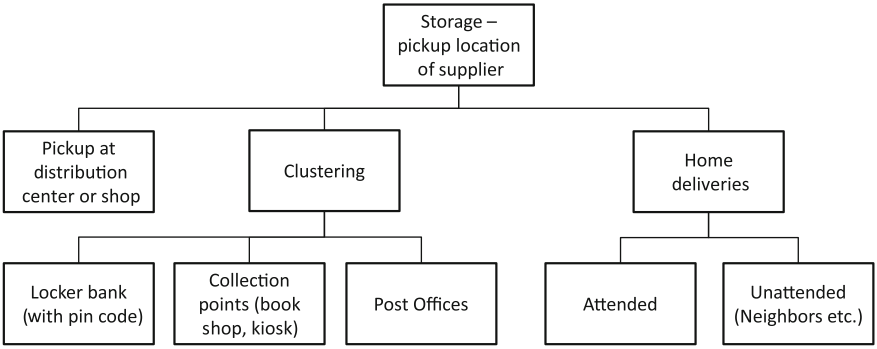


Fig. 3.4 Types of last-mile delivery (adapted from Gevaers et al. 2010)

local distribution center. Transshipment often takes place close to the city border, where goods are picked up by city freighters (in a local distribution center or UCC, cf. Sect. 2.2). A consolidated delivery tour starts either from the fulfillment center, from the retailer's warehouse, or from a shipping terminal. Either own vehicles or third-party vehicles of city logistics service providers are used. Deliveries may be made to the consumer's home, to reception/delivery boxes, collection points, locker banks, or alternative places defined by the consumer.

Gevaers et al. (2010) establish a typology of last-mile delivery options (cf. Fig. 3.4). They distinguish the place of delivery, which can be a pickup location (e.g., a distribution center or a shop), a facility for the consolidation of consignments or the consumer's home. For the consolidation of consignments, a wide range of options exist, for example, conventional post offices, reception boxes accessible by pin codes, or third-party stores such as petrol stations and kiosks. For home deliveries, the crucial question is whether the consumer has to be present at time of delivery or not (attended/unattended home delivery).

Last-mile deliveries may be realized by different variants, depending on the type of goods delivered, available technology, and type of delivery service. In Table 3.1, characteristics of attended and unattended delivery concepts are summarized. There is a tradeoff between consumer convenience and efficiency of delivery tours: non-timed home deliveries and deliveries to pickup points allow for

Table 3.1 Comparison of unattended and attended delivery systems (adapted from Allen et al. 2007)

Delivery point	Attended delivery	Reception box/delivery box	Controlled access system	Locker bank	Collection point
Who covers the last mile?	City logistics service provider	City logistics service provider	City logistics service provider	Consumer	Consumer
Consumer at home problem	Yes	No	No	No	No
Types of products	Any	Packages, groceries	Packages, groceries	Packages, groceries	Packages
Failed deliveries	High	Virtually none	Virtually none	Virtually none	Virtually none
Delivery window	Fixed delivery hours	Service provider operating hours	Service provider operating hours	Service provider operating hours	Collection point opening times
Collection times	Not appropriate	24 h	24 h	24 h	Collection point opening times
Retrieval time for customers	None	Very short	Very short	Short–Long	Short–Long
Drop-off time for service provider	Long	Short	Short	Very short	Very short
Initial investment	Low	High/Medium	Medium	Medium	Low/Medium
Delivery Costs	High	Low	Low	Lowest	Lowest
Operational problems	High number of failed deliveries	Large number of boxes needed	Safety. Need for suitable location	Customer has to travel	Customer has to travel

a higher degree of freedom in planning of delivery tours, but are less consumer friendly than deliveries on appointment. This is contrasted by attended home delivery services, where suppliers and consumers mutually agree on a delivery window. Delivery points and dedicated pickup and delivery centers allow for the bundling of goods.

The most substantial last-mile issues occur in attended home delivery, for example, when the consumer is required for the signature of reception or for the immediate storage of goods. In case of groceries, attended home delivery is the most common type of delivery (Allen et al. 2007). If no specific time window of delivery has been arranged, the failure rate due to absence of customers will be inevitably high (Gevaers et al. 2010). Thus, attended home deliveries usually take place on a prearranged day and within a given customer time window. In some cases, consumers make an explicit payment for the delivery; in other cases, a delivery charge is only applied below an agreed value of goods.

In sum, the solution for efficient and consumer-oriented last-mile delivery is challenging, especially in the case of attended home deliveries. The success of online retailers depends on their ability to build on efficient last-mile delivery solutions offered by city logistics service providers. In the following, the design of delivery time windows and the planning of delivery tours are discussed.

3.3 Customer Time Windows

The efficiency of delivery tours is mainly determined by the number and configuration of customer time windows (Agatz et al. 2011). Customer time windows do not only reduce the proportion of time available per service area, but may also decrease the density of customers per tour (Figliozzi 2007). The design of customer time windows occurs within tactical planning, whereas the realization of customer time windows is prepared by operational planning.

3.3.1 Tactical Planning

The design of customer time windows is a task of *tactical planning*. Typical decisions comprise the length, possible overlap, and the overall number of offered customer time windows as well as the general concept of delivery fees. In tactical planning, there is a conflict of interests between customers, who expect tight, reliable delivery time windows, and city logistics service providers, who have to realize delivery time windows.

Agatz et al. (2011) discuss the tactical planning of “time slot schedule design,” which assigns specific time slots to spatial areas defined by zip codes corresponding to service requirements. Interaction between tactical and operational planning is exemplified by “time slot schedule management,” for example, when to “close” a time window due to capacity or routing restrictions. Campbell and Savelsbergh (2005) focus on the interaction of order promise and order delivery. They introduce several heuristics for the determination whether a delivery request can be feasibly accommodated in any of the time slots, based on the set of already accepted customers and still expected customers.

Punakivi and Saranen (2001) report that completely flexible, unattended deliveries reduce costs by up to a third relative to attended deliveries based on 2-hour time windows. Campbell and Savelsbergh (2005) found that expanding a 1-hour delivery time window to 2 hours can increase profits by more than 6% and an additional 5% if further expanded to 3 hours. Campbell and Savelsbergh (2006) encourage customers to accept wider delivery time windows that ensure more freedom for operational planning. If customers select favorable time windows (from a logistics perspective), not only total distances of delivery tours decrease, but a more efficient use of transportation resources may increase the number of orders that can be accepted. They show that incentive schemes can substantially reduce delivery costs and thus enhance profits.

3.3.2 Operational Planning

The main challenge of *operational planning* is to produce delivery tours that efficiently and reliably meet customers’ requirements based on parameters set in tactical planning. Here, customer orders are assigned to freight vehicles, resulting in a cost efficient order of deliveries. Common planning systems, however, are not well suited for the requirements of attended home delivery. Eglese et al. (2006) report on a survey that reveals ongoing dissatisfaction with planning systems: dispatchers and drivers declare that there is “significant inaccuracy due to the credibility of the forecasted travel times for individual vehicle trips.” The majority of planning systems refers to average travel times which are assumed to be constant for all planning horizons, contrasting well-known variation of travel times due to congestion in urban areas. Planning systems ignoring these variations produce inefficient and suboptimal delivery tours. Poorly designed delivery tours may lead freight vehicles into congested trunk roads, which does not only lead to increasing delivery costs, but also exacerbates negative impacts of urban freight traffic (Figliozzi 2009).

As a result, logistics service providers assign fewer customers per vehicle to cope with travel time uncertainties due to traffic congestion. Alternatively, “buffer time” is incorporated in delivery tours. This tends to increase the likelihood of on time delivery. However, buffer times also lead to decreasing measures of productivity, for example, increasing driver and equipment idle time, increasing costs, and a decreasing number of kilometers traveled per hour (Kim et al. 2005). A lack of travel time observations may result in overestimations of the actual travel time, which in turn may lead to unnecessary buffers in delivery tours, resulting in an increase of vehicle mileage and negative impacts on city traffic (Figliozzi et al. 2007).

Consequently, the expected state of the city road network and its variability cannot be ignored in operational planning. The collection and processing of congestion data is usually a difficult, expensive, and cumbersome process, and the sheer size of the road network often even precludes the collection of a sufficient number of travel time measurements. Nonetheless, more realistic travel time anticipations would increase the reliability of schedules, leading freight vehicles into rather uncongested parts of the city road network.

3.4 Implications

In this section, the approach of this work is clarified based on the challenges, solution concepts, and application examples presented above. First, requirements for an advanced planning system are summed up. Then, related work is presented and delimited from the integrated approach pursued in this work.

City logistics service providers operate in an environment of congested urban areas. They are exposed to a variety of challenges, especially with respect to increasing complexity of supply chains. Here, distributors and retailers are more and more eliminated from the supply chain. Solution concepts for optimization of urban freight transportation are provided by city logistics. City logistics concepts highlight the integrated collection and allocation of detailed traffic data to optimize urban freight transportation. However, common planning systems are not capable of considering such data, which could improve operational planning of delivery tours and decrease the planning gap between estimated and actual travel times. Attended home delivery exemplifies a time-critical application of online retail, where the determination of reliable delivery tours and the consideration of customer time windows are crucial for the viability of the business model. In sum, advanced planning systems for operational planning of delivery tours are required. Following the idea of city logistics, empirical data on urban transportation systems should be considered by automated planning procedures.

In the literature, only a few authors have come up with the integrated improvement of planning systems by more suitable input data and corresponding alignment of planning procedures:

- For city logistics, Fleischmann et al. (2004) design a traffic information system. Speed data is collected in a field test with stationary measurement facilities and specially equipped vehicles in the metropolitan area of Berlin, Germany. The data were then aggregated and utilized by adapted planning procedures. The data collection methods have become obsolete due to technological progress. FCD nowadays produces travel times at low costs citywide.
- Kim et al. (2005) conduct computational experiments for a small traffic network consisting of 10 OD pairs in Southeast Michigan. They use historical as well as real-time traffic data. To evaluate the benefits of historical and real-time traffic information, different scenarios are analyzed for several shipping time slots.

Usage of historical travel times leads to savings of up to 8% as well as to a reduction of truck usage time of up to 12% for certain time slots. In this work, a citywide approach is presented, considering challenges of routing in urban road networks.

- Eglese et al. (2006) refer to FCD for time-dependent routing in a supra-regional road network in the UK. The FCD originate from a communication network consisting of trucks and coaches. Data were transmitted via text messages and stored as a "road timetable" in a central database. In city logistics, text messages are not appropriate for data collection due to high communication costs. The complexity of urban traffic networks demands for a more thorough data collection method.
- Van Woensel et al. (2008) generate travel times based on queuing theory, alleviating methodological obstacles in the determination of travel times. Contrasting their approach, this work aims at processing of telematics-based travel time data, which allows for a more efficient alignment of information models and optimization models in the environment of city logistics routing.
- Maden et al. (2010) present a case study for the distribution of goods by an electrical goods wholesaler. Their planning system is used to schedule a fleet of vehicles operating in the South West of the United Kingdom. They report on savings in CO_2 emissions of approximately 7% compared to planning methods based on constant speeds. Similar to Eglese et al. (2006), they rely on travel time data from well-chosen vehicles traveling on freeways mostly.

In this work, drawbacks of recent approaches are counteracted by integration of different perspectives. An integrated approach is crucial to improve common planning systems for routing in city logistics. In particular, economic, technological, and methodological perspectives are investigated. For each perspective, related work is introduced and extended in the following chapters:

- Increasing customer requirements and cost pressure must be counteracted by advanced planning systems for routing in city logistics (*economic perspective*). In the following, a planning framework is presented and exemplarily instantiated based on empirical traffic data of the Stuttgart area in Germany. Computational experiments allow for the evaluation of planning data sets with respect to reliability and cost savings.
- Advanced planning systems require information on the typical behavior of urban transportation systems. It is referred to state-of-the-art technology of data collection by FCD (*technological perspective*). Especially for data collection in urban areas, FCD establishes a sufficient number of travel time measurements at low costs, though the transformation of such data into useful information as well as their proper exploitation is complex.
- Handling of operational data and incorporation of mass data in planning procedures requires methodological support (*methodological perspective*). Common static information models as well as static optimization procedures are not suited for the generation of reliable delivery tours, because they ignore information about time-dependent appearance of congestion. Compared to the

actual realization, the planning gap between scheduled and realized tours can be reduced substantially when taking into account time-dependent travel times. This may avoid violation of delivery time windows and reduce the likelihood of driver overtime, though time-dependent optimization models have seldom been addressed because they are harder to model and to solve (Ichoua et al. 2003).

In sum, the provision of time-dependent planning data as well as the effective alignment of information models and optimization models is usually neglected. Following Crainic et al. (2009a), advanced planning systems are required, which aim at processing this information and integrating them into transportation plans to achieve a more timely operation, efficient allocation utilization of freight vehicles, and satisfaction of customer requests.

Part II
Information Models

Chapter 4
Knowledge Discovery and Data Mining

Nowadays, it is possible to collect detailed data about the current state and the operations of systems such as transportation systems. Due to technological advancements, we may collect and store enormous amounts of operational data at low costs. These data are usually not properly exploited, because the derivation of relevant information for the improvement of planning systems is challenging. However, planning systems rely on such information describing the typical behavior of a system, which can be derived from aggregates of operational data. Based on typical system behavior, future operations can be planned.

In this part, methodology for the aggregation and analysis of operational data as input for planning systems is investigated. It is related to the field of DM, which faces challenges resulting from collection and storage of mass data with respect to exploration and exploitation of useful information. DM considers operational data for the provision of information models. Information models describe the typical behavior of a system based on a comprehensive number of operational data records.

In Sect. 4.1, the Knowledge Discovery Process is introduced, which serves as a framework for the analysis of mass data by DM techniques. DM represents the core task of the Knowledge Discovery Process. It is inherently multi-disciplinary and comprises methods from the areas of statistics, visualization, machine learning, pattern detection, databases, and artificial intelligence. In Sects. 4.2 and 4.3, cluster analysis and EDA are discussed, exemplifying the wide range of techniques in the field of DM for traffic data analysis.

4.1 Knowledge Discovery Process

Since the 1990s, there has been an extreme increase in collection and storage of operational data. Recent technology is able to collect and store huge amounts of empirical data at low costs provided by, automatic sensors, telematics-based

J. F. Ehmke, *Integration of Information and Optimization Models for Routing in City Logistics*, International Series in Operations Research & Management Science 177,
DOI: 10.1007/978-1-4614-3628-7_4,

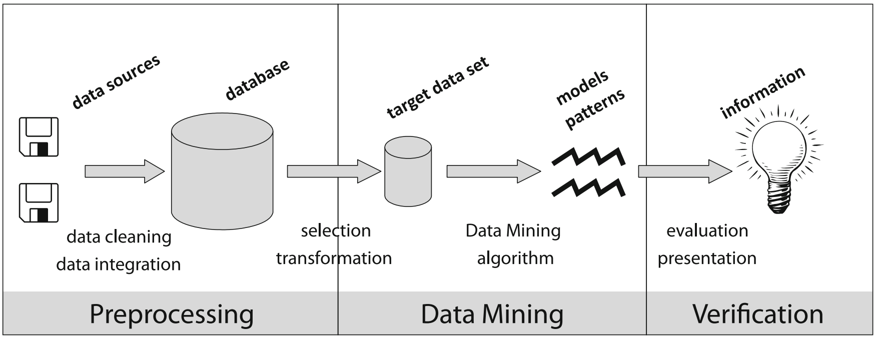


Fig. 4.1 Knowledge Discovery Process (adapted from Han and Kamber 2006)

data collection, or transactional information systems, for example. However, well-known methods from the field of statistics are not sufficient for the exploitation of mass data in order to derive relevant information as input for planning systems. Advanced methodology is required that transforms data into information. According to Berthold et al. (2010), *data* refer to single instances (e.g., single objects, events, points in time), contrasting *information*, which refers to classes of instances (e.g., sets of objects, events, functions of points in time). Data are often easy to collect and available to a large extent, but do not allow us to make predictions or forecasts, whereas information is often difficult to obtain, but is needed in order to anticipate future system behavior. To this end, DM has arisen as a new field of methodology aiming at structuring of mass data in order to derive information from huge databases (Fayyad et al. 1996), i.e., transforming data into information. DM deals with the revelation of (unsuspected) relationships by identifying structures in large data sets.

Several definitions of DM exist, depending on the focus on this multi-disciplinary area. Han and Kamber (2006) follow a rather wide perspective and define DM as "the process of discovering interesting knowledge from large amounts of data stored either in databases, data warehouses, or other information repositories". Fayyad et al. (1996) refer to DM as "knowledge discovery in databases" (KDD), focusing on the appropriate integration of tools for data collection, data analysis and data evaluation. The goal of the KDD is the identification of valid, novel, potentially useful and understandable information models or information patterns.

The multi-disciplinary perspective of DM makes the KDD a non-trivial, partially interactive process, which integrates methods from the fields of statistics, visualization, machine learning, pattern detection, databases, and artificial intelligence. KDD comprises the extraction of information models from operational data in terms of a multi-step procedure. In particular, a preprocessing step is followed by the core DM activity, and a verification step (cf. Fig. 4.1). The term *intelligent data analysis* has recently been used synonymously, underlining that every data analysis project is individual and potentially challenging (Berthold and Hand 2009).

The single steps of KDD are as follows:

- *Preprocessing* integrates data sources containing operational data. Data may be imported from conventional text files or several databases into a central database ("data integration"). Missing values and outliers are filtered ("data cleaning"). Relevant data records are selected and transformed for subsequent analysis by DM (cf. Sect. 4.1.1).
- The second step refers to data analysis concerning a specific problem or hypothesis in terms of the core *Data Mining* step. Hand et al. (2001) elaborate on the latter by specifying a framework for core DM algorithms (cf. Sect. 4.1.2). Underlying data have usually not been collected for the purpose of this analysis, but have arisen as a byproduct from the operation of a system. This distinguishes analysis by DM from well-known statistical analysis.
- Resulting information models and patterns are checked in the verification step by sophisticated presentation and evaluation techniques (cf. Sect. 4.1.3).

An alternative organization of data analysis is pursued by the cross industry standard process for data mining (CRISP-DM, Chapman et al. 2000), which consists of six phases ranging from project and data understanding via data preparation and modeling to its evaluation and deployment. CRISP has been developed by a consortium of large companies such as NCR, Daimler, and SPSS, and is widely used for data analysis in practice.

In the following, the single steps of the KDD according to Fayyad et al. (1996) as well as corresponding methods are introduced from a scientific point of view. In particular, approaches for the analysis of operational data arising from traffic and transportation systems are introduced. Where techniques are not discussed in detail, references are provided.

4.1.1 Preprocessing

The first step of the KDD refers to *preprocessing*. Here, operational data are examined and prepared for subsequent analysis by DM. The goal of preprocessing is to generate a valid target data set by handling incomplete, erroneous data records. To this end, well-known methods of statistical analysis and visualization come into play. A common approach is to analyze data sets by descriptive data analysis in terms of mean, median, mode, range, quartiles, variation, standard deviation, and graphic displays such as histograms and boxplots. Thus, abnormalities within the data set can be investigated, supporting understanding of data.

The impact of preprocessing on subsequent analysis by DM and corresponding efforts are often underestimated. The temporal effort, for example, may amount up to 30% of the KDD process (Berthold 2006). This is due to automatically collected data sets regularly suffering from incomplete, inconsistent, and biased data records. A sensor may fail or produce biased measurements, or manual data collection may result in deficient data records, for example. Besides, conversion and import of data records from several data sources may lead to inconsistent target data sets.

4.1.1.1 Data Cleaning

Data cleaning, data integration, data transformation, and data reduction ensure properly preparation of the target data set. *Data cleaning* examines data quality in terms of how well the data fit to their intended use (Berthold et al. 2010). It aims at the detection of missing values, the smoothing of noisy data, the identification and removal of outliers, and the detection of inconsistencies:

- *Missing values* may result from a broken sensor, for example. They may be ignored, filled in manually, replaced by a global constant, or estimated by a meaningful value such as the mean of an attribute (Han and Kamber 2006).
- *Noisy data* is caused by random error or inherent variance of a measured variable. For data cleaning, values can be "smoothed" by aggregation in different kinds of ways. *Binning* calculates the mean of a data value's "neighborhood". Then, affected data values are replaced by the mean. *Regression* fits data records to a function, and *clustering* organizes data records in groups (cf. Sect. 4.2).
- *Outliers* are data records that do not comply with the general behavior or model of the whole data set, i.e., they are different from all or most of the other data records (Berthold et al. 2010). The main challenge is to differentiate between outliers and significant system behavior, since outliers can be caused by measurement errors or inherent data variability. A large number of (statistical) methods for the detection of outliers exist. For single attributes, statistical tests may be taken into account. For multidimensional data, clustering of data objects may identify data records that cannot be assigned reasonably to any cluster (Rehm et al. 2006; Santos-Pereira and Pires 2002).

More methods of data cleaning can be found in Berthold et al. (2010); Tan et al. (2009).

4.1.1.2 Data Integration

Data integration refers to the conceptual and physical integration of several sources of raw data. The focal point is the alignment of data schemas for tables from different data sources, since attributes representing similar content could be denominated differently or could have been measured differently (e.g., in meters and centimeters). Data integration concepts ensure data quality in terms of syntactic and semantic accuracy as well as efficient loading of large data sources. Redundancy of different attributes is revealed by correlation analysis. Given two attributes, correlation analysis depicts how strongly one attribute implies the other. For numerical attributes, the correlation coefficient is a well-known figure; for discrete data objects, a correlation between two attributes can be discovered by a Chi-square test.

The allocation of a target data set may require *transformation* of data by generalization and normalization procedures (Han and Kamber 2006). For instance, the analysis of time series expects the generalization of the corresponding data records in terms of averages. In case of differently scaled attributes, normalization of a data set is required. Therefore, the range of values of an attribute is mapped to a new range such as [0, 1]. Normalization often occurs in terms of min–max normalization, where $\min_a$ and $\max_a$ refer to minimum and maximum values of an attribute a. Min–max normalization maps a value w by linear transformation to a new value w' in the new range $[\min_{a'}, \max_{a'}]$. Calculation of w' occurs as follows:

$$w' = \frac{w - \min_a}{\max_a - \min_a}(\max_{a'} - \min_{a'}) + \min_{a'}$$

Hence, relations between original values are obtained. Further examples for normalization procedures are the normalization by the average or by decimal scaling.

Preprocessing may also refer to the *reduction* of a data set if subsequent processing cannot be achieved within reasonable time or if processing is limited by available RAM. To this end, data records are aggregated, their dimensions are reduced with respect to most relevant attributes, and values are discretized. Subsequent analysis may then refer to a compressed target data set.

In sum, preprocessing generates a valid target data set that supports subsequent analysis by DM. The single steps depend on the goal, structure, and requirements of the specific DM technique.

4.1.2 Data Mining

DM can be understood as the core step of the KDD. It is concerned with secondary analysis of large amounts of operational data, i.e., the analysis of mass data that has not intentionally been collected for this purpose. Hand et al. (2001) summarize DM as "the analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner". Both the aspect of secondary analysis and the sheer amount of data records distinguish the field from common statistics.

DM usually copes with data arising as a byproduct of either operational systems or simulation of such systems. It aims at generation of information about a system from mass data by means of information models and information patterns. In this section, different tasks of DM are distinguished based on a typology by Hand et al. (2001). For a more comprehensive overview on DM, it is referred to recent textbooks by Berthold et al. (2010); Tan et al. (2009); Han and Kamber (2006).

Meisel and Mattfeld (2010) sum up different process models in order to describe DM functionality. In Fig. 4.2, the relationship between preprocessing and analysis of a system by DM techniques is depicted. Hypotheses about system structure define the structure of the desired information. They may be derived from general knowledge about a system by discussions with system experts, for example.

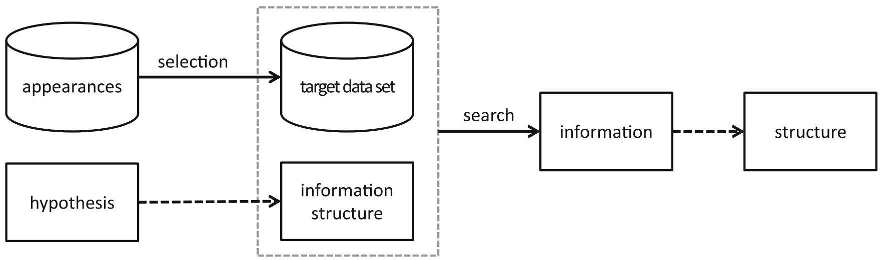


Fig. 4.2 Data Mining process model (adapted from Meisel and Mattfeld 2010)

As described above, the preprocessing step refers to the extraction of a target data set from mass data ("appearances"). The core task of the DM step is then to derive and model information about system structure by a search procedure. The search procedure determines the best instance representing "system relations according to given measures for the quality of fit".

Hand et al. (2001) define two types of information about system structure: information models and information patterns. Whereas patterns represent exceptional system behavior, information models allow for global statements on system behavior. Information models and information patterns result from the analysis of underlying data in terms of five different DM tasks: Descriptive Modeling, Predictive Modeling, Exploratory Data Analysis, Discovery of Patterns, and Retrieval by Content.

Descriptive Modeling aims at representing data in a concise and summarizing manner in order to reveal general properties of a huge data set, i.e., an aggregated view on the system is provided in terms of typical system behavior. A descriptive model represents the main features of the data set. Hand et al. (2001) present an overview on corresponding DM techniques, describing data by probability distributions and densities, or highlighting relevant features of data sets by cluster analysis, for example. Cluster analysis is the most popular among the techniques for this DM task and is discussed in Sect. 4.2 in detail.

Predictive Modeling contrasts descriptive modeling by forecasting of system behavior. Classification models predict categorical attributes, whereas prediction models focus on continuously valued functions. A well-known method is the classification by decision tree induction. A decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node holds a class label. Given a data record for which the associated class label is unknown, the attribute values are tested against the decision tree. Decision trees can be derived by algorithms such as ID3 (Quinlan 1986), C4.5 (Quinlan 1993) and CART (Breiman et al. 1983). A variety of further classification approaches exist, for example, Bayesian Classification, Rule-Based Classification, Classification by Backpropagation, or Support Vector Machines (Berthold et al. 2010). Prediction models result from linear or nonlinear regression. They depict the relationship between one or more independent variables and a dependent variable. Given a set of independent variables, the associated value of the dependent variable is predicted.

Exploratory data analysis (EDA) allows for a concise visualization of mass data, probably without any clear ideas of what we are looking for. There are many effective graphical display methods for low-dimensional data. However, EDA explicitly deals with the visualization of high-dimensional data. It can be used as a single DM technique as well as a complement for previous analysis in order to visualize its results, validate hypotheses, or to produce ideas for further analysis. Details on EDA are discussed in Sect. 4.3.

Whereas the tasks reported above are concerned with model building, *Discovering Patterns and Rules* aims at the exploration of specific phenomena, i.e., the detection of recurring patterns in system behavior. This task allows for the analysis of inference of attribute values for certain system appearances. The challenge is to decide what constitutes truly unusual behavior in the context of system inherent variability. Well-known approaches are association rule algorithms or rule induction methods (Hand et al. 2001).

Retrieval by Content facilitates the detection of similar patterns given a user-defined set of patterns. This task refers to search in text and image data sets. A pattern may be a set of keywords, and the user may wish to find relevant documents within a large set of documents, for example.

For each of the DM tasks described above, a variety of algorithms exists. Hand et al. (2001) distinguish DM algorithms by four basic components. The underlying structure or functional form that we seek from the data is defined by *model or pattern structure*. A *score function* judges the quality of a model. It quantifies how well a model or parameter structure fits a given data set. The *optimization* or *search method* determines the structure and the parameter values that lead to a minimum (or maximum) value of the score function. The task of finding the "best" values of parameters for model calibration is typically cast as an optimization problem and often solved by heuristic search techniques. *Data management strategy* is concerned with handling data access efficiently during the search or optimization process in terms of how data are technically stored and indexed in very large databases.

In sum, DM aims at structuring of large operational data sets for subsequent interpretation and application. To this end, mass data are related in a novel, sometimes unexpected, useful way. Resulting relationships are represented by information models and information patterns.

4.1.3 Verification

The verification step deals with the presentation, verification, and exemplary application of information models and information patterns resulting from the DM step. Information models and information patterns are investigated regarding validity within the context of their purpose. According to Meisel and Mattfeld

(2010), it must be determined to what extent the derived model or pattern represents the structure of the underlying system.

For verification, standard methods rarely exist. In general, results must be processed in an understandable way in order to support evaluation and further analysis or usage in subsequent applications. Validation depends on subjective and objective criteria. On the one hand, verification may occur by comparison of attribute values derived from the information model or from information patterns with attribute values derived from observed system appearances. On the other hand, expert interviews may help judging relevance and validity of results. For temporal and spatial interpretation of operational data from traffic and transportation systems, analysis by Geographical Information Systems (GIS) is useful (cf. Sect. 4.3).

KDD comprises a powerful, non-trivial process supporting the exploitation of operational data by determination of relevant characteristics of the underlying data set. In the following section, cluster analysis is highlighted due to its importance for summarizing characteristics of mass data.

4.2 Cluster Analysis

Cluster analysis or *clustering* is “the process of grouping data into classes or cluster so that objects within a cluster have high similarity in comparison to one another, but are very dissimilar to objects in other clusters” (Han and Kamber 2006). Cluster analysis belongs to the group of descriptive modeling approaches. In contrast to classification, which relies on predefined classes for the assignment of data objects, clustering aims at the identification of “natural” groups in a data set. From the perspective of machine learning, clustering can be assigned to unsupervised learning (Erman et al. 2006), since data records are assigned to clusters without exact knowledge on the design and number of clusters. Cluster analysis may also be used to analyze the distribution of a data set. Results of clustering should be visualized in a user-interpretable way. This is important for the evaluation of clusterings, since every clustering facilitates a (sometimes futile) result.

The prerequisite for clustering is a system-specific hypothesis describing the assumed system structure. Based on the hypothesis, tools and structure of cluster analysis as well as techniques for the evaluation of results are defined. The process of a typical cluster analysis is depicted in Fig. 4.3.

Each clustering begins with the *selection of data objects and variables*. Based on the underlying hypothesis, variables providing the desired information are identified and selected. A careful choice limits the number and dimensions of variables, keeping computational efforts low. At this point, methods from pre-processing come into play (cf. Sect. 4.1.1). As a result, a target data set is loaded from the database in an appropriate form, being transformed or normalized as described above in order to allow for an efficient and meaningful clustering.

In the second step, an appropriate *clustering approach* as well as a suitable clustering algorithm is selected and executed. Ideally, that clustering algorithm is

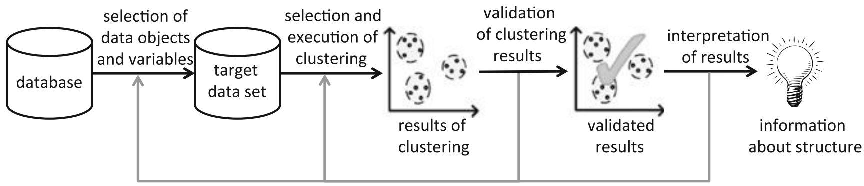


Fig. 4.3 Cluster analysis within the KDD (adapted from Halkidi et al. 2001)

chosen that leads to the best clustering result with regard to the underlying hypothesis. A variety of clustering approaches exists. Tan et al. (2009) propose the following criteria for the appropriate selection of a clustering algorithm:

- *Type and structure of clustering:* Cluster algorithms feature different structures, supporting or preventing the evaluation of the hypothesis. For applications in biology, hierarchical structures may be advantageous, for example, for the exploration of taxonomies, whereas the aggregation of a data set as required by transportation systems is well supported by partitioning algorithms.
- *Type and definition of clusters:* Cluster algorithms are based on different assumptions and definitions of a cluster. Partitioning methods are able to find circular and spherical clusters, whereas density-based methods feature arbitrarily shaped clusters.
- *Features of the data set:* Cluster algorithms require a well-defined target data set. Some methods are only suited for interval scaled data, for example. The required form has to be provided by appropriate preprocessing.
- *Number of data objects and variables:* Cluster algorithms differ in terms of their ability for scaling, i.e., some algorithms are very sensitive to the number of variables and data objects' dimensions.
- *Representation of results:* Cluster algorithms establish different ways of representation for resulting clusters. Some algorithms facilitate the assignment of data records to clusters only, whereas others determine a representative data record or data model.

The selection of a cluster algorithm expects general knowledge about the underlying system as well as detailed knowledge about the structure of the target data set. As shown in Fig. 4.3, the multiple execution of the clustering step allows for the comparison of different clustering approaches.

After the execution of a clustering algorithm, results are *validated* to determine the most suitable result with regard to the underlying hypothesis. At this point, different assumptions and abilities of particular cluster algorithms come into play. Based on subjective and objective criteria, the quality of clusterings can be evaluated qualitatively and quantitatively. System experts analyze results based on their system-specific knowledge through expert interviews, for example (subjective evaluation). If the quality of a clustering is not sufficient, the process is reconfigured and restarted. After validation and evaluation, results are interpreted and prepared for integration into applications, inducing further analysis by other DM techniques. Results may also lead to contradictions or new hypotheses for further investigation.

In the following section, a typology of clustering approaches is introduced. Functionality of selected algorithms is discussed in Sect. 4.2.2. Validation of clusterings is the subject of Sect. 4.2.3.

4.2.1 Clustering Approaches

In this section, different approaches of cluster analysis are introduced. Therefore, partitioning, hierarchical, density-based, grid-based, and model based algorithms are distinguished. For each category, the basic functionality is outlined. A summary of the most prominent algorithms is provided in Table 4.1, containing references to textbooks as well as to original publications.

Given a database of n data objects, a *partitioning* clustering approach constructs k partitions of the given data objects ($k \leq n$). Each partition represents a cluster. As a result, data objects are classified into k groups, where each group must contain at least one data object, and each data object must belong to exactly one group. In particular, an initial (random) partitioning is created and then improved by an iterative relocation technique, i.e., data objects are moved from one partition to another. The improvement is evaluated based on the distance of data objects within a cluster (which should be relatively low) and the distance between the clusters themselves (which should be relatively large). k-Means and k-Medoids are prominent representatives of this approach (cf. Sect. 4.2.2).

Hierarchical clustering approaches create a hierarchical decomposition of data objects, which can be represented by a tree of data objects. The agglomerative approach ("bottom up") regards each data object as a separate cluster and then successively merges them until a termination condition holds. In contrast, the divisive approach ("top-down") begins with all data objects in one cluster, which is successively split up into smaller clusters until a termination condition holds, for example, the desired number of clusters. Hierarchical decompositions can be visualized in dendrograms, showing the merger of data objects from the bottom to the top.

The main disadvantage of hierarchical approaches is the fixing of operations: once a merge or split is done, it cannot be undone. Thus, probably erroneous mergers cannot be corrected, leading to the risk of low-quality clusterings. Besides, hierarchical clusterings suffer from bad scaling, because the decision of merger or split needs to examine and evaluate a large number of data objects.

Density-based clusterings grow a given number of clusters as long as the density (i.e., the number of data objects) in the neighborhood exceeds some threshold. Since density-based clustering does not evaluate the distance between the individual clusters, but between single data objects only, clusters with arbitrary shape can be discovered. DBSCAN exemplifies the principle of the density-based approach (cf. Sect. 4.2.2). *Grid-based* algorithms assign data objects to a grid structure, i.e., they divide the data objects' space into a finite number of cells,

Table 4.1 Overview on clustering algorithms

Type	Algorithms	Sources
Partitioning algorithms	k-Means	Lloyd (1982), MacQueen (1967)
	k-Medoids (PAM, CLARA)	Kaufman and Rousseeuw (1990)
	k-Modes	Huang (1998); Chaturvedi et al. (1994), (2001)
	CLARANS	Ng and Han (1994), Ester et al. (1995), Bradley et al. (1998)
Hierarchical algorithms	Survey	Day and Edelsbrunner (1984)
	AGNES, DIANA	Kaufman and Rousseeuw (1990)
	BIRCH	Zhang et al. (1996)
	CURE	Guha et al. (1998)
	ROCK	Guha et al. (1999), (2000)
	Chameleon	Karypis et al. (1999), He et al. (2006)
Density based algorithms	DBSCAN	Ester et al. (1996)
	OPTICS	Ankerst et al. (1999)
	DENCLUE	Hinneburg and Keim (1998)
Grid based algorithms	STING	Wang et al. (1997)
	WaveCluster	Sheikholeslami et al. (1998)
Model based algorithms	Expectation Maximization (EM)	Dempster et al. (1977), Lauritzen (1995)
	AutoClass	Cheeseman and Stutz (1996)
	Conceptual clustering	Michalski et al. (1983)
	COBWEB	Fisher (1987)
	CLASSIT	Gennari et al. (1989)
	Self-organizing maps (SOM)	Kohonen (1982), Kohonen (1989), Carpenter and Grossberg (1991), Ritter et al. (1992), Kaski et al. (1999), Kohonen et al. (2000)
	Competitive learning	Rumelhart and Zipser (1985)
Textbooks and surveys	Hartigan (1975), Jain and Dubes (1988), Kaufman and Rousseeuw (1990), Arabie et al. (1996), Jain et al. (1999), Everitt et al. (2001), Parsons et al. (2004), Han and Kamber (2006), Tan et al. (2009), Berthold et al. (2010)	

which summarize information about the underlying data objects. Grid-based algorithms usually facilitate fast processing times, which are independent of the number of data objects, as well as parallel processing and incremental updating.

Partitioning and hierarchical clustering approaches require a distance function in order to properly assign data objects to clusters as well as to evaluate resulting clusters. Well-known similarity measures are the *Euclidean distance* and the *Manhattan distance*, which can be traced to the *Minkowski distance* (Kruskal 1964) . The Euclidean distance d between two p-dimensional objects i and j is defined as

$$d(i,j) = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \ldots + |x_{ip} - x_{jp}|^2},$$

whereas the Manhattan distance is defined as

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \ldots + |x_{ip} - x_{jp}|.$$

The generalizing Minkowski distance is given as

$$d(i,j) = \left(|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \ldots + |x_{ip} - x_{jp}|^q\right)^{1/q},$$

with q being a positive number. The Euclidean distance represents a linear distance between two points, whereas the Manhattan distance permits movements on x and y axis only. Since the Manhattan distance refrains from squaring distances, the impact of outliers is limited, which often bias the results of a clustering. Specific distance measures for binary, nominal, and ordinal scaled variables also exist, but are not discussed here; it is referenced to the aforementioned textbooks.

4.2.2 Clustering Algorithms

Based on partitioning and density-based clustering approaches, a selection of clustering algorithms being relevant for the analysis of data from traffic and transportation systems is discussed in the following. In particular, k-Means, k-Medoids and DBSCAN are described in detail.

4.2.2.1 *k*-Means

k-Means is one of the most prominent cluster algorithms, belonging to the group of partitioning clustering approaches. It has been introduced by MacQueen (1967) and works as follows. The algorithm iteratively minimizes the error sum ERR of data objects' p distances to the centroids c_i of k clusters C_i:

$$\mathrm{ERR} = \sum_{i=1}^{k} \sum_{p \in C_i} |p - c_i|^2$$

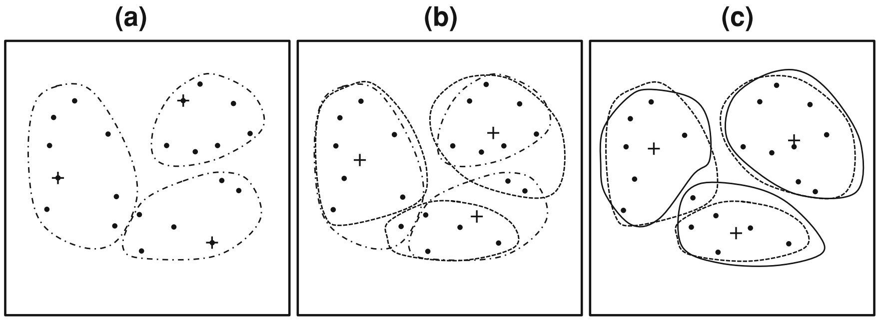


Fig. 4.4 Execution of *k*-Means

The *centroid* refers to the center of a cluster, representing the mean of all data objects within a cluster. *k*-Means considers interval scaled variables exclusively and implies the number k of desired clusters C_i and a distance function as input. It commonly leads to compact and well-separated clusters.

```
Algorithm 1: k-Means
Input: number k of desired clusters, distance function d(i,j)
Processing:
(1) arbitrarily choose k objects as the initial cluster centers;
(2) repeat...
(3) (re)assign each object to the cluster to which the object is the most
    similar, based on the mean value of the objects in the cluster;
(4) update the cluster means, i.e., calculate the mean value of the
    objects or each cluster;
(5) ... until cluster means converge
```

In `Algorithm 1`, *k*-Means is described by pseudo code. Exemplary execution of the algorithm is shown in Fig. 4.4. Centroids are marked by '+' and data objects by '•'. In Fig. 4.4a, the initial (random) assignment is depicted (step (1) and (2) of the algorithm), which is improved iteratively by recalculation of the cluster centers in b (step (3) and (4)). An update of cluster centroids leads to the final state in (c), where the assignment of data objects has converged. Since the initial assignment may lead to a local optimum only, *k*-Means should be executed several times with different initial assignments of customers.

k-Means is a very efficient heuristic, since only distances between data objects and centroids as well as the centroids themselves have to be calculated. Thus, the number of combinatorial assignments is limited, which recommends the algorithm for the processing of large data sets. Besides, the algorithm is simple and easy to code. *k*-Means also shows some disadvantages, though: only interval scaled variables are supported, and the algorithm is capable of determining circularly shaped clusters only. It is sensitive to outliers, because the distance calculation refers to means. Furthermore, the number of clusters has to be defined in advance, although the optimal number of desired clusters may not be known beforehand. Thus, *k*-Means implies a procedure that estimates the preferable number of clusters (cf. Sect. 4.2.3), or the number of clusters has to be determined by means of experiment.

4.2.2.2 *k*-Medoids

k-Medoids differs from *k*-Means in a different representation of the cluster centers, i.e., instead of taking the mean value as a reference, the medoid is used. In contrast to centroids, which represent an "artificial" reference point, medoids locate an existing data object. The medoid is defined as the most centrally located object in a cluster. Data objects are assigned to that cluster that is the nearest in terms of its medoid. As only pair-wise distances between data objects and medoids have to be calculated, *k*-Medoids is capable of handling mixed scaled variables. The quality of a tentative clustering is evaluated by

$$\mathrm{ERR} = \sum\nolimits_{i=1}^{k} \sum\nolimits_{p \in C_i} |p - o_i|$$

with p as data object being assigned to cluster C_i represented by its medoid o_i.

The algorithm works as follows: find k clusters for p data objects by arbitrarily setting a representative medoid for each cluster. Each remaining data object is assigned to the medoid to which it is the most similar. Medoids are iteratively replaced by one of the non-medoids as long as the quality of the clustering can be improved, i.e., ERR is reduced. The corresponding pseudo code is shown in `Algorithm 2`.

Implementations of *k*-Medoid differ at most in the selection of initial medoids. One of the first *k*-Medoid variants by Kaufman and Rousseeuw (1990) determines initial medoids more sophisticatedly, for example. The additional effort in the initialization phase is contrasted by lower computational efforts during the update of medoids. *k*-Medoids is not as sensitive to outliers as *k*-Means. Nonetheless, the relative large number of pair-wise comparisons leads to higher computational efforts and poor scaling. As *k*-Means, *k*-Medoids implies the number k of desired clusters to be specified in advance.

`Algorithm 2:` *k*`-Medoids`

`Input: number` k `of desired clusters`
`Processing:`
`(1) arbitrarily choose` k `objects as the initial medoids;`
`(2) repeat …`
`(3) assign each object to the cluster to which the object is the most similar, based on the medoid` o_i;
`(4) randomly choose a new medoid` o_{random};
`(5) calculate costs` S `for every possible exchange of` o_j and o_{random};
`(6) if`$S < 0$, `accept` o_{random} `as new medoid and omit` o_j;
`(7) … until cluster medoids converge`

4.2.2.3 DBSCAN

DBSCAN is a prominent representative of density-based clustering approaches. Density-based cluster algorithms regard clusters as dense regions of objects in the "data space", which are separated by regions of low density. Contrasting most partitioning algorithms, DBSCAN is able to detect clusters of arbitrary shape, i.e., it is not limited to circularly shaped clusters. The main idea of DBSCAN is to search for clusters by checking the neighborhood of each data object. If the neighborhood of a data object p contains more than a given number of *MinPts* data objects, a new cluster around p is created. DBSCAN iteratively merges clusters around data objects when they "overlap".

In particular, the neighborhood of a data object p is defined by an ε-neighborhood. If the ε-neighborhood around p contains at least the minimum number of data objects *MinPts*, then p is called a *core object*. Data objects are distinguished into *density-reachable* and *density-connected* objects with respect to ε and *MinPts* (Ester et al. 1996). p is density-reachable from data object q if there is a chain of data objects $p_1,\cdots,p_n$ where $p_1 = q$ and $p_n = p$ such that p_{i+1} is directly density-reachable from p_i. p is density-connected to q if there is a data object o such that both p and q are density-reachable from o. An example is given in Fig. 4.5. At this point, let $MinPts = 3$. Then, m, p, o and r are core objects, because each of them is in an ε-neighborhood containing at least three points. q is directly reachable from p, since there is a chain of core objects that connects q and p. o, r and s are all density-connected, as they are linked by "overlapping" neighborhoods of core objects.

DBSCAN checks the ε-neighborhood of each data object p in the database, and if the neighborhood contains more than *MinPts*, a new cluster with p as core object is created. Directly density-reachable objects are collected from these core objects, leading to the merger of density-reachable clusters. The algorithm terminates when no additional data object can be added to any cluster. The corresponding pseudo code can be found in Berthold et al. (2010).

As shown above, a large variety of clustering algorithms for different purposes and with different advantages and disadvantages regarding different types of data sets exist. Most prominent algorithms are well-elaborated and thus available within open-source software such as the *Konstanz Information Miner* (KNIME, www.knime.org). KNIME comprises state-of-the-art DM algorithms for clusterings, decision trees, support vector machines, etc. It is a modular data exploration platform that enables the user to visually create data flows, selectively execute some or all analysis steps of the KDD, and later investigate the results through interactive analysis of data sets and information models. Berthold et al. (2010) provide an introduction into the usage of KNIME. Another option is the tool WEKA (www.cs.waikato.ac.nz/ml/weka/), which comprises a collection of machine learning algorithms for the solution of DM tasks (Hall et al. 2009). WEKA is seamlessly integrated into the open-source software RapidMiner (www.rapid-i.com), featuring an interactive support of DM algorithms in terms of KDD modeling and corresponding algorithms for data analysis, data evaluation, and data visualization.

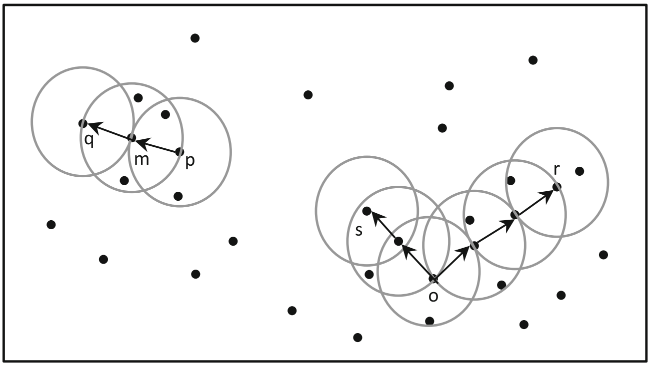


Fig. 4.5 Principle of DBSCAN (adapted from Han and Kamber 2006)

4.2.3 Validation of Clusterings

For the validation of clusterings, no standard approach exists. In the literature, concepts like "cluster evaluation" (Tan et al. 2009), "cluster validation" (Halkidi et al. 2001) and "cluster validity" (Jain and Dubes 1988) are discussed. Estivill-Castro (2002) defines the quality of a clustering in terms of confidence, validity, and naturality with respect to the underlying data set.

Validation may occur by computation of *internal, external* and *relative* measures. From a machine learning perspective, internal measures are referenced as unsupervised measures, whereas external measures are referenced as supervised measures. Internal measures evaluate the quality of clusterings based on the underlying data set exclusively. They can be distinguished into measures that denote *cluster cohesion* or *cluster separation,* respectively. External measures evaluate a clustering by comparison of clusters with a well-known (external) structure or information (e.g., expert knowledge). Relative measures compare different clusterings by combination of internal and external measures. Jain and Dubes (1988) give a comprehensive review on clustering measures.

In the following, the focus is on internal measures, since they can be derived from the underlying data set immediately. Three exemplary validation methods are discussed: *Hopkin's statistics* estimates the clustering tendency of a target data set in order to evaluate its potential suitability for clustering. The *Davies-Bouldin index* compares the quality of different clusterings. The *elbow criteria* allows for the estimation of the suitable number of clusters. The definition of external measures may be supported by EDA, cf. Sect. 4.3.

4.2.3.1 Clustering Tendency (Hopkin's Statistics)

The potential suitability of a target data set for clustering can be investigated by *clustering tendency* measures (Jain and Dubes 1988). Clustering tendency measures denote if a data set tends to natural clusters, inducing the general applicability of clustering approaches. Clustering tendency tests estimate the expected usefulness of clusterings. They do not require the actual computation of clusters. Corresponding tests are also known as tests on the "spatial randomness of data" (Banerjee and Davé 2004). Clustering tendency tests prevent the superfluous application of clustering approaches to non-suited target data sets.

A prominent approach for the estimation of clustering tendency is Hopkin's statistics (Jain and Dubes 1988; Tan et al. 2009). For the computation of the corresponding measure, data objects of the target data set are selected randomly. Then, distances w_i to their nearest neighbors are calculated (cf. Fig. 4.6). New data objects are randomly generated and compared to their nearest neighbors in terms of distances u_i. More formally, the Hopkin's statistics in d-dimensional data space is defined as

$$HOP = \frac{\sum_{i=1}^{p} u_i^d}{\sum_{i=1}^{p} u_i^d + \sum_{i=1}^{p} w_i}.$$

If the present data objects' distances w_i dominate, one can assume a cluster structure in the original data set (HOP > 0.5). If the present data objects' distances w_i are comparable to random data objects' distances u_i, the original data set does not tend to natural clusters (HOP ≤ 0.5).

4.2.3.2 Davies-Bouldin Index

The Davies-Bouldin Index (DBI) is an internal measure for the validation of clusterings (Davies and Bouldin 1979; Stein et al. 2003). DBI considers the variation of data objects within (intra-cluster) and between (inter-cluster) clusters. The index has originally been developed for the determination of the optimal number of clusters in hierarchical clustering approaches, i.e., at what point of merging or splitting resulting clusters establish the most natural clustering. DBI evaluates cluster separation and cluster cohesion. It is calculated as follows:

$$\mathrm{DBI} = \frac{1}{k} * \sum_{i=1}^{k} R_i$$

$$R_i = \max_{j=1\ldots k, i \neq j} R_{i,j} \text{ and } R_{i,j} = \frac{s(C_i) + s(C_j)}{s(C_i, C_j)}$$

s: $C \to \boldsymbol{R}$ denotes the variation within a cluster (cohesion), whereas δ: $\mathrm{C} \times C \to \boldsymbol{R}$ evaluates the distance between two clusters (separation).

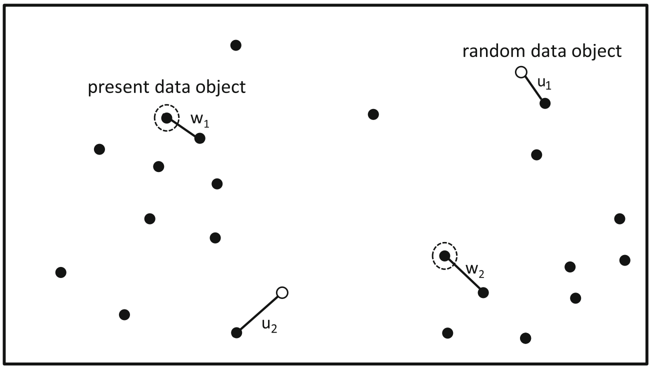


Fig. 4.6 Hopkin's statistics

A typical definition for cohesion and separation of a clustering with centroids c_i and clusters C_i is the average distance $s(C_i)$ of all objects within a cluster and the corresponding centroid as well as the distance $\delta(C_i, C_j)$ between two centroids:

$$s(C_i) = \frac{1}{|C_i|} \sum_{x \in C_i} |x - c_i|$$

and

$$\delta(C_i, C_j) = |c_i - c_j|$$

If variation of data objects within a cluster is rather low and distances between clusters are relatively high, $R_{i,j}$ is rather low, denoting compact and separated clusters. This usually conforms to a good clustering result, i.e., clusterings with smaller DBI are preferred. The idea is to determine the optimal number k of clusters successively by increasing k until DBI denotes a (local) minimum.

4.2.3.3 Suitable Number of Clusters (Elbow Criterion)

For the determination of the suited number of clusters, the elbow criterion can be applied (Ketchen and Shook 1996). The tradeoff is as follows: on the one hand, the number k of clusters must be large enough to give a good approximation of the underlying data set. On the other hand, k should be kept as small as possible in order to minimize the data input for subsequent analysis, facilitating comprehensible visualization, presentation, and evaluation.

A meaningful estimation of the number of clusters can be achieved by the determination of internal indices, which evaluate the quality of a clustering. The idea is to repeat a cluster analysis with ascending size of k, and then compare the results of the specific index (Tan et al. 2009; Jain and Dubes 1988). In partitioning clustering approaches, for example, such an index may be the *sum of squared errors*, which is supposed to decrease with increasing number of clusters.

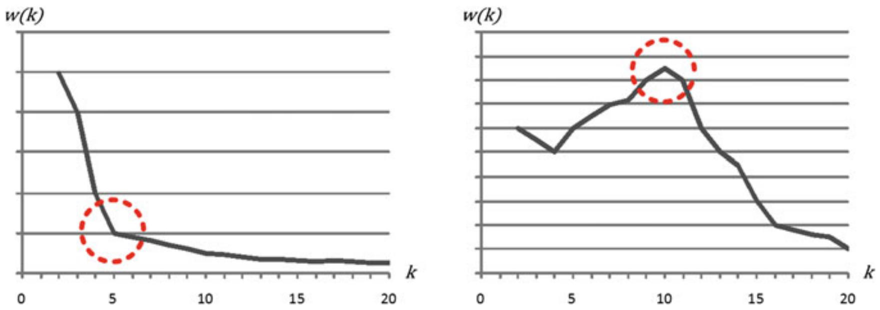


Fig. 4.7 Internal indices depending on the size of *k*

In particular, the result of the internal index *w(k)* is visualized for each clustering. A common approach is to identify "leaps" or "elbows" to get indications for a reasonable number of clusters. In Fig. 4.7, two different internal indices are depicted. On the left-hand side, *w(k)* is supposed to decrease with increasing *k*, leading to a first local optimum at $k = 5$. On the right-hand side, a noticeable "elbow" can be seen at $k = 10$, which indicates a potentially appropriate number of clusters.

The elbow criterion allows for a simple estimation of the number of clusters. However, the method features some drawbacks. If clusters overlap or if they are nested, results can be misleading (Tan et al. 2009). Thus, further evaluation by EDA or by examplary application are required to validate the selected number of clusters. Although the elbow criterion may have suggested a specific number, results may not be interpretable or reasonable for their intended use.

4.3 Exploratory Data Analysis

EDA is concerned with the visualization of mass data in order to reveal and evaluate structures within a data set. It may also serve as a data-driven technique for the generation of hypotheses (Hand et al. 2001). Data records are examined by visualization of structures that indicate relevant relationships between attributes.

Within the KDD, EDA can undertake several functions. In the preprocessing step, data records can be visualized to identify abnormalities such as outliers or distributions of raw data. In the DM step, EDA addresses the identification of patterns by visual inspection of the data. For visualization of information models and patterns, EDA serves as verification technique to support the evaluation and interpretation of DM results.

Techniques for the exploration of low-dimensional and high-dimensional data sets are manifold. Based on Tan et al. (2009); Berthold et al. (2010), the following list summarizes the most prominent approaches. For low-dimensional data, methods from the area of common statistics are applied. For high-dimensional data, more sophisticated methods are reported.

Low-dimensional data

- *Summary statistics* describe a data set by its mean and standard deviation, frequencies, measures of location, and spread.
- *Bar charts* depict the frequencies for the values of a categorical attribute.
- *Histograms* show the frequencies for a numerical attribute. The range of the numerical attribute is discretized into a fixed number of intervals.
- *Boxplots* are a very compact way to visualize and summarize the main characteristics of a sample from a numerical attribute, including mean and quartiles.
- *Scatter plots* refer to displays where two attributes are plotted against each other.

High-dimensional data

- *Principal component analysis* is a statistical method supporting the determination of a projection to a plane, preserving the original variance in the data set. The visualization of high-dimensional data in a lower dimensional space is obtained by projection to a linear subspace.
- *Multidimensional scaling* is also a technique for the reduction of dimensions, but it is not restricted to mappings by simple projections. Multidimensional scaling does not construct an explicit mapping of the high-dimensional space to the low-dimensional space. Instead, data objects are arranged in the low-dimensional space, preserving the distances between the data objects instead of the variance of the original data set.
- *Parallel coordinates* visualize data by organizing attributes in parallel axes. For each attribute, a polyline is drawn, connecting the values of the data object for the attributes on the corresponding axes. There is no limitation for the number of axes to be displayed.

Since EDA is a quite interactive task, the usage of sophisticated tools for visualization and presentation of data objects is common. KNIME and WEKA are two recent examples for DM tools featuring popular DM techniques as well as visualization functionality, supporting interactive analysis and evaluation of results. For evaluation of real-world data in the area of traffic and transportation systems, GIS facilitate spatial and temporal interpretation of DM results. A recent example is *Google Earth*, which features aerial views of the earth's surface including digital roadmaps and additional information such as three-dimensional views of urban areas (http://earth.google.com). Google Earth supplies an open interface in terms of the *Keyhole Markup Language* (KML, http://code.google.com/intl/de-DE/apis/kml/documentation/), which allows for integration into a variety of applications. KML is a derivate of the Extended Markup Language (XML), describing geo data by geographical positions, type, style, and temporal information. KML has become an open standard named the OpenGIS KML Encoding Standard (OGC KML). In `Example 1`, a KML file is shown, exemplifying the geographical position of the city of Zurich, Switzerland.

Example 1: Description of a point with KML

```
<?xml version=''1.0'' encoding=''UTF-8''?>
<kml xmlns=''http://www.opengis.net/kml/2.2''>
<Document>
<Placemark>
<name>Zurich</name>
<description>Zurich</description>
<Point>
<coordinates>8.55,47.3666667,0</coordinates>
</Point>
</Placemark>
</Document>
</kml>
```

Possibilities and approaches of EDA are by far not exploited by this section. For further reading, it is referred to Tukey (1977); Tufte (1983); Chambers et al. (1983); Jacoby (1997); Wilkinson (1999) who present overviews on data visualization techniques. Hand et al. (2001) give a literature overview on methods for the analysis of low- and high-dimensional data sets.

Part III
Integration of Information Models

Chapter 5
Analysis of Floating Car Data

Planning systems for routing in city logistics require realistic travel time estimations. Time-dependent information models establish such travel times at different levels of aggregation. The starting point for the construction of time-dependent information models is a sufficient number of system appearances in terms of historical traffic data. Historical traffic data can be derived from telematics-based data collection to a sufficient extent at low costs.

In this chapter, the KDD is instantiated for analysis and aggregation of historical travel times in urban areas. To this end, technology for data collection, analysis of historical travel times by DM techniques, and verification of information models by EDA are investigated. Historical travel times are transformed into planning data sets of different volume and complexity.

In particular, traditional as well as telematics-based approaches for the collection of travel times are discussed. In contrast to the traditional approach, telematics-based data collection supplies a sufficient number of historical travel times. Nonetheless, in their pure form, historical travel times are not suited as input for planning systems, since they refer to individual measurements only. Thus, the KDD is instantiated and implemented for the aggregation of FCD (cf. Fig. 5.1), ranging from telematics-based data collection of raw travel times to the provision of time-dependent information models and corresponding planning data sets. Raw empirical traffic data are transformed into first level aggregated data and into second level aggregated data. The elements involved are as follows:

- *Data collection*. The generation of time-dependent information models expects a sufficient number of empirical travel times. Taxi-FCD as a recent GPS-based data collection method supplies travel times for urban areas at low costs. The corresponding technology is introduced and delimited to traditional traffic data collection approaches (Sect. 5.1).
- *Preprocessing*. In the preprocessing step, erroneous Taxi-FCD records are removed. Deficient speed observations arising from measurement failures are

J. F. Ehmke, *Integration of Information and Optimization Models for Routing in City Logistics*, International Series in Operations Research & Management Science 177,
DOI: 10.1007/978-1-4614-3628-7_5, © Springer Science+Business Media New York 2012

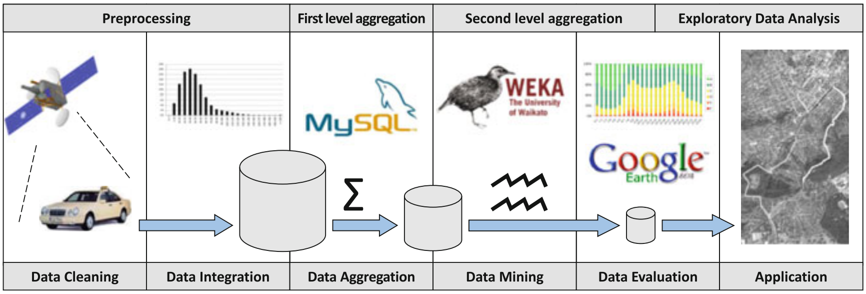


Fig. 5.1 KDD in the context of FCD analysis

filtered. Distributions of Taxi-FCD give information about the validity of measurements as well as on potential outliers. Taxi-FCD records (empirical traffic data) are amended by a common digital roadmap (infrastructure data). The data are integrated into a central database. FCD related details are discussed in Sect. 5.2.

- *First level aggregation*. The provision of comprehensive planning data occurs by aggregation of empirical traffic data. Here, FCD evolve into time-dependent travel times by simple aggregation (Sect. 5.3). The management of large data volumes is undertaken by a MySQL database. The resulting planning data set is too voluminous for efficient consideration in planning systems, though, inducing further aggregation by cluster analysis.
- *Second level aggregation*. Time-dependent travel times resulting from first level aggregation are analyzed by DM techniques. Cluster analysis is used for the generation of a compact time-dependent information model. Links are clustered according to their time-dependent speed variation by cluster algorithms available in the DM tool WEKA. Thus, allocation of compact time-dependent travel time data sets for advanced planning systems is facilitated (Sect. 5.4).
- *Exploratory Data Analysis*. Information models from first and second level aggregation provide time-dependent travel times, which are subject to presentation and evaluation by EDA. Since travel times are inherently spatio temporal, they are visualized in daily courses to be compared to well-known traffic patterns. Here, Google Earth is involved (Sect. 5.5). Also an application-oriented evaluation of information models for routing in city logistics is introduced. Corresponding computational experiments are conducted in Chaps. 7 and 9.

Implementation of the KDD and subsequent experiments is based on Taxi-FCD collected in the urban area of Stuttgart, Germany. The data are usually one of several inputs for a traffic monitoring system. About 9 million itineraries are realized in the wider area of Stuttgart every day. Over the past years, a large increase in passenger kilometers to a level of 119 million passenger kilometers per working day has been expected (Verband Region Stuttgart 2002). Due to this challenge, Taxi-FCD has continuously been collected since 2003. The DLR has provided Taxi-FCD of the years 2003–2005, making a total of about 230 million data records.

5.1 Data Collection

The generation of time-dependent information models for routing in city logistics requires information about traffic network behavior. Typical behavior of urban traffic networks is represented by *traffic quality,* which refers to an aggregate measure of service quality offered by the transportation infrastructure (Ehmke et al. 2010). In recent years, telematics-based traffic data collection has improved the availability of traffic data in urban areas, enabling the city-wide determination of traffic quality measures.

In the following, the traditional approach for the determination of traffic quality in urban road networks is described (Sect. 5.1.1). Then, this approach is extended with respect to emerging technologies contributing to the field recently. The role and functionality of telematics-based systems such as FCD is discussed (Sect. 5.1.2). In contrast to the traditional approach, it is referred to DM for aggregation and analysis of traffic data as a prerequisite for the generation of traffic quality measures.

5.1.1 Traditional Approach

The starting point for the determination of traffic quality is the collection of empirical traffic data. Reliable routing decisions must be derived from these raw data. Therefore, empirical traffic data have to be transformed into time-dependent information models. The determination of traffic quality traditionally consists of three basic steps (cf. Fig. 5.2):

- *Data collection.* The first step refers to the collection of (empirical) traffic data by (manual) traffic census. Several collection methods for the collection of microscopic traffic data exist, see below.
- *Data analysis.* Microscopic traffic data have to be aggregated to macroscopic traffic measures in the data analysis step, which comprises conventional data aggregation and statistical analysis. Average travel times are calculated or derived from traffic flow models.
- *Data evaluation.* The derivation of traffic quality measures demands for an aggregated, system-wide analysis of macroscopic traffic measures. To this end, traffic measures are evaluated. Evaluation can be done by evaluation schemes such as the Highway Capacity Manual (HCM, National Research Council 2000), leading to evaluated macroscopic measures denoting traffic quality.

The data collection step is usually carried out by stationary sensors [e.g., infrared sensors, induction loops (cf. Fig. 5.3), video surveillance] or by manual short-time census. Traffic flows on urban roads are subject to a large variety of influences leading to modeling obstacles, though. A detailed reconstruction of travel times from traffic flow samples is not always possible (van Woensel et al. 2008). Furthermore, it is often virtually impossible to generalize travel times

	DATA COLLECTION	DATA ANALYSIS	DATA EVALUATION
step	(manual) traffic census	data aggregation	measure evaluation
method	collection method	traffic flow model	evaluation scheme
result (goal)	traffic data	traffic measure	traffic quality

Fig. 5.2 Traditional approach: process steps for the determination of traffic quality

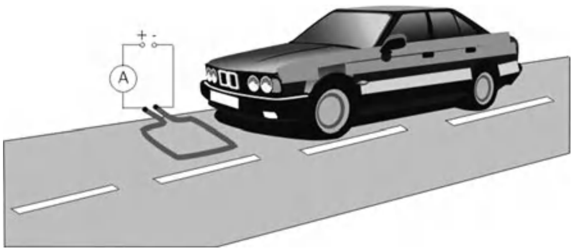


Fig. 5.3 Conventional traffic data collection by loop detectors

obtained for one certain network link to other links without detailed infrastructure data (e.g., vertical alignment or road profile) as well as the function and the location of the road section (Gössel 2005). Due to cost issues, traditional data collection methods do not cover the whole city road network, but are located at only a few, significant points of the city road network (Gühnemann et al. 2004).

Data analysis commonly occurs by conventional data aggregation and statistical analysis. Data samples are utilized for the instantiation of traffic flow models (e.g., speed-flow diagrams, hydrograph curves, cf. Daganzo 1997). In recent years, much work has been done with respect to traffic data analysis for wide area networks. For instance, Vanajakshi (2004) derives travel time estimations from loop detector data, whereas Wu et al. (2004) specialize on travel time estimation on freeways by support vector regression. Chrobok et al. (2004) sum up approaches of aggregating loop detector data to analyze and predict traffic flows. Kim et al. (2005) consider traffic data from 100 loop detectors in Southeast Michigan to model time-varying traffic flows based on a Markov decision process.

Traffic flows in urban road networks are highly fluctuant with respect to different network links and times of the day. To derive traffic quality in terms of city-wide, time-dependent travel times, area-wide data collection is necessary. Empirical traffic data have not been available to a sufficient extent due to prohibitive census costs up to now. Such data have recently arisen from telematics-based traffic data collection.

5.1.2 Telematics-Based Approach

The weaknesses of the traditional approach may be alleviated by the use of emerging technologies. In particular, data collection can be improved by telematics systems such as FCD. The term *telematics* refers to the transmission of data over a

	DATA COLLECTION by telematics	DATA ANALYSIS	DATA EVALUATION
step	(automated) traffic census	data aggregation	measure evaluation
method	floating car data	traffic flow model	evaluation scheme
result (goal)	traffic data	traffic measure	traffic quality

Fig. 5.4 Telematics-based approach: telematics complements traffic data collection

telecommunication network, followed by the automated processing of these data. Goel (2008) gives an overview on enabling technologies for applications with respect to transportation systems, i.e., wireless communication, positioning systems, and GIS.

From a traffic research point of view, telematics-based information systems are usually discussed with respect to their features for traffic management and traffic control, namely "Advanced Traveler Information Systems" (ATIS). According to Toledo and Beinhaker (2006), ATIS are designed to provide users with information about the current state of the traffic system. They are based on a technological infrastructure that collects data, processes them to generate traveler information and guidance, and disseminates the resulting information to the users.

ATIS comprise a wide area of sensor technologies that monitor traffic conditions. Collected data is transmitted to a central management center, where it is automatically processed and analyzed to extract the information of interest, for example, expected travel times, incidents, weather and road conditions, and speed recommendations. Traffic information is disseminated to users via various media, for example, web sites, wireless communication technology, and variable message signs.

5.1.2.1 Data Collection by FCD

FCD technology enriches or even substitutes manual traffic census by automated traffic census, especially for the improvement of ATIS. In Fig. 5.4, substitution of conventional data collection in the process of traffic quality determination is depicted, resulting in large amounts of data supporting the analysis of traffic quality in urban areas.

Several types of FCD may contribute to improved data collection:

- *Floating Phone Data* refers to travel times collected in cellular networks. Movements of a mobile phone lead to altering of the cell the mobile phone is assigned to. Mobile network operators are able to anonymize these data such that it can be used for the derivation of average travel times. The German cellphone provider O2 collects and anonymizes location data arising from about 17 million cellphones for the analysis and the prediction of traffic jams, for example. Vodafone operates an even larger data collection project, observing movements of more than 30 million customers for the improvement of navigation systems (Teltarif.de 2011).

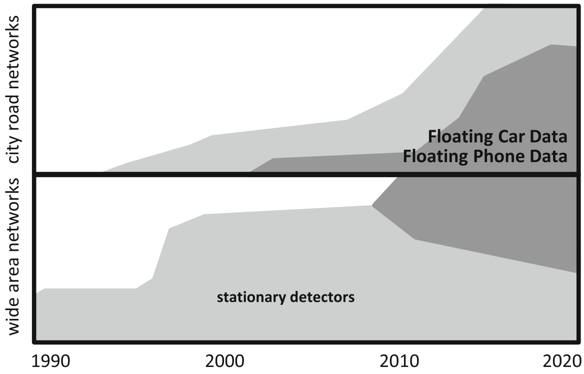


Fig. 5.5 Traffic data collection in Germany (adapted from Fastenrath 2008)

- *Floating Truck Data* is a convenient data source for the collection of travel time data in wide area networks. Trucks equipped with a GPS receiver transmit their current state within the traffic flow. In the UK, *KeepMoving* supplies travel time estimations for freeways based on text messages of about 22,000 probe vehicles (www.keepmoving.co.uk). Information is disseminated via websites, navigation systems, and mobile phone applications.

The evolution of data collection technology in Germany is exemplified in Fig. 5.5. Here, it is distinguished between traditional, stationary data collection, and FCD technology. At the time of 2010, the largest density of stationary sensors can be found at freeways. In urban areas, only a limited number of sensors exist. Here, FCD-based data collection has been introduced for several years; increasing importance and availability of FCD based sensors is predicted for oncoming years.

For an overview on FCD projects worldwide in the context of traffic and transportation systems, see Bishop (2005). A discussion on FCD applications and their recent extensions can be found in Messelody et al. (2009). For an overview on FCD technology and its application in traffic management, see Lorkowski et al. (2005), and Breitenberger et al. (2004).

5.1.2.2 Data Collection by Taxi-FCD

For traffic data collection, this work refers to the *Taxi-FCD* system run by DLR. DLR uses taxis as mobile data sources ("probe vehicles") for the automated collection of FCD. Taxi-FCD supplies traffic data in terms of travel times for a single vehicle. To this end, a fleet of taxis must be equipped with communication devices, enabling area-wide collection and transmission of traffic data. DLR collects and analyzes Taxi-FCD in a number of cities worldwide, for example, Hamburg, Berlin, Stuttgart, Munich (Germany), Vienna (Austria), and Ningbo (China).

The typical architecture of a Taxi-FCD system is shown in Fig. 5.6. GPS locations are collected in the taxi office and then sent to the traffic information center, where vehicles' individual trajectories are derived from the locations. These

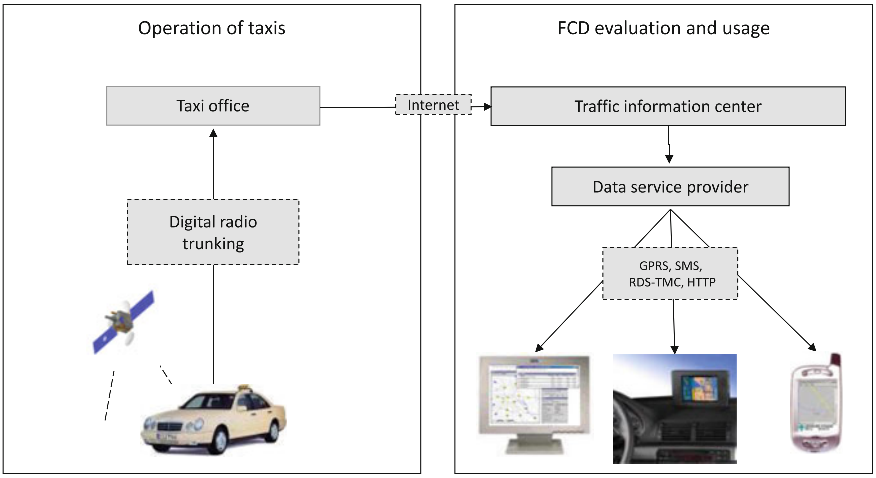


Fig. 5.6 Taxi-FCD system (adapted from Gühnemann et al. 2004)

trajectories are matched to a digital roadmap (cf. Sect. 6.1.2), and travel times are assigned. Raw travel time data is filtered to eliminate erroneous GPS signals, implausible travel time measurements, and nonrepresentative data records, for example, data resulting from taxis using bus lanes where regular traffic is prohibited. DLR has developed several map matching and data handling algorithms. For an overview on the functionality of Taxi-FCD see Lorkowski et al. (2004).

Taxi-FCD is commonly utilized for the depiction of the current traffic state, for off-board navigation, for the real-time disposition of fleets, and for the construction of digital roadmaps (Lorkowski et al. 2005). The determination of the current traffic state and short-term traffic prediction requires the smoothing of potentially noisy FCD. If a sufficient number of taxis are operating in desired network areas, i.e., the penetration rate is sufficient, this can be done by simple aggregation. Otherwise, more sophisticated smoothing approaches are required. Sohr et al. (2009) derive daily curves from FCD by approximation of Lomb functions, for example.

A Taxi-FCD record represents the travel time of a single vehicle being part of the current traffic flow. Speeds are derived from the location of taxis and subsequent map matching. Given a fleet of taxis operating in an urban area, it is possible to collect a sufficient number of speed measurements for almost all of the links of the traffic network. The structure of a resulting speed record being subject of subsequent analysis is shown in Table 5.1, representing a single measurement of a single taxi at a certain location and date.

For Taxi-FCD collection, an important question is the number of vehicles required for a representative measurement campaign, since taxis operate with varying spatio temporal intensity. Brockfeld et al. (2007a) discuss the penetration rate of Taxi-FCD systems in relation to overall traffic. They present results from an urban measurement campaign and conclude that “the few taxi data may be able to

Table 5.1 Structure of FCD records

Time	Link	Speed
Time of positioning	Road segment ID	Calculated speed [km/h]
2003-08-01 07:01:22	54362718	50.73

represent the characteristics of the whole traffic stream." The reliability of measurements derived from Taxi-FCD is investigated by Brockfeld et al. (2007b), who conduct test drives and show that differences between the travel time estimates of the Taxi-FCD system and the conducted test drives are in the range of traffic system immanent variation.

In the following, the KDD is instantiated by 230 million FCD records of the area of Stuttgart.

5.2 Preprocessing

In the preprocessing step, incomplete or apparently erroneous data records are removed. Here, Taxi-FCD records are revised (data cleaning) and therefore investigated in terms of range, completeness, and denomination. Furthermore, Taxi-FCD and infrastructure data (based on a digital roadmap) are integrated into a single travel time database (data integration). For FCD analysis, the integration of the digital roadmap data schema with link IDs of FCD records is crucial for spatial interpretation.

Concerning the given Taxi-FCD records (cf. Table 5.1), the following preprocessing activities have been undertaken.

5.2.1 Attribute "Time"

Taxi-FCD has been collected in the years 2003–2005. The majority of data records are in the expected time range. In individual cases, missing entries occur, which cannot be complemented by, for example, estimated values, since no systematic deficiencies of the Taxi-FCD system are known. Affected data records are deleted, because the temporal interpretation of data records strongly depends on recorded time and date.

5.2.2 Attribute "Link"

Link IDs relate speed measurements to road sections. A link is the smallest entity within a road network, connecting two points of interests (e.g., two intersections).

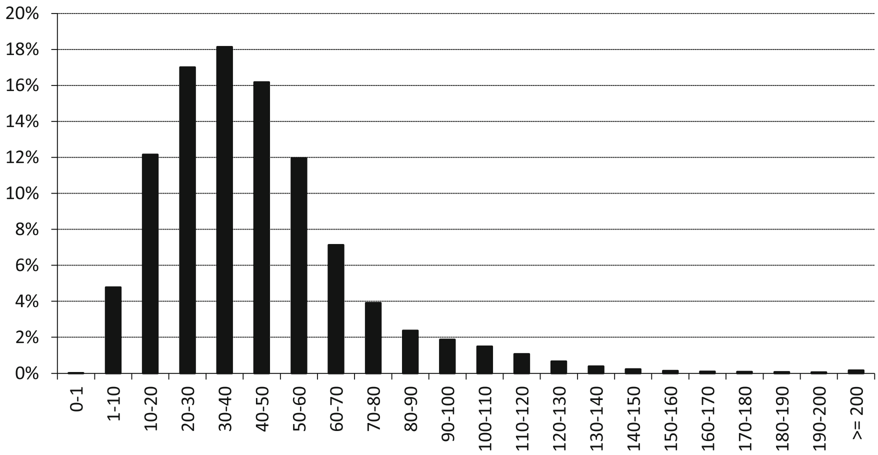


Fig. 5.7 Distribution of speed measurements before outlier elimination

To ensure spatial interpretation and proper assignment of measurements to the road network, given link IDs must match with a digital roadmap of the corresponding area. Digital roadmaps represent the road network in the travel time database and supply spatial and physiognomic data for each network link (cf. Sect. 6.1.1). At this point, some FCD records appertaining to other conurbations appeared, and hence did not match the digital roadmap of Stuttgart. Corresponding data records were removed because spatial interpretation of measurements is impossible without compatible link data.

5.2.3 Attribute "Speed"

This attribute contains the core information for the generation of time-dependent travel times. The calculation of statistical measures is required to identify erroneous data records by computation of the arithmetic mean of values, minimum and maximum as well as the underlying distribution of the attribute's values.

The distribution of all speed measurements is depicted in Fig. 5.7. Here, a log-normal speed distribution is shown, which corresponds to typical speeds driven in urban areas. However, speed measurements larger than 200 km/h are suspicious. Although freeways are part of the city road network of Stuttgart, there is a high probability that extreme measurements result from measurement failures such as GPS shadings, which increase the calculated speed extremely. Even if such a speed had actually been realized, a consideration for travel time estimation would not be reasonable in a city logistics context. Thus, extreme measurements are treated as outliers. In particular, all speed measurements are filtered that exceed a level of legal speed $\times 1.5$ (the legal speed is available from digital roadmap data). This

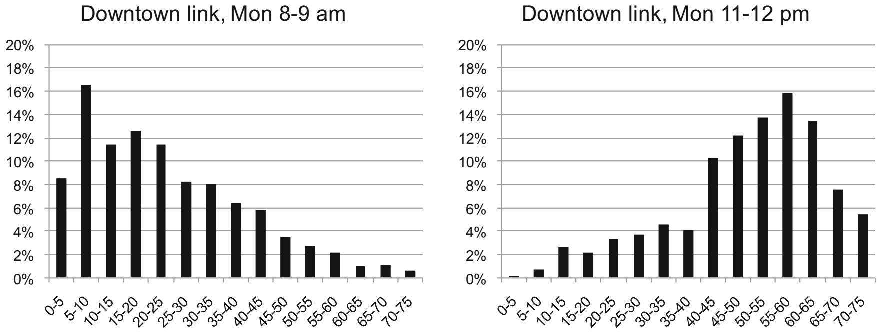


Fig. 5.8 Relative frequencies of speed measurements for different time bins (in fractions, km/h)

eliminates potential speed biases and ensures a reasonable foundation for the estimation of typical travel times. About 3% of all speed measurements are affected.

Next to an overall picture, the distribution of measurements for specific links and times of the day may offer insights into the suitability of collected speed data for travel time estimation. Figure 5.8 depicts two exemplary distributions for a downtown link, distinguishing the relative frequencies of measurements in the morning rush hour from measurement frequencies at night. In the morning, a propensity of frequencies in the direction of lower speed classes is clearly visible, whereas in the night a more or less free traffic flow is shown. This underlines the necessity of time-dependent travel times for routing in city logistics, where such situations have not been considered by common planning systems yet.

5.2.4 Temporal Distribution of Measurements

Although the collection of Taxi-FCD results in a voluminous database of historical travel times, FCD covering varies spatially and temporally, depending on the utilization rate of taxi services. In Fig. 5.9, an impression of the temporal distribution of FCD measurements is given, distinguishing measurements within 24 h of a weekday and 7 days of the week. During the day, about the same overall number of Taxi-FCD is available for each hour at each working day, amounting to 1.5 million measurements each. In the night hours of working days, a relatively small number of FCD are available only, contrasted by a high number of measurements at weekends.

5.2.5 Spatial Distribution of Measurements

In Fig. 5.10, the spatial distribution of collected Taxi-FCD is depicted by means of an aerial view produced by Google Earth. The overall number of Taxi-FCD measurements

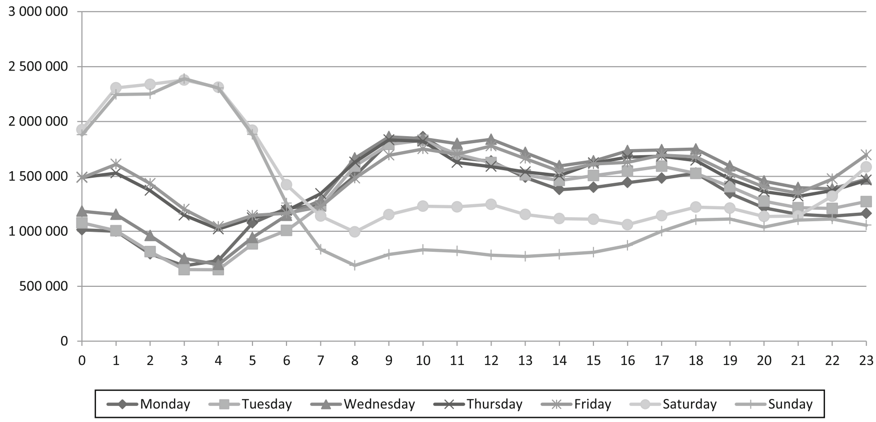


Fig. 5.9 Temporal distribution of FCD measurements

is implied for each link, derived by coloring road sections according to the number of recorded Taxi-FCD measurements. Red or orange links denote a rather small number of Taxi-FCD, whereas yellow and green links indicate a rather large number of Taxi-FCD. As a result, the central area of Stuttgart is characterized by sufficient Taxi-FCD availability with more than 4,000 measurements per link, whereas for the outskirts only a small number of Taxi-FCD are available (except from trunk roads).

In sum, preprocessing offers insights into empirically collected travel time data. Data records containing suspicious and futile speed measurements as well as data records with deficient time and link entries are filtered. Temporal and spatial distributions of FCD measurements establish hints for subsequent data analysis by DM techniques, for example, a rather high variation of speeds at certain times of the day at different places in the city road network, or an insufficient number of data records in the outskirts. The result of the preprocessing step is a consistent historical travel time database, supporting aggregation of travel time data by DM techniques. The following investigations focus on the core area of the city of Stuttgart, since FCD availability and resulting data quality are sufficient here.

5.3 First Level Aggregation

The transformation of raw empirical traffic data into planning data occurs by first level aggregation. Appropriate aggregation of operational data is crucial for the quality of subsequent planning procedures. The idea is to generate representative travel time estimations for each network link. Therefore, filtered Taxi-FCD records are processed in terms of time-dependent aggregation.

FCD measurements commonly result in a data set with high variation of speeds. Single speed measurements may vary significantly due to manifold perturbations

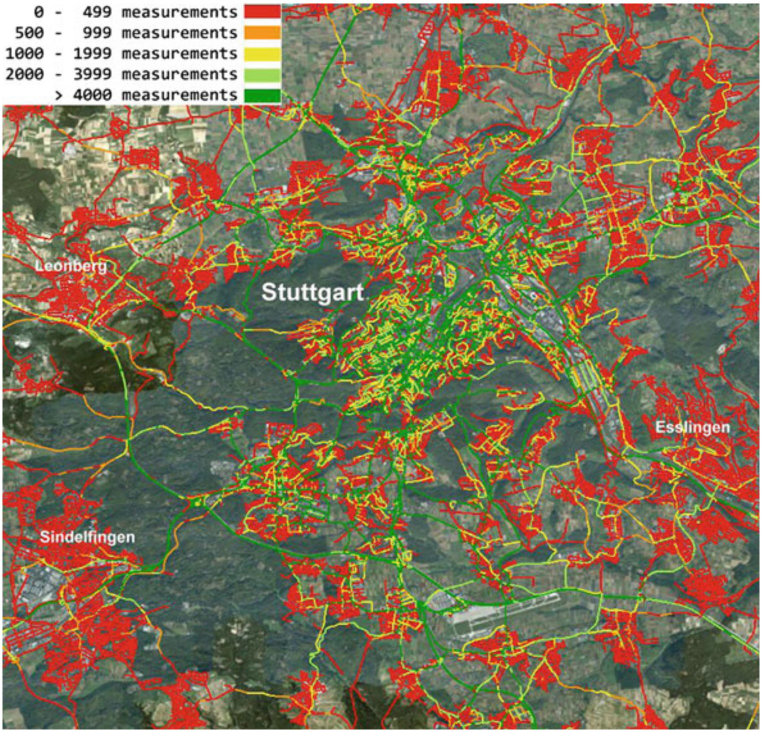


Fig. 5.10 Spatial distribution of FCD measurements (aerial view with kind permission of GeoContent GmbH, Magdeburg, Germany)

in urban traffic networks. Thus, a summary measure is required, featuring a typical speed or travel time (Eglese et al. 2006). To this end, speed estimations are derived for each link l and each time bin b by median calculation, which is supposed to be more robust against outliers than arithmetic mean calculation. The median v_l^b refers to all measurements $v_{l,i}^b$ assigned to a specific time bin b. Seasonal or monthly effects are not investigated here; they are smoothed out due to the large number of data records stretching over several years.

For aggregation, the total number B of time bins must be determined. In the literature, several choices with respect to the setting of B have been made. There is a tradeoff between a smaller number of larger time bins, which might smooth out relevant observations, and a larger number of smaller time bins, which refer to a smaller number of measurements only; the smaller the time bins, the larger the fluctuations. Eglese et al. (2006) use 15 time bins of different lengths per day for time-dependent vehicle routing ($B = 15 \times 7$), whereas Fleischmann et al. (2004) refer to 217 time bins per day ($B = 217 \times 7$). In this work, FCD is aggregated in 24 uniform, 1-hour time bins per day ($B = 24 \times 7$). This is due to the following reasons:

Table 5.2 Typical speeds resulting from median calculation for an example link

Time Day	0–1	1–2	…	4–5	…	8–9	…	16–17	…	22–23	23–24
Sun	46.51	51.78	…	55.04	…	50.11	…	42.07	…	46.61	49.80
Mon	48.71	49.32	…	48.51	…	24.96	…	32.09	…	45.07	47.32
Tue	46.35	49.75	…	49.49	…	25.73	…	28.72	…	46.82	46.20
Wed	47.91	51.08	…	46.81	…	25.51	…	29.81	…	44.93	45.89
Thu	47.94	51.42	…	49.24	…	26.07	…	30.30	…	44.46	46.26
Fri	46.93	52.21	…	51.05	…	32.79	…	31.20	…	43.95	45.95
Sat	46.55	45.79	…	54.26	…	46.16	…	39.79	…	42.59	44.44

- The penetration rate of Taxi-FCD with respect to a 1-hour time bin ensures almost city-wide travel time estimation. Smaller time bins would lead to decreasing data quality and a significant reduction of the possible area of investigation.
- The selection of 24 uniform time bins corresponds to common analysis in the area of traffic research (e.g., Pinkofsky 2006). Results of FCD analysis can be evaluated by traffic experts who are familiar with the depiction of traffic quality in terms of daily curves.
- One-hour time bins ensure the distinction of typical traffic states such as "rush hour" or "free flow" traffic. This allows for the utilization of Taxi-FCD for EDA as well as for the analysis of their impact on routing in city logistics.
- Optimization of delivery tours strongly depends on customer requirements (cf. Chap. 3). Customer time windows offered by retailers commonly have a minimum length of 2 hours. Thus, routing based on 1-hour time bins may be sufficient. Remaining variations of travel times resulting from short-term traffic disturbances should be adjusted by real-time control of delivery tours.
- The following cluster analysis approach demands for a uniform structure of time bins on all links, which may be a drawback since a dynamic structure of time bins could lead to a better representation of typical traffic states. Though, the uniform structure ensures additional degrees of freedom within second level aggregation, resulting in a more effective compression of FCD by cluster analysis.

Speed estimations from first level aggregation are referenced to as *FCD hourly average* (FH). FH represents typical speed fluctuations during 24 h of the day and 7 days of the week. FCD originating from public holidays are treated as Sunday data. A result of aggregation for an example link is shown in Table 5.2, denoting typical speeds for the course of a day and for several days of the week. For a static benchmark, FCD is also aggregated in only one time bin per weekday ($B = 1 \times 7$). This is referenced to as *FCD average* (FA). Both FA and FH figures can be transformed into travel times by consideration of link lengths.

The FH information model induces a comprehensive planning data set, since for each link of the traffic network $24 \times 7 = 168$ travel times arise. This contrasts the efficiency of automated planning procedures. The effective provision of time-dependent travel times expects the reduction of the volume of input data without a significant decrease of reliability. Second level aggregation responds to these requirements.

5.4 Second Level Aggregation

The aim of second level aggregation is to focus on the relevant characteristics of travel time variation to establish a compact, time-dependent information model for routing in city logistics. To this end, cluster analysis is utilized, which produces a compact representation of travel time variation for the links of a city road network. The idea is to group links according to their travel time variation. Planning systems may refer to the representation of a group of links instead of a comprehensive travel time data set for each link. First, preprocessing of FH data is discussed and a target data set for cluster analysis is generated. Then, the clustering tendency of the target data set is evaluated. Several cluster algorithms are selected and discussed regarding their suitability for FCD aggregation. An appropriate number of clusters is determined, and exemplary results are shown.

5.4.1 Preprocessing

Cluster analysis builds on an appropriate target data set. FH data in its pure form is not suited for immediate cluster analysis. The clustering of pure speed data would result in an assignment of links according to their average speeds, for example, slow road sections and fast road sections, because differences between absolute speed values would dominate differences in speed variation. Thus, the idea is to preprocess FH data by normalization by the mean to focus on variation of speeds instead of absolute speeds. For each link, daily curves of speed are transformed into daily curves of relative speed variation, representing the deviation from the links' average speeds. An exemplary data record is shown in Table 5.3, exemplifying the form of the target data set resulting from normalized FH values presented in Table 5.2.

5.4.2 Clustering Tendency

A test on clustering tendency reveals the suitability of the target data set for clustering. In Fig. 5.11, results from the calculation of Hopkin's statistics are reported, denoting a value larger than 0.5 for all days of the week. This indicates a predisposition of the target data set for clustering: the critical bound of 0.5 is exceeded in all cases, but clustering tendency is rather moderate. This is due to

Table 5.3 Typical speeds resulting from median calculation for an example link

Time Day	0–1	1–2	…	4–5	…	8–9	…	16–17	…	22–23	23–24
Sun	0.98	1.09	…	1.16	…	1.06	…	0.89	…	0.98	1.05
Mon	1.17	1.19	…	1.17	…	0.60	…	0.77	…	1.09	1.14
Tue	1.16	1.24	…	1.24	…	0.64	…	0.72	…	1.17	1.15
Wed	1.18	1.26	…	1.16	…	0.63	…	0.74	…	1.11	1.13
Thu	1.20	1.29	…	1.23	…	0.65	…	0.76	…	1.11	1.16
Fri	1.15	1.28	…	1.25	…	0.80	…	0.76	…	1.07	1.12
Sat	1.06	1.16	…	1.20	…	1.02	…	0.88	…	0.94	0.98

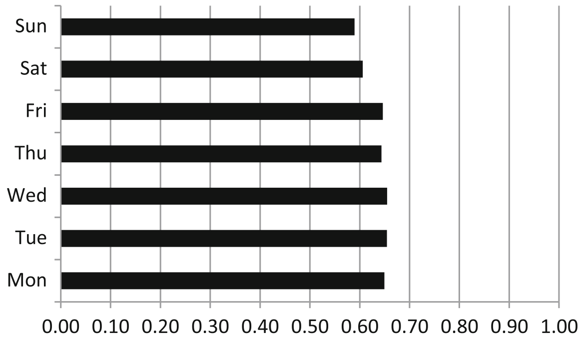


Fig. 5.11 Clustering tendency of normalized FH data (Hopkin's statistics)

Taxi-FCD generally suffering from noise caused by a large variety of perturbations in city road traffic.

5.4.3 Clustering Approach

Resulting clusters must be comprehensible and useful for routing in city logistics. A reasonable clustering approach is selected based on the following parameters:

- The number of clusters should be as small as possible and as large as necessary to conserve precision as well as to feature time-dependent variation.
- Planning procedures suppose a convenient form of travel time representation. Thus, for each cluster, a meaningful cluster representation is required, for example, a representative daily course of speed or speed variation.
- Normalized FH data contains of 24 attributes per data record, featuring a large number of dimensions.

Based on well-known cluster algorithms being available in WEKA, two variants of partitioning based clustering, a hierarchical clustering, as well as a density-based clustering technique are investigated. Partitioning approaches in

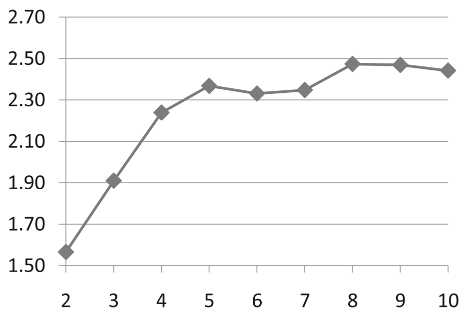


Fig. 5.12 DBI for clustered FH data (typical weekday)

terms of k-Means and k-Medoid lead to comprehensible results; k-Means supplies comparable results in less time. An agglomerative, hierarchical cluster algorithm only works on a heavily reduced version of the target data set, whereas DBSCAN fails due to the high dimensionality. Here, all data objects are assigned to one cluster. A reason may be that the density-based approach is not well suited for data records with high dimensionality due to increasing dispersion of data records.

Thus, the k-Means approach is chosen due to its simplicity and fast and interpretable results in terms of centroids. k-Means is parameterized by a Euclidean distance function. Resulting centroids can be interpreted as daily courses of speed variation, which is a well-known mode of presentation for traffic experts. The centroid can be considered in time-dependent routing procedures (cf. Sect. 6.2).

5.4.4 Number of Clusters

A remaining question is the appropriate number of clusters. For routing in city logistics, the tradeoff is as follows: on the one hand, the number k of desired clusters must be large enough to give a sufficient approximation of the actual link travel times as well as their variation. On the other hand, k should be kept as small as possible to minimize the data input for complex routing procedures and for comprehensible visualization.

Hints for a suitable number of k are provided by the DBI, which is exemplarily depicted with increasing k for data of a typical weekday (cf. Fig. 5.12). Here, a first local minimum can be detected at $k = 6$ (“elbow criterion”), which acts as a starting point for subsequent experiments. Although this number seems to be suitable for routing in city logistics, the quality of travel time precision by such a small number of clusters representing the whole city road network remains open for now. EDA (Sect. 5.5) and computational evaluation by simulation (Chap. 7) give detailed insights into the reliability of data sets, as well as into the resulting service quality of delivery tours.

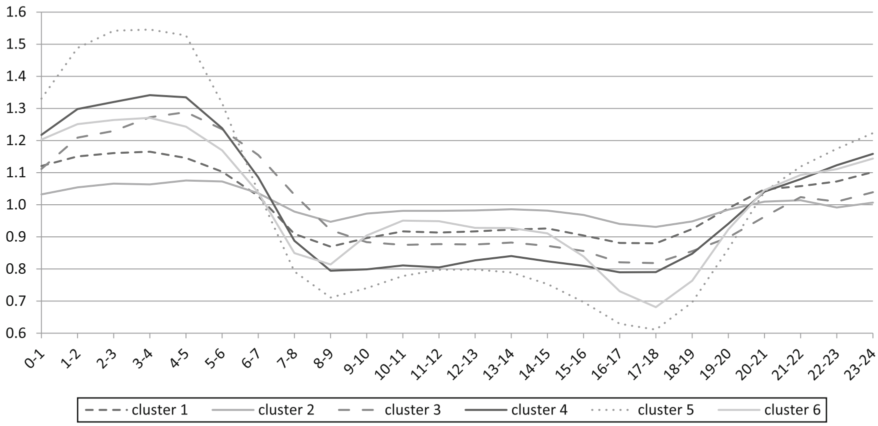


Fig. 5.13 Example of a clustering result (weekday, k = 6)

An example result of clustering by k-Means with $k = 6$ is shown in Fig. 5.13. Here, the cluster centroids are depicted in terms of their speed variations in the course of a typical weekday. Each cluster centroid represents a group of links by 24 *speed reduction factors*. Generally, city traffic patterns such as decreasing speeds during morning and evening rush hours can be identified at several levels of intensity. Whereas links assigned to Cluster 5 show relatively high speeds during the night (more than 1.5 $\times$ average speed) and relatively low speeds in morning and evening rush hours (fewer than 0.7 $\times$ average speed), links assigned to Cluster 2 feature only small variations. The derivation of a link specific, time-dependent speed is as follows: since each link is associated with its groups' vector of 24 speed reduction factors, a time-dependent speed can be derived by weighting FA data with the corresponding speed reduction factor.

In sum, second level aggregation produces a compact, time-dependent information model, which is referenced to as *Floating Car Weighted Average* (FW) in the following. Key points are the normalization of FH data to determine speed variation instead of absolute speeds, accompanied by the clustering of links according to their speed variation. This leads to a compact representation of time-dependent travel times.

5.5 Exploratory Data Analysis

FCD analysis is based on the hypothesis that typical traffic states may be represented by FH and FW information models. To ensure that results of first and second level aggregation are reasonable, EDA is utilized for verification. In particular, FH and FW information models are investigated in terms of the expected temporal and spatial variation of travel times. Travel times generated by first and second level aggregation are supposed to depict time-dependent variation of traffic quality. For instance, a more or less "free traffic flow" with shorter travel times is expected at night, contrasting supposed longer travel times during "rush hours."

EDA is conducted in relation with expert interviews, i.e., discussions with representatives managing the traffic information system in the city of Stuttgart. On the one hand, Google Earth produces high-resolution satellite pictures, which allow for the investigation of hypotheses on variances of travel times with respect to the links' geographical locations. On the other hand, the evaluation of links by a level of service concept examines assumptions on expected traffic quality.

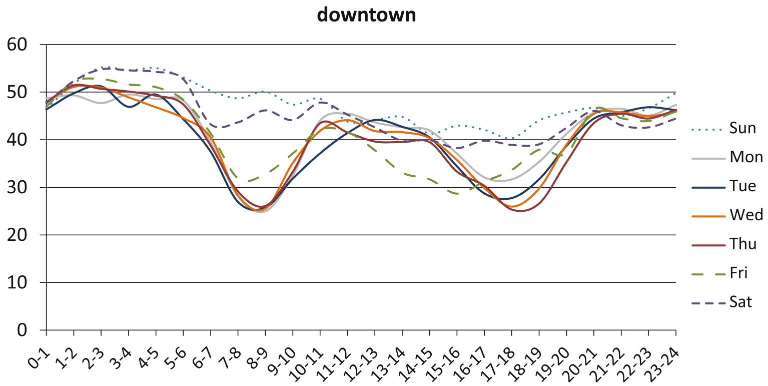
5.5.1 First Level Aggregation

Two techniques of EDA for the evaluation of FH data are presented in this section. First, daily curves are examined, which corresponds to a well-known method of illustration in the area of traffic research (Pinkofsky 2006). Speed evolution can be investigated for a specific link or for the whole area of investigation. Second, a speed evaluation scheme facilitates the area-wide investigation of speed data, supporting the temporal analysis of traffic quality evolution.

Daily curves give insights into the temporal evolution of speeds for single road sections. Figure 5.14 exemplifies weekday-dependent speed curves depicting the variation of average speeds along the network links of the road "Cannstatter Straße." The inbound lane is characterized by sharply decreasing speeds in morning and afternoon rush hours on working days. Evolution on Fridays is slightly different; on weekends, no sharp falls can be noticed. The outbound lane features an average speed of about 40 km/h, decreasing sharply on all working days except Fridays. On weekends, no late rush hour can be observed.

An area-wide overview of traffic quality evolution is facilitated by a six-step evaluation scheme adapted from Busch et al. (2004). Here, traffic quality is derived from the ratio of the "current" speed and the "ideal" speed of a network link for every hour of a weekday. The ratio is parameterized as follows: the link specific daily maximum of its 24 FH values defines the *ideal* speed, whereas the current time bin's FH value corresponds to the *current* speed. The evaluation of this ratio leads to a link-specific traffic quality in terms of traffic quality levels 'A' (very good) to 'F' (heavily congested).

The result of the evaluation is depicted in Fig. 5.15. At night, more than 80% of the network links are evaluated with levels A and B, illustrating a "free flow" network. In contrast to night hours, traffic quality generally decreases in rush hours (e.g., 8:00–9:00 and 17:00–18:00). Here, only 30% of the links feature a traffic quality of A or B, whereas 28% of the network links suffer from relatively long average travel times implying "bad" or "very bad" traffic quality (quality levels



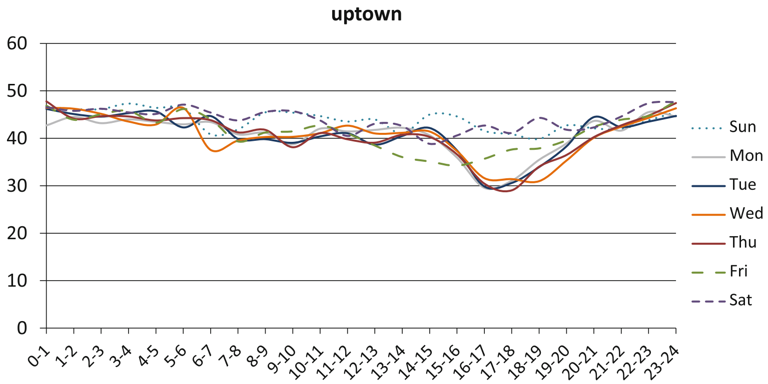


Fig. 5.14 Daily curves depicting average speeds along the road 'Cannstatter Straße'

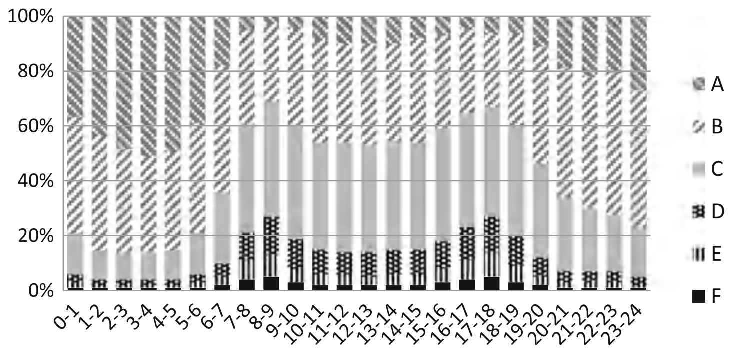


Fig. 5.15 Share of traffic quality levels in the course of a typical weekday

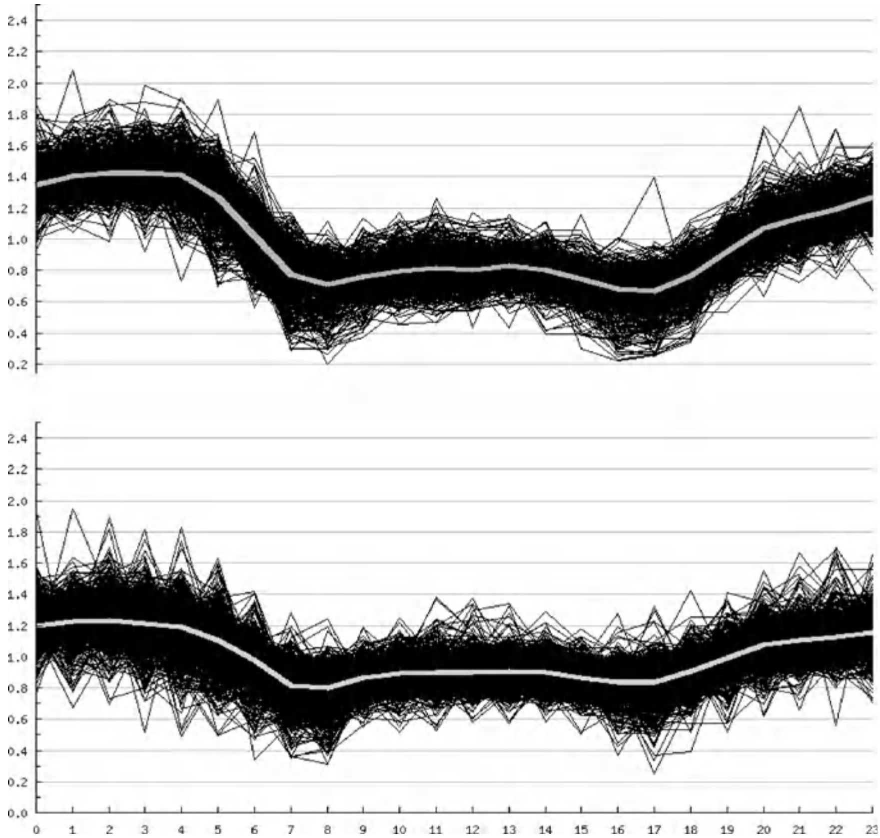


Fig. 5.16 Exemplary results of clustering with k-Means ($k = 6$, weekday), temporal view

E and F). For a detailed spatial examination of traffic quality and typical traffic states, certain time slots may be visualized by Google Earth.

Altogether, well-known variations of traffic quality can be observed from FH data. Evaluated travel times are likely feasible and interpretable. They offer an aggregated view on the evolution of traffic quality in the course of the day.

5.5.2 Second Level Aggregation

Daily courses derived from FH information models are reasonable for the representation of traffic quality. Nonetheless, efficient routing demands for a heavily reduced amount of input data. This can be achieved by more efficient, but also more complex information models. At this point, FW represents daily courses of

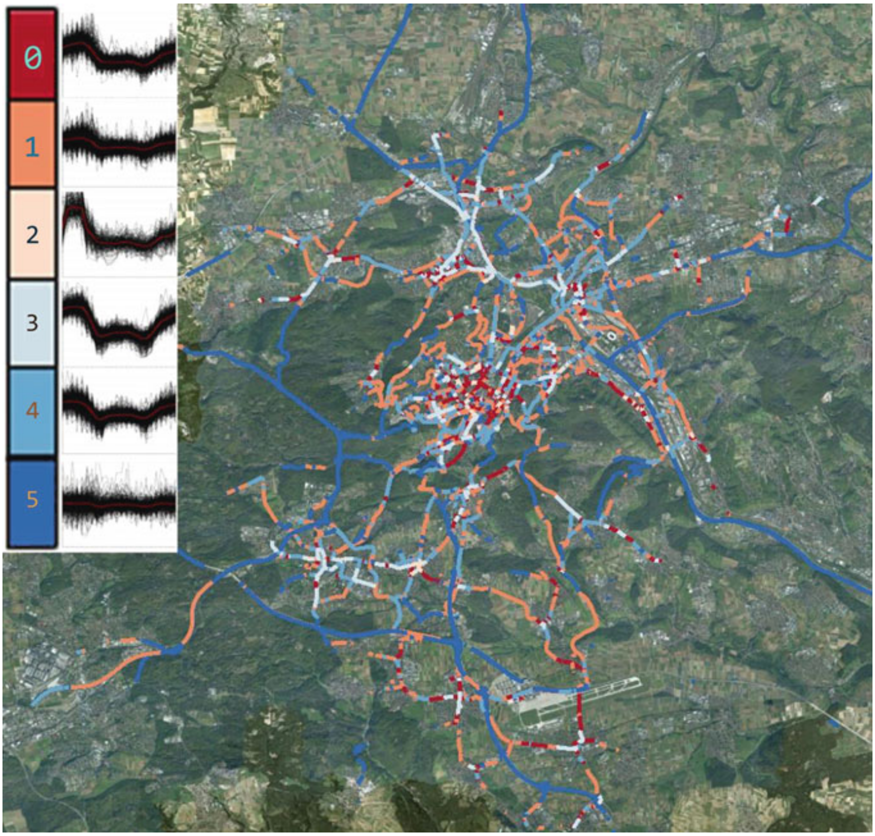


Fig. 5.17 Overall results of clustering with k-Means ($k = 6$, weekday), spatial view (aerial view with kind permission by GeoContent GmbH, Magdeburg, Germany)

speed variation. In the following, exemplary results of clustering with k-Means ($k = 6$) are evaluated temporally and spatially.

The temporal view allows for insights into the clustering result as well as into underlying curves of speed variation. In Fig. 5.16, two (of six) exemplary clusters are depicted. For each cluster, the representing centroid (gray colored) as well as the underlying data records of assigned links are shown, which correspond to normalized FH data. Due to median-based aggregation, cluster cohesion seems to be sufficient. Morning and evening rush hours are clearly visible. The upper cluster features variations of speeds in the range of approximately 70–140%, depending on the time of day. The lower cluster denotes variation of speeds in a smaller range, i.e., between 80 and 120% relative to individual link speeds.

Google Earth features the spatial visualization of all six clusters by an aerial view, depicting links in the color of the assigned cluster. Figure 5.17 shows a very homogeneous clustering where coherent parts of the road network are clearly noticeable. Links assigned to Clusters 2, 3, and 5 mainly correspond to freeways and trunk roads, whereas links of Clusters 0 and 1 comprise inner-city roads with rather high or low variation of speeds, respectively. Links assigned to Cluster 4 connect outskirts with inner city areas. They suffer from high variation of travel times due to rush hour traffic.

Google Earth pictures have been used in expert interviews for discussions on the clusters' spreading as well as the number of clusters and the usefulness of the cluster data set; they comprise an abstract aggregation of the travel time evolution in the road network of Stuttgart. Qualitative evaluation of the FW data set is amended by quantitative evaluation in terms of comprehensive routing experiments in Chaps. 7 and 9.

All in all, analysis of a large data set of Taxi-FCD has prepared information models that represent the typical behavior of a city road network with different granularity complexity:

- FA lead to day-specific travel times resulting from aggregating FCD to one average measure per link and day of the week (first level aggregation).
- FH refer to travel times resulting from aggregating FCD to 24 averages per link, dependent on day of the week and time of the day (first level aggregation).
- FW denote travel times resulting from speed reduction factors combined with FA data (second level aggregation).

In the following chapter, processing of the different information models for integration into automated planning procedures is discussed.

Chapter 6
Provision of Distance Matrices

State-of-the-art planning systems for routing in logistics allow for automatic processing of customer orders, considering a large variety of constraints and operational conditions (Fleischmann and Gietz 2002). As a core feature, they support automated planning by optimization procedures. Planning systems rely on tailored input data, which are commonly assumed to be available as required by optimization procedures. If input data sets become more complex, efficiency of optimization procedures decreases or optimization procedures are not even able to consider such data, respectively. The provision of input data according to the requirements of optimization procedures is usually neglected

This part deals with the integration of information models into advanced planning systems for routing in city logistics. To this end, information about the typical behavior of the city road network is incorporated (Chap. 6). This information results from (time-dependent) information models providing (time-dependent) travel times. Time-dependent travel times are evaluated with respect to resulting planning quality and reliability within shortest path computation (Chap. 7).

The preparation of information models according to the requirements of optimization models is focused in this chapter. Optimization models are the core subject of OR methodology. Figure 6.1 depicts the way OR works: comparable to the DM process model discussed in Chap. 4, OR expects a hypothesis regarding the structure of a system to determine optimal decisions. The preparation of an optimization model is based on appearances of the system, i.e., data records describing system behavior, and an optimization model structure supporting the analysis of the hypothesis. The definition of an optimization model allows automated search procedures to determine the optimal decision with respect to given input data. System behavior may change after the implementation of the optimal decision.

OR commonly focuses on optimization models and the improvement of search algorithms in order to efficiently compute optimal or near-optimal decisions. The

J. F. Ehmke, *Integration of Information and Optimization Models for Routing in City Logistics*, International Series in Operations Research & Management Science 177,
DOI: 10.1007/978-1-4614-3628-7_6,

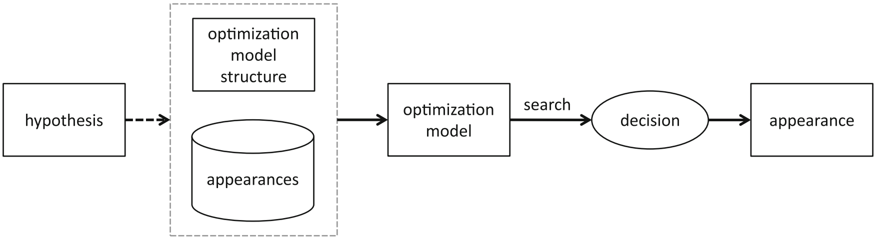


Fig. 6.1 OR process model (adapted from Meisel and Mattfeld 2010)

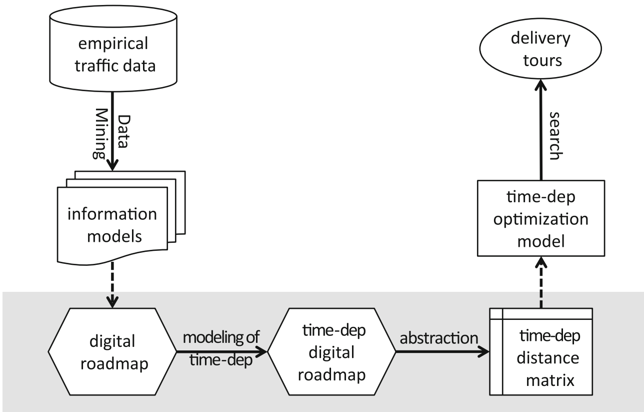


Fig. 6.2 Integration of information models

effective preparation of input data for more efficient search methods is usually neglected. At this point, information models come into play. Information models may "increase effectiveness of OR by refinement" (Meisel and Mattfeld 2010), i.e., a more appropriate representation of system appearances may lead to improvements of optimization models. In the following, time-dependent information models are integrated into time-dependent optimization models for routing in city logistics.

In Fig. 6.2, the integration of information models and optimization models is depicted for routing in city logistics. Analysis of empirical traffic data by DM leads to information models at different levels of aggregation (cf. Chap. 5). In this form, information models are not suited for optimization models. Required integration steps are underlined by grey shadowing. In particular, a common *digital roadmap* representing typical distances and travel times is enhanced by data arising from information models. The integration of time-dependent information models expects a consistent representation of the digital roadmap.

The corresponding *time-dependent digital roadmap* contains much more information than suitable for time-dependent optimization models. Time-dependent distance matrices serve as a problem-specific abstraction of the time-dependent digital roadmap, representing customers, depots, and their spatio-temporal relations. Time-dependent optimization models build on time-dependent distance matrices to determine cost-efficient delivery tours. Search algorithms have to be adapted to more complex information models (cf. Chap. 8).

The information models presented above differ in volume and complexity. They can be classified as follows (Toledo and Beinhaker 2006):

- *Static routing information* refers to distances and typical travel times, for example, average speeds or legal speeds. RT and FA represent corresponding information models. They do not allow for capturing of congestion or varying patterns of traffic demand. This kind of information is already contained in common digital roadmaps, describing core features of the road network. They are the standard input for the calculation of distance matrices. The role and provision of static routing information is subject of Sect. 6.1.
- *Historic routing information* describes typical traffic network behavior based on a database of historical travel times. Recurring congestion and traffic demand patterns are captured by consideration of time dependence. FH and FW information models correspond to this kind of routing information. Historic routing information results in a more comprehensive representation of the city road network, requiring a time-dependent variant of a digital roadmap. The modeling and integration of comprehensive and compact historic routing information is subject of Sect. 6.2.
- *Instantaneous routing information* relies on historical travel times, which are amended by real-time estimates of travel times provided by ATIS (cf. Sect. 5.1.2). *Predictive routing information* is based on a prediction model that allows for the (short-term) prediction of future travel times. Both types of information are required for the real-time management of city logistics service providers, which is beyond the purpose of this work.

In the following, the particular steps of the integration are presented. City logistics routing is commonly based on a static representation of the city road network in terms of a digital roadmap (cf. Sect. 6.1.1), providing static routing information by typical distances and travel times (cf. Sect. 6.1.2). The consideration of historic routing information leads to conceptual challenges with regard to the representation of the road network (cf. Sect. 6.2.1) as well as for optimization models, which are usually not suited for the incorporation of time-dependent distance matrices (cf. Sect. 6.2.2). Here, a suitable representation of time dependence is required, aligning time-dependent information models with optimization models. Calculation of (time-dependent) distance matrices occurs by (time-dependent) shortest path computation (cf. Sect. 6.3).

6.1 Static Information Models

The majority of today's optimization models for routing in city logistics relies on static routing information, which is provided by static information models in terms of average travel times or distances. Based on a digital representation of the road network, shortest path computation transforms this information into static distance matrices, which are considered in the optimization of delivery tours. In this section, the architecture of a common digital roadmap and its implementation for FCD collected in the area of Stuttgart are presented.

6.1.1 Digital Roadmap

A digital roadmap facilitates the digital representation of the city road network. Digital roadmaps consist of data models which are able to encode, store, retrieve, modify, analyze, and display transportation networks, especially for GIS. They are commonly based on a vector-oriented model of the road network in terms of links and nodes, allowing for the utilization of methods from the area of graph theory. Graph theory gives a "topological and mathematical representation of the nature and structure of transportation networks" (Rodrigue et al. 2009).

According to Rodrigue et al. (2009), digital roadmaps are the foundation for a number of applications in traffic and transportation:

- The main purpose of a digital roadmap is to establish an *accurate representation of a road network* as a set of links and nodes. Whereas graph theory aims at the abstraction of transportation networks, a digital roadmap should be as close as possible to the real world structure it represents.
- A digital roadmap supports the *visualization of a road network*, for example, for the purpose of navigation.
- An important feature is *geocoding*, which assigns a real-world entity to an object of the digital roadmap. For instance, digital roadmaps are used to derive a precise location of customer locations with respect to their position in the road network.
- Digital roadmaps may be used to *find optimal paths* and to assign flows with capacity constraints in a road network, supporting individual routing as well as traffic management. For routing in city logistics, information on traffic lights, congestion, etc. must be taken into account.

Different applications of a digital roadmap for routing in city logistics are exemplified in Fig. 6.3. On the left, an exemplary road network is presented as a model of the transportation network by elements of graph theory, featuring an abstract representation in terms of links and nodes ("topology"). An extract of information being relevant for routing of a delivery tour is shown on the right, denoting warehouses as well as pickup and delivery locations. In the following,

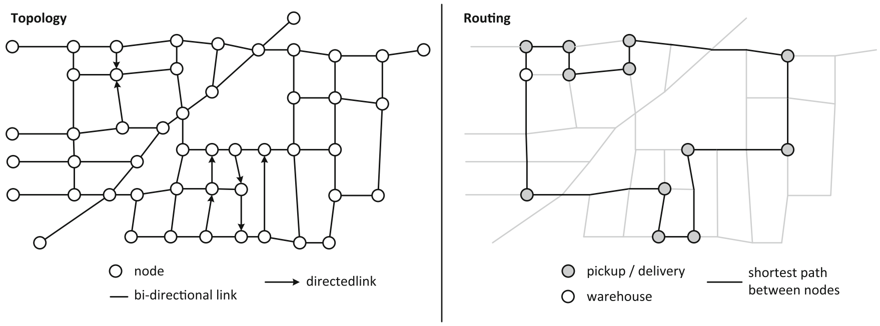


Fig. 6.3 Different applications of digital roadmaps (adapted from Rodrigue et al. 2009)

the focus is on the (time-dependent) representation of the topology, which is a fundamental requirement for routing in city logistics. Shortest path computation transforms the detailed information about a road network topology into input data for routing procedures.

6.1.1.1 Representation of a Road Network

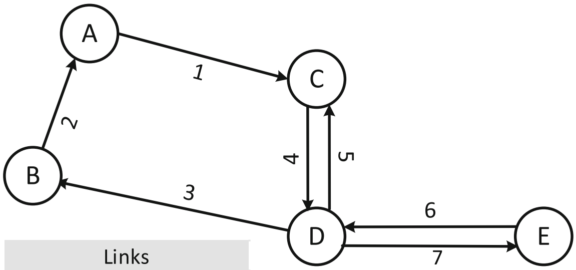
The digital representation of a road network works as follows:

- A *node table* contains data concerning intersections or other points of interests within the road network, for example, points where a speed limit changes. This table consists of at least three attributes: one to store a unique identifier and the others to store a node's latitude and longitude to facilitate the geographical allocation of nodes.
- A *link table* contains a field with a unique identifier, one to store the node of the origin, and the other to store the node of the destination.

Once node and link table are relationally linked, a basic network topology can be constructed, providing input for algorithms of graph theory. Both the nodes and links tables have little value if they are considered individually, as a network arises from the combination of the data contained in both tables. An example road network represented by nodes and links and its corresponding representation in a database is shown in Fig. 6.4. Here, nodes A to E denote geographical information about intersections of the road network. Links 1 to 7 represent road infrastructure and connect intersections.

In order to ensure a realistic representation of the road network geometry, each link can be fragmented into a multitude of segments. Additional information can be gathered from further attributes being assigned to links, for example, lengths, speed limits, road names, road categories, one-ways, and prohibition of turns. Rodrigue et al. (2009) give an overview on relevant attributes. Digital roadmaps can be purchased by a number of vendors such as NAVTEQ or TeleAtlas.

Fig. 6.4 Relational database representation and the corresponding graph of a simple road network (adapted from Rodrigue et al. 2009)



Links		
ID	From	To
1	A	C
2	B	A
3	D	B
4	C	D
5	D	C
6	E	D
7	D	E

Nodes		
ID	Latitude	Longitude
A	40.42345	-75.12450
B	40.31021	-75.25100
C	40.41882	-74.91240
D	40.25908	-75.00310
E	40.28990	-74.78930

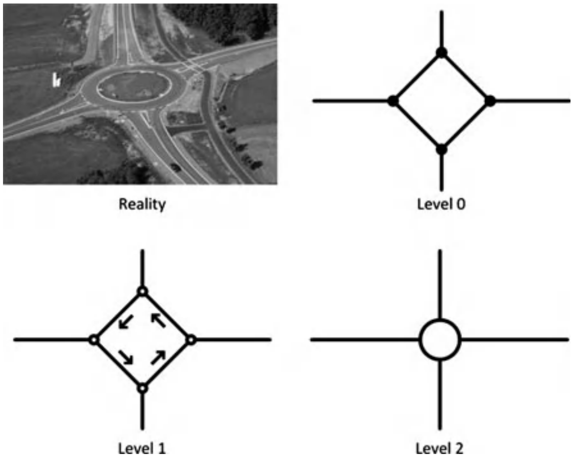
The reference manual on NAVTEQ's NAVSTREETS street data illustrates the variety of recent digital roadmaps' features *(NAVTEQ's NAVSTREETS Street Data Reference Manual v2.0* 2006).

6.1.1.2 Geographic Data File Standard

Interoperability of digital roadmaps by different map vendors is facilitated by the ISO Geographic Data File standard (GDF). The GDF is used to describe and exchange road network related data, comprising a conceptual data model which represents features, attributes and relationships by different "catalogues" (International Organization for Standardization 2009). The *Feature Catalogue* defines real-world objects such as infrastructure and buildings. The *Attribute Catalogue* contains typical characteristics of real-world objects, whereas the *Relationship Catalogue* describes relations between features (e.g., to indicate the right-of way).

Supporting requirements of different applications, GDF features three levels of abstraction. In Level 0, the fundamental geometrical and topological entities are described. Those entities are nodes (0-dimensional), links or polylines (1-dimensional) and faces or polygons (2-dimensional). Level 1 allows for the description of real world geographic objects by their characterizing features. The simple features of Level 1 refer to Level 0 entities as their geometrical and topological representation, and combine them with attributes and relationships (e.g., intersections, road segments, and address areas). The features are characterized by attributes such as the number of lanes or the permissible direction of travel. Within Level 2, complex features consisting of aggregated simple features are described (e.g., traffic circles and highway intersections).

Fig. 6.5 GDF levels



In Fig. 6.5, the level of detail of a digital roadmap is exemplified with respect to different GDF levels. In the upper left side, an aerial view of the road infrastructure can be seen, denoting a traffic circle ("reality"). In the upper right side, the representation of the traffic circle by simple elements is shown (Level 0). In Level 1, these basic entities are enhanced by attributes and relationships in terms of one-way information. In Level 2, the traffic circle itself is a complex feature, aggregated by individual one-way links of Level 1. Planning systems for routing in city logistics require a digital roadmap of Level 1 in order to facilitate a well-defined topology of the city road network.

In the following, the implementation of a static digital roadmap is described, featuring static information in terms of distances and average travel times.

6.1.2 Implementation

For routing in city logistics, a digital roadmap represents information about the topology of the road network as well as about the typical behavior of the road network. Implying a relational representation of links and nodes, the following information models are raised to facilitate static routing information:

- *Roadmap distance* (RD) refers to the length of a link. Link lengths are required to derive travel times from information models representing speeds. They are available in common digital roadmaps by default. RD is often used for the optimization of delivery tours in wide area networks. This is not appropriate for routing in urban areas, though, since temporal impacts play a more important role here.

- *Roadmap travel time* (RT) refers to speeds associated with the links of the digital roadmap. Speed estimations are usually derived from a classification of the road network. In urban areas, travel time estimations based on RT are commonly rough estimations and regularly lead to underestimations of travel times, because waiting times at traffic signals as well as decreasing speeds due to congestion are not considered.
- *FCD averages* (FA) as provided by first level aggregation resulting from FCD being aggregated to one average travel time estimate per link and day of the week (cf. Sect. 5.3). Arising from empirical data collection, FCD aggregates represent the various perturbations on urban traffic flows such as waiting times at traffic signals, congestion, etc. FA may result in more realistic travel time estimations compared to RT. Though, FA is not capable of depicting varying travel times throughout the day.

For subsequent experiments, a common digital roadmap of the area of Stuttgart is implemented. The digital roadmap refers to an area of about 35 km^2 around the city of Stuttgart and consists of 100,446 nodes and 128,529 links, respectively. A node and a link table as well as RD and RT data are available by default. For visualization purposes, geographical coordinates are provided, which divide a link into up to 80 virtual segments. Additionally, FA data is generated as described in Sect. 5.3 and is incorporated by extension of the link table.

Static information models supply distance matrices as input for static optimization models. Therefore, travel times between a set of source nodes S and a set of target nodes A are required. A common approach is to compute a distance matrix of size $|S| \times |A|$, reducing subsequent travel time computations to a simple table lookup (Geisberger and Sanders 2010). Corresponding shortest path functionality for static information models is described in Sect. 6.3.1. However, static information models are not capable of capturing congestion. This is reserved for time-dependent information models.

6.2 Time-Dependent Information Models

Contrasting static information models, time-dependent information models feature capturing of time-dependent variations of travel times. Time-dependent information models expect the appropriate modeling of time dependence within a digital roadmap, since complexity of a time-dependent topology may limit efficiency of optimization procedures. Besides, consideration of time dependence may cause logical contradictions in the computation of shortest paths. In order to ensure efficient as well as methodologically consistent optimization, several alternatives of the modeling of time dependence are discussed (cf. Sect. 6.2.1). The implementation of the variant which is suitable for FCD based information models is presented in Sect. 6.2.2.

6.2.1 Modeling of Time Dependence

Time-dependent optimization asks for an adequate representation of time-varying travel times for each network link. Whereas in static networks typical travel times are represented by a single scalar or a combination of scalars per link, the consideration of a time dimension induces enhancements of digital roadmaps' data structures. In the following, several approaches for the modeling of time dependence are presented. The idea is to define a suitable data model for the computation of time-dependent shortest paths. This data model is instantiated by time-dependent information models.

Since the modeling of a digital roadmap determines the efficiency of optimization in transportation networks, switching from a static to a time-dependent network representation is challenging. According to George et al. (2007) and Delling and Wagner (2009), the design of time-dependent data models and optimization procedures may differ significantly from their static counterparts:

- Data input for optimization models increases significantly, as travel times on time-dependent connections may change frequently during the day. FH and FW information models establish comprehensive data input for time-dependent optimization.
- Due to its potentially large size, the storage-efficient representation of time-dependent data is crucial in order to eliminate redundant information across different time bins. FW corresponds to these requirements, offering a heavily compressed data set.
- If the network topology is not defined conveniently, subsequent optimization may become inconsistent and/or computationally expensive. Efficient and correct query processing strategies and algorithms must be defined, since some of the commonly assumed graph properties may not hold for spatio-temporal networks. The FIFO property introduced below ensures a suitable and convenient network structure.
- New (and more complex) data models need to be investigated in order to represent and classify potentially alternative semantics for common graph operations such as shortest paths computation. Two concepts are introduced below.
- Conventional graph algorithms cannot be easily adapted to "snapshot graphs" at discrete time instants without accounting for relationships among snapshots. Speedup techniques for static optimization procedures (e.g., bidirectional search) do not work and have to be adapted to the time-dependent setting. This is discussed in Chap. 8 in detail.

In the following, the focus is on a time-dependent topology of the city road network facilitating the determination of time-dependent distance matrices by shortest path computation.

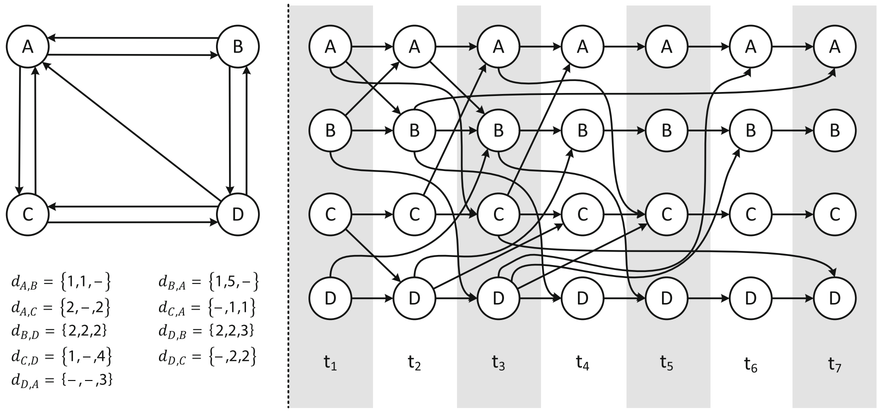


Fig. 6.6 Space–time network representation (adapted from Pallottino and Scutella 1997)

6.2.1.1 Time-Dependent Graphs

The consideration of a time dimension in network modeling occurs either by duplicating a static network for each required time instance or by introducing more complex data models for the representation of time-varying travel times. Pallottino and Scutella (1997) elaborate on the former. Given a graph $G = (V, E)$, they define an "edge delay function" $d_{i,j}(t)$ which is associated with each edge (i, j) with the following meaning: if t is the non-negative leaving time from node i, then $t + d_{i,j}(t)$ is the arrival time at node j. The time horizon is discretized, i.e., t can vary in the discrete set $T = \{t_1, t_2, \cdots, t_q\}$, leading to q different snapshots of the road network. The resulting *space–time network* $R = (V, E)$ is defined as follows:

$$V = \{i_h | 1 \leq h \leq q\}$$
$$E = \left\{(i_h, j_k) : (i, j) \in V,\ t_h + d_{i,j}(t_h) = t_k, 1 \leq h < k \leq q\right\}$$

An example graph and the resulting space–time network are given in Fig. 6.6. The example graph consists of 4 nodes, 9 edges and a time series with a length of $q = 7$. For each edge, time-varying travel times are denoted. If there is no travel time given for a specific time bin, the edge is not available in the corresponding snapshot. The network is duplicated for each time bin arising from the travel time series. Time-dependent travel times are denoted in brackets for each edge, leading to the space–time network on the right hand side. Arrows denote the transition of one or more time bins. Travel times are represented implicitly by the linking of different time bins' nodes. For (A, B), a direct connection of time bin t_1 and t_2 results from $d_{A,B}(1) = 1$. For (A, C), t_2 is skipped since $d_{A,C}(1) = 2$, for example. In sum, the space–time network consists of 28 nodes and 43 edges.

Space–time networks increase the size of the original network significantly. They are very expensive with respect to memory requirements, which can be estimated as $O\ (nT) + O(n + mT)$, where n is the number of nodes, m is the

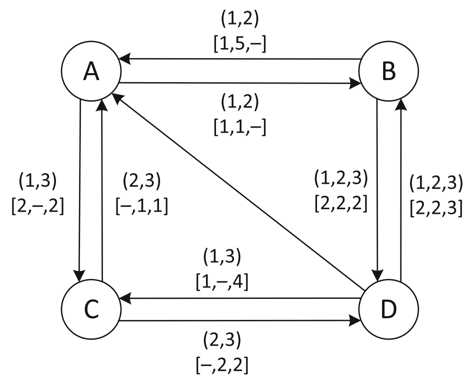


Fig. 6.7 Time-aggregated graph representation

number of edges in the original graph, and T is the length of the travel time series. Nonetheless, space–time networks have been used in a number of OR publications (e.g., Dreyfus 1969; Orda and Rom 1991; Kaufman and Smith 1993; Köhler et al. 2002; Lu et al. 2005). For the representation of a city road network of realistic size, space–time networks are only suitable to a very limited extent.

The extension of data structures for a more compact representation of time dependence is discussed by George et al. (2007). They sum up recent research dealing with spatio-temporal networks and routing algorithms from a database point of view. Contrasting space–time networks, a *time-aggregated graph* is introduced, which more efficiently represents time-varying edge weights. Time-aggregated graphs require only $O(n + m)T$, implying an adjacency list implementation.

In Fig. 6.7, a time-aggregated representation of the graph of Fig. 6.6 is shown. For each edge, a travel time series is associated instead of a scalar, contrasting static graphs and space–time networks. Travel time series are given in squared brackets, whereas the availability of a link in individual time bins is denoted by round brackets. Computational experiments show the advantages of algorithms based on the time-aggregated graph representation in contrast to space–time networks. However, the usage of time-aggregated graphs expects adaptions of well-known shortest path algorithms with respect to the more complex data structure of edge weights.

6.2.1.2 Consistency of Time-Dependent Graphs

The inherent discretization of the time horizon in time-dependent graphs may lead to inconsistencies in the network structure with regard to the transition of particular time bins. “Passing” may occur, i.e., a potential delay at a node may lead to shorter travel times in forthcoming time bins, which may be illogically advantageous in terms of overall travel time. *FIFO networks* ensure that the structure of time bins and edge weights prevents such inconsistencies. In a FIFO network “commodities travel along each arc in a First-In-First-Out manner” (Dean 2004b). Here, vehicles are not able to “pass” each other, i.e., vehicles arrive in the order they commence an edge, which is an important prerequisite for the adaptation of

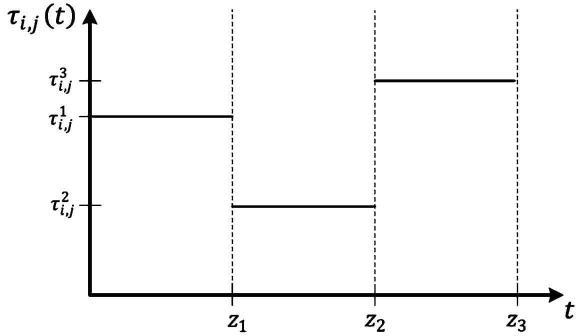


Fig. 6.8 Time-dependent travel time function for an example link

well-known shortest path algorithms. In the literature, this behavior is also known as "non-passing condition" (Kaufman and Smith 1993; Ichoua et al. 2003).

A FIFO network can be defined as follows: let $G = (V,E)$ be a directed graph, and $a_{i,j}(t)$ an *arrival time function* for each link $(i, j) \in E$. $a_{i,j}(t)$ denotes the arrival time at j if one departs at time t. The function $\tau_{i,j}(t) = a_{i,j}(t) - t$ gives the *travel time* along link (i, j) if one departs at time t. If $a_{i,j}(t)$ is non-decreasing for all $(i, j) \in E$, G is a FIFO network. $\tau_{i,j}(t)$ is usually defined as piecewise-linear function based on a discrete set of time bins. For a literature overview on the continuous case, see Ichoua et al. (2003).

FIFO networks feature structural properties that enable the development of efficient solution algorithms (Dean 2004b):

- In FIFO networks, waiting at nodes is never beneficial, i.e., it will never reduce the arrival time at the destination. Thus, waiting makes no sense in FIFO networks, which reduces the complexity of search procedures in comparison with non-FIFO networks.
- In FIFO networks, one may find shortest paths which are acyclic.
- In FIFO networks, one may find shortest paths whose sub paths are also shortest paths.

FH and FW information models lead to piecewise-linear travel time functions, not necessarily complying with conditions of FIFO networks. In particular, the FIFO property may be violated if a rather long travel time is followed by a rather short travel time. Then, the travel time function jumps between the two time bins, and passing may occur. In Fig. 6.8, such a situation is depicted. The travel time function $\tau_{i,j}(t)$ is derived from FH or FW data and corresponding link lengths. It jumps at times z_1, z_2 and z_3, which denote start and endpoints of individual time bins. Travel times evolve from a rather long travel time at the level of $\tau^1_{i,j}$ to a rather short travel time of $\tau^2_{i,j}$, followed by a long travel time at the level of $\tau^3_{i,j}$. Hence, at z_1, the travel time decreases significantly. This transition is not FIFO consistent; a vehicle starting shortly before z_1 would be passed by a vehicle starting shortly after z_1. Due to optimality conditions not being fulfilled in transitions between the particular time bins, this may lead to inconsistencies in the calculation of shortest paths, where the determination of the shortest path would require an artificial waiting time at some nodes.

6.2.1.3 Generation of FIFO Networks

In the literature, several approaches for the transformation of non-FIFO networks into FIFO networks have been discussed. Ichoua et al. (2003) perform simple calculations to adjust the travel time between two nodes of a time-dependent network. They show that the resulting travel time function satisfies the FIFO property. Eglese et al. (2006) take up on their idea and construct a time-dependent digital roadmap for a wide area network by discretizing the planning horizon to 15 time bins of different length per day. They adjust the time of arrival at a node if and only if the FIFO property might fail. To this end, the speed is artificially increased after transition of a time bin boundary in order to avoid passing.

The corresponding algorithm is shown in `Algorithm 3`, maintaining the FIFO consistent computation of the arrival time a_j at node j considering a departure time T_i at node i on a link (i,j). Following the style of Fig. 6.8, the time-dependent travel time from i to j in time bin k is indicated as $\tau_{i,j}^{k}$. $z_k < z_{k+1}$ denote the boundaries of time bin k, and $d_{i,j}$ denotes the length of the link (i,j). The algorithm works as follows. First, it is checked if a departure at T_i might pass over the boundary z_{k+1} of the current time bin k. If so, the travel time is adjusted by increasing the speed; if not, the arrival time a_j simply corresponds to departure time T_i plus travel time $\tau_{i,j}^{k}$ The adjustment occurs by calculation of the distance d that remains when driving at speed v_k between departure time and the boundary of the time bin. Due to v_k being relatively low, not the entire distance of (i,j) is traveled until the point of transition. Hence, the time required for traversing the residual of (i,j) with increased speed v_{k+1} is calculated, resulting in the adjusted arrival time a_j'.

Algorithm 3: Maintaining the FIFO property	
Input:	T_i departure time at node i in time bin k, $\tau_{i,j}^{k}$ travel time between node i and j in time bin k, $d_{i,j}$ length of link (i,j)
Processing:	**If** $T_i + \tau_{i,j}^{k} \le z_{k+1}$ then $a_j = T_i + \tau_{i,j}^{k}$ **else** $a_j' = T_i + \tau_{i,j}^{k},\ d = d_{i,j},\ t = T_i,\ v_k := d_{i,j} \big/ \tau_{i,j}^{k}$ **While** $\left(a_j' > z_{k+1}\right)$ $d = d - v_k \times (z_{k+1} - t)$ $t = z_{k+1}$ $v_{k+1} = d_{i,j} \big/ \tau_{i,j}^{k+1}$ $a_j' = t + (d/v_{k+1})$ $k = k + 1$ $a_j = a_j'$

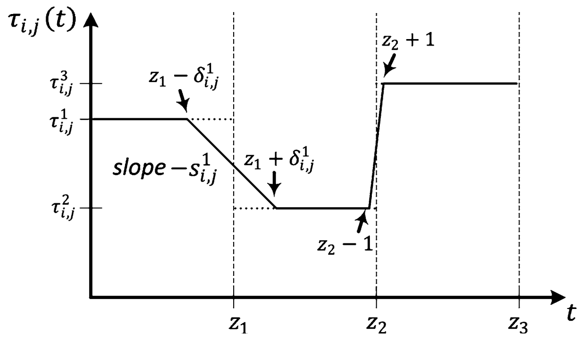


Fig. 6.9 Derivation of a FIFO consistent travel time function

For example, suppose a link (i, j) with a length of $d_{i,j} = 2$ km and two time bins, one from 8:00 to 9:00 with $\tau^1_{i,j} = 4$ min, and one from 9:00 to 10:00 with $\tau^2_{i,j} = 2$ min. If the departure time $T_i \leq$ 8:56, the arrival time is simply $T_i + 4$. If 8:56 $< T_i \leq$ 9:00, the travel speed is adjusted, i.e., the code corresponding to the else statement is used. For $T_i =$ 8:58 follows $d = 2$ km, t = 8:58 and $v_1 = 0.5$ km/min. Updating the variables gives $d = 2 - 0.5(9{:}00 - 8{:}58) = 1$ km, $t = 9{:}00$, $v_2 = \frac{2}{2} = 1$ km/min, $a'_j = 9{:}00 + \frac{1}{1} = 9{:}01$, $k = 2$ and $a_j = 9{:}01$.

Whereas the approach by Eglese et al. (2006) artificially increases speed values next to boundaries of time bins, the following approach by Fleischmann et al. (2004) provides a more "natural" adaption of travel time functions. The latter handle jumps of travel time functions by linearization of transition areas. Based on Gietz (1994), they "smooth" the travel time function by piecewise linearization.

In particular, linearization parameters $\delta^k_{i,j}$ and slope $s^k_{i,j} = \frac{\tau^{k+1}_{i,j} - \tau^k_{i,j}}{2\delta^k_{i,j}}$ are calculated for each transition area $\left[z_k - \delta^k_{i,j},\ z_k + \delta^k_{i,j}\right]$. Fleischmann et al. (2004) show that the resulting arrival time function $a_{i,j}(t) = t + \tau_{i,j}(t)$ is continuous and strictly monotonic as long as $s^k_{i,j} < 1$, ensuring FIFO compliance and the exclusion of passing. In case of increasing travel times, the slope can be arbitrarily steep. A graphic example is given in Fig. 6.9. Here, the linearization of the non-FIFO transition at z_1 occurs equally in the time bin before and after the point of transition.

An example of a smoothed travel time function based on FCD is given in Fig. 6.10. Resulting travel time functions allow for a simple calculation of $a_{i,j}$ (t). The calculation of the inverse function $a^{-1}_{i,j}$(t), though, is more complicated and takes more computational effort. Fleischmann et al. (2004) recommend to rely on "forward calculation" whenever it is possible.

Since the approach by Eglese et al. (2006) refers to artificial speed reduction, a FIFO consistent travel time function as described by Fleischmann et al. (2004) is

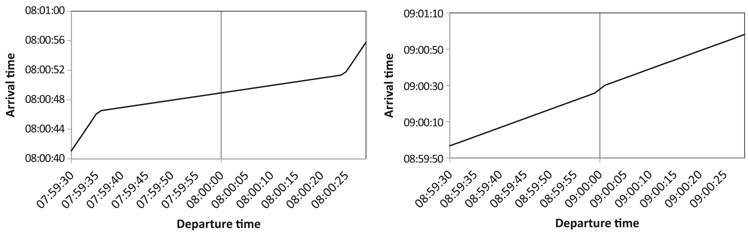


Fig. 6.10 Arrival time functions for an example link

established in the following. Here, FW information models feature a compact data set that allows for efficient provision of time-dependent travel times.

6.2.2 Implementation

The implementation of a time-dependent digital roadmap enhances the static digital roadmap introduced in Sect. 6.1.2. Time-dependent travel times result from the integration of time-dependent information models FH and FW as follows:

- With regard to FH, each link is associated with the corresponding travel time series resulting from first level aggregation. Twenty four average speeds per weekday or 168 average speeds overall are assigned, i.e., the link table is amended by 168 additional speed values per link. Travel times can be derived from the processing of speeds based on link lengths (RD data).
- With regard to FW, each link is associated with cluster representations of typical speed variations. Each cluster is characterized by its 24 speed reduction factors per weekday. Then, each link is assigned to its day-specific speed reduction factors. Travel times are derived by a lookup of the corresponding speed reduction factor, weighted by day-specific FA data.

Since time-dependent information models are derived from empirical traffic data, FIFO compliance must be ensured. Time-dependent information models are only valid for the core road network of the city of Stuttgart. Based on this FIFO compliant network, time-dependent distance matrices are calculated.

6.2.2.1 Implementation of a FIFO Network

A time-dependent digital roadmap containing FH and FW data is well suited for the instantiation of a time-aggregated graph. FH and FW data conveniently establish the travel time series for each edge. Since FH and FW time series arise from empirical traffic data, resulting time-aggregated graphs may not conform to requirements of FIFO networks, though in order to smooth the transitions between consecutive time bins, the approach of Fleischmann et al. (2004) is implemented as introduced above.

In particular, a FIFO network is constructed by calculation of parameters $s^k_{i,j}$ and $\delta^k_{i,j}$ for each transition area of consecutive time bins. Regarding FCD for Stuttgart, the slope $s^k_{i,j}$ is fixed to $s_0 = 0.9$, since 1-hour time bins allow for sufficient transition areas $\delta^k_{i,j}$, which are calculated dynamically whenever required. For the case of increasing travel times, the travel time functions are smoothed exactly one unit before and one unit after a transition $\left(\delta = 1,\ s^k_{i,j} = \left(\tau^k_{i,j} - \tau^{k-1}_{i,j}\right)/2\right)$. Thus, the FIFO property is ensured for all transitions, enabling a consistent, time-dependent representation of the city road network for subsequent shortest path computation.

Two examples of arrival time functions derived from FH data are shown in Fig. 6.10. The travel times of the particular time bins are as follows: 7:00–8:00, 71 s; 8:00–9:00, 27 s; 9:00–10:00, 29 s. On the left-hand side, the transition from the first to the second time bin is shown, which is not FIFO compliant, since travel times of the second time bin are significantly lower. To this end, travel times have to be adjusted in the area of transition. $s_o = 0.9$ leads to an interval $\delta = (71 - 26)/(2 \times 0.9)$, i.e., travel times are adjusted in the range of departure times from 7:59:35 to 8:00:25. On the right hand side, no FIFO problem occurs since travel times increase. Even so, also a linearization occurs, which can be noticed between departure times 8:59:59 and 9:00:01.

6.2.2.2 Derivation of a Core Network

Proper application and evaluation of time-dependent information models needs a sufficient amount of measurements for each network link. Thus, a *core network* is defined which corresponds to an extract of the digital roadmap of the Stuttgart area. It is constructed from a coherent network structure, beginning with a link of the inner city of Stuttgart which is well-connected and featured by a sufficient number of measurements. The core network consists of 11,148 links and 6,684 nodes. Links of the core network cover about 86% of all FCD measurements, alleviating low data availability due to taxis passing some regions infrequently at specific times and dates. They feature at least 858 and a maximum of 392,780 measurements. As a minimum, every link is endued with at least five FCD measurements per time bin. Thus, a reasonable usage within computational experiments is facilitated.

In Table 6.1, characteristics of the (time-dependent) digital roadmap for Stuttgart are summarized regarding memory efforts and preparation of information models. For a comparison of the information models from an algorithmic point of view, the resulting input data per link is pointed out. n denotes the number of links in the road network and d denotes the number of different types of days

Table 6.1 Comparison of the required volume of input data regarding different information models

Information model	Roadmap travel times (RT)	FCD averages (FA)	FCD hourly averages (FH)	FCD weighted averages (FW)
Input data	n	$d \times n$	$t \times d \times n$	$[n + (t \times k)] \times d$
Input data Stuttgart	11,148	78,036	1,872,864	79,044
Input data Stuttgart per link	1	7	168	7.1

n number of links, d number of weekdays, t number of time bins, k number of clusters

considered, whereas t indicates the number of time bins, and k the number of desired clusters. As a static benchmark, estimated travel times from the digital roadmap of Stuttgart are considered (RT data).

In case of a static information model such as RT, memory efforts correspond to the number of links of the underlying core network. FA multiplies efforts by a factor of 7, according to the number of weekdays considered. In case of time-dependent information models, a huge increase in memory efforts due to 168 travel time values per link occurs (FH). FW is able to reduce the data volume of time-dependent networks to about 7.1 input data values per link. This is due to the utilization of speed reduction factors (cf. Sect. 5.4) instead of a comprehensive time series for each link.

6.2.2.3 Calculation of Time-Dependent Distance Matrices

Time-dependent distance matrices are determined by time-dependent shortest path calculation (cf. Sect. 6.3.2). This takes a lot longer to compute and occupies a lot more space than its static counterpart, since a distance matrix has to be computed for each time bin, providing the expected travel time of a shortest path for every possible departure time.

For wide area networks with sufficient data quality, Maden et al. (2010) suggest to compute time-dependent shortest paths for departures on every full quarter-hour only in order to limit the computational burden and memory efforts of optimization. Since the maximum travel time between two nodes in the Stuttgart area is below one hour and time-dependent travel times are available in 1-hour time bins, this approach might also be sufficient for routing in city logistics. Thus, for every quarter hour, a shortest path between two points of interest is calculated, resulting in a distance matrix of $|S| \times |A| \times 672$ entries with S denoting the set of source nodes and A denoting the set of target nodes. Due to shortest paths being derived from 1-hour time bins, using narrower time bins would not provide a better approximation but increase the computational burden.

Although time-dependent shortest path calculation is based on a FIFO consistent network topology, resulting time-dependent distance matrices may still suffer from FIFO inconsistency. This is due to the inherent discretization of the time horizon. When traveling from A to B, the time of arrival at B could be earlier if the vehicle waits at A until the starting time falls in a time bin where speeds are faster. Although the construction of the time-dependent distance matrix ensures that this cannot happen as long as the best route between A and B does not change, this phenomenon may occur if the best route changes parallel to the change to a new time bin. Due to the fine-grained computation of shortest paths, no relevant impact on the computation of delivery tours is expected, though. Furthermore, FIFO inconsistency does not exacerbate optimization of delivery tours from a methodological point of view, contrasting the determination of shortest paths, which may become deficient when derived from an inconsistent topology.

6.3 Computation of Shortest Paths

The fundamental function of a planning system for routing in city logistics is the computation of shortest paths. Shortest path computation establishes input data for more complex routing problems by transformation of the detailed information of a digital roadmap (cf. Sect. 6.1.1) into a problem specific graph, representing customers, depots and their spatio-temporal relations. In the following, fundamentals of static shortest path computation are sketched. Then, the focus is on computationally efficient algorithms that take into account the dynamic nature of road networks in terms of time-dependent travel times. This is the prerequisite for time-dependent routing of a single vehicle and for time-dependent routing of a fleet of vehicles (cf. Chap. 8).

6.3.1 Shortest Path Problem

The *Shortest Path Problem* (SPP) is one of the most studied problems in graph theory with transportation as one of the most relevant application fields. SPP computation is the prerequisite for more complex routing problems, which rely on a simplified, problem-specific abstraction of the road network. For routing in city logistics, distance matrices represent a problem-specific extract of the digital roadmap in terms of typical travel times between the depot and customer locations. SPP computation establishes the entries of a distance matrix, which allows more complex routing algorithms to reduce the determination of shortest paths to a simple table lookup.

More formally, SPP computation considers a directed graph $G = (V, E)$ with n nodes and m edges representing nodes and links of a digital roadmap (cf. Sect. 6.1.1). The traversal of an edge (i, j) leads to costs $c_{i,j}$. Edge costs may be related to any mix of travel time, distance, toll, energy consumption, etc. A shortest path query between a source node s and a target node a asks for the minimal weight $c_{s,a}$ of any path from s to a. A linear programming formulation for this static version of the SPP can be found in Pallottino and Scutella (1997). The SPP as described above can be handled in linear time (Dean 2004a).

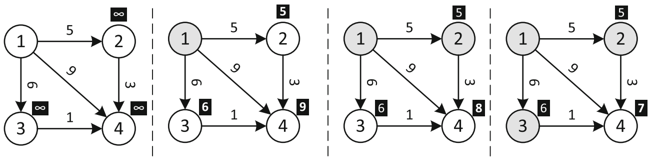


Fig. 6.11 Principle of Dijkstra's algorithm

The SPP is commonly solved by algorithms that label nodes in order to establish a tree of shortest paths in a graph G, beginning with the source node ("root"). The most famous approach was published by Dijkstra (1959), who manages an array of tentative distances $D[u] \geq d_{s,u}$, which represents the tentative distances between the source node s and all remaining nodes u. The algorithm processes the nodes in order of their distance to the source node and maintains the invariant that $D[u] = d_{s,u}$ for visited nodes. When a node u is processed, its outgoing edges (u, v) are relaxed, i.e., $D[v]$ is set to $min\{D[v], d_{s,u} + d_{u,v}\}$. Dijkstra's algorithm terminates when the target node has been processed. Its performance depends on the performance of the data structure chosen for $D[u]$, resulting in a worst case performance of $O(n^2)$ in the case of an unordered list. An overview about implementation variants and the resulting computational efforts is given by Pallottino and Scutella (1997).

In Fig. 6.11, the search process for the shortest path from node 1 to node 4 is exemplified. First, all nodes except the start node are labeled with a distance of "infinity". Then, the start node is relaxed (denoted by grey shadowing), i.e., the distance $D[2]$ is updated from "infinity" to "5", $D[3]$ is updated to "6" and $D[4]$ is updated to "9", since these nodes are directly connected to node 1. Then, node 2 is relaxed, leading to an update of $D[4] = 8$. Finally, node 3 is relaxed, leading to an update of $D[4] = 7$, which denotes the shortest path from node 1 via node 3 to node 4 with a distance of 7.

State-of-the-art extensions and concepts for the efficient computation of shortest paths in road networks are summarized by Sanders and Schultes (2007). A well-known extension of Dijkstra's approach is the *bidirectional search* variant. Here, Dijkstra's algorithm is simultaneously executed from the source node and backwards from the target node. This leads to a speedup factor around two in road networks, since labeling usually takes a roughly circular shape around the source node (Dantzig 1963). A further extension is the *A* algorithm*, which conducts a directed search and thus avoids unnecessary labeling of nodes. The main idea is that the shortest path should roughly appear in the direction of the destination (Hart et al. 1968). *A** requires a lower bound estimate of the distance to the destination. A common way is to use the Euclidean distance between the current location and the destination in combination with the average speed of the fastest road in the network.

In the following section, the enhancement of SPP algorithms to time-dependent edge costs is discussed. Here, edge costs are replaced by travel time series, representing time-dependent travel times in time-aggregated graphs.

6.3.2 Time-Dependent Shortest Path Problem

In this section, the formulation and solution of the *Time-Dependent Shortest Path Problem* (TDSPP) is discussed. The TDSPP ensures efficient calculation of time-dependent distance matrices as input for time-dependent routing problems. Resulting schedules are expected to be more reliable with respect to travel time anticipation and realization (Fleischmann et al. 2004). The influence of time dependence is commonly neglected, and shortest paths are derived based on, for example, average travel times, which are then simply adjusted to temporal variations. Delling and Wagner (2009) stated that switching from a static to a time-dependent scenario is more challenging than one might expect: The data input increases drastically as travel times on time-dependent connections change frequently during the day. Moreover, shortest paths depend heavily on the time of departure, for example, during rush hours it might pay off to avoid certain trunk roads.

Contrasting the static SPP, which has resulted in more than 2,000 scientific works since the end of the 1950s (Pallottino and Scutella 1997), the TDSPP has received significantly less attention. The TDSPP initially dates back to 1966, when it was first proposed for a discrete time setting (Cooke and Halsey 1966). Up to the year 2008, research has focused either on efficient speedup techniques for time-*independent* route planning in road networks or on modeling issues in time-dependent networks for public transportation. The focus has recently shifted to the development of efficient routing algorithms for time-dependent transportation networks.

The most important difference between SPP and TDSPP is the structure of the underlying graph, which changes fundamentally in the time-dependent case (cf. Sect. 6.2.1). In this context, it is assumed that costs resulting from the traversal of a link depend on the time a link is entered. Costs are represented by time-dependent functions $c_{i,j}(t)$ instead of scalars $c_{i,j}$. For routing in city logistics, a time-aggregated, FIFO consistent graph representation is considered. Time-dependent link costs $c_{i,j}(t)$ are estimated by time-dependent travel time functions $\tau_{i,j}(t)$. For solution methodology in the context of non-FIFO networks, i.e., where waiting at nodes is allowed, Dell'Amico et al. (2008) present a Dijkstra-like algorithm that is able to identify time-dependent shortest paths by maintaining additional labels.

Dean (2004b) gives a comprehensive overview on solution approaches for the TDSPP in FIFO graphs. He enumerates TDSPP issues that result from time-dependent graphs and ascribes them to two fundamental problems: (1) the calculation of the shortest path from a source node to all possible target nodes at a given time, and (2) the calculation of all shortest paths from one source node to all other nodes at all possible times. (1) is similar to the SPP and can be solved by a modified variant of any label-setting or label-correcting static shortest path algorithm, if the underlying graph is a FIFO graph. For discrete time networks consisting of B time bins, the computation of (2) occurs by decomposing the problem into B computations of (1).

The pseudo code of a time-dependent label-setting algorithm is introduced in Algorithm 4. In the beginning, all nodes except the source node are labeled to infinity; the source node is labeled with the desired start time. Then, remaining nodes are processed so that the node with the earliest arrival time at the current state is selected and removed from the set of remaining nodes. Labels of its successors are updated by comparing the tentative arrival time to a potential earlier arrival time when considering a path via the current node. Note that labeling of nodes occurs in terms of arrival times instead of distances, and travel time computation is based on the tentative arrival time D[i]. Corresponding to the static case, the run time of the algorithm depends on the efficient implementation of S. Dean (2004b) also introduces a label-correcting variant but states that in discrete time the label-setting algorithm is more efficient than the label-correcting algorithm.

Algorithm 4 : Time-Dependent Label-setting Algorithm	
Input:	$G = (V, E)$ with $c_{i,j}(t) = \tau_{i,j}(t)$,
	$D[i]$ array of tentative arrival times,
	$T_s =$ start time, source node $v_s \in V$, set of nodes $S \subseteq V$
Initialize:	$\forall i \in V \backslash \{v_s\}: D[i] \leftarrow \infty;$
	$D[s] \leftarrow T_0;$
	$S \leftarrow V;$
Processing:	while $S \neq \phi$ do
	$i \leftarrow i \mid min\{D[i]\};$
	$S \leftarrow S \backslash \{i\};$
	$\forall j$ such that $(i,j) \in E$ do
	$D[j] \leftarrow min\{D[j], D[i] + c_{i,j}(D[i])\};$

For efficient computation of TDSPP in large graphs, Delling and Wagner (2009) introduce a framework based on the principle of Dijkstra's algorithm. The main idea is to conduct a more intelligent guidance in time-dependent graphs. Thus, only as much nodes as really necessary are visited during the search process. Computational experiments show that recent approaches are capable of handling networks up to 18 million nodes and 42.6 million edges, corresponding to a digital roadmap of Western Europe.

The computation of "travel time profiles" has recently been discussed as an alternative for the provision of input data for time-dependent routing problems. Geisberger and Sanders (2010) summarize static approaches and propose a complex shortest path algorithm for very large time-dependent networks. Travel time profiles denote the minimum travel times from each source node to each target node. A "min–max search" gives a rough approximation of the travel time profile based on global minima and maxima of the links' travel times, resulting in a lower and an upper bound of the travel time profile. Their approach is evaluated by computational experiments for a time-dependent wide-area road network of 4.7 million nodes and 10.8 million edges, featuring about 8% time-dependent edges. Based on such a network, time-dependent travel times can be computed for up to

500 source and destination nodes. Lots of computational power is required due to the complexity of this approach and the underlying network size; computational experiments are conducted on a machine with two Intel Xeon X5550 processors (Quad-Core) at 2.67 GHz and 48 GB of RAM.

TDSPP computation represents the fundamental function for time-dependent routing by transforming the detailed level of a time-dependent digital roadmap into a problem specific abstraction required for more complex routing problems. FIFO consistent processing of input data ensures the efficient computation of time-dependent distance matrices, enabling the determination of time-dependent delivery tours.

Part IV
Optimization Models

Chapter 7
Evaluation of Information Models

In this chapter, presented information models are evaluated with respect to their usage in time-dependent shortest path computation. The aim is to find the most suitable information model for complex routing problems in city logistics, which are the subject of Part IV. Simulation of shortest paths allows for a detailed analysis of quality and reliability of static and time-dependent information models. In particular, it is explored which information model provides the most reliable travel-time prediction, and which information model leads to the realization of the time-shortest itinerary.

To this end, a large number of randomly chosen origin–destination (OD) pairs are investigated. Itineraries are examined according to five representative traffic scenarios, reflecting the impact of time-dependent travel times. Information models introduced in Part II are used for the scheduling of itineraries for each combination of OD pair and information model. Planning quality and reliability are then evaluated by simulation of scheduled itineraries. The experiments are based on Taxi-FCD collected in the area of Stuttgart. First, the experimental setup is explained (cf. Sect. 7.1), followed by the simulation approach (cf. Sect. 7.2). Then, simulation results are presented for a choice of itineraries as well as for all OD pairs (cf. Sect. 7.3).

7.1 Experimental Setup

Evaluation of information models for routing in city logistics expects a sufficient number of itineraries and their realization by simulation. Origin and destination of itineraries are defined by OD pairs. To consider different traffic flow patterns and different local variability of urban traffic flows, OD pairs are randomly chosen in terms of *traveler scenarios*. Traveler scenarios represent different types of itineraries commonly occurring in conurbations. Therefore, the road network under

J. F. Ehmke, *Integration of Information and Optimization Models for Routing in City Logistics*, International Series in Operations Research & Management Science 177,
DOI: 10.1007/978-1-4614-3628-7_7,

investigation is divided into an inner city and an outer city area. Thus, slow inner city traffic suffering from, for example, traffic light perturbations can be separated from faster interurban traffic on trunk roads in the outskirts.

7.1.1 Traveler Scenarios

Based on the EDA of Stuttgart FCD (cf. Sect. 5.5), traveler scenarios are defined as follows:

- Traveler Scenario 1 comprises trips from the outskirts of the city to the inner city area (downtown), for example, from the airport to the main station. The origin of itineraries is located outside the city, whereas the destination is located within the inner city area. These routes are heavily congested at certain times of the day, leading to high fluctuation of travel times.
- Traveler Scenario 2 refers to trips starting and terminating in the inner city district. Such trips are sensitive to manifold perturbations in inner city traffic.
- Traveler Scenario 3 comprises itineraries that start and terminate in the outskirts. This traveler scenario offers a range of alternatives for passing or avoiding the city center. In addition to the highly frequented roads of the city center, vast parts of the itineraries follow roads in the outskirts. Thus, variation of inner city traffic can be analyzed in combination with temporarily congested trunk roads.

Simulation occurs based on the core network of the city of Stuttgart introduced in Sect. 6.2.2. For each of the traveler scenarios, 100 OD pairs are selected randomly from the core network's node table, making a total of 300 basic itineraries. Hence, the spatial structure of computational experiments is defined.

7.1.2 Traffic Scenarios

Time-dependent routing strongly correlates with time-dependent variation of travel times throughout the day. Thus, itineraries are scheduled according to the expected realization time, i.e., for a specific time bin, which is referred to as *traffic scenario*. Representative traffic scenarios have been identified by EDA, resulting in a spatio-temporal overview of traffic quality and traffic quality variation for the traffic system under investigation. Based on the results of the EDA for the Stuttgart area (cf. Fig. 5.14), the following time bins have been selected:

- "Free flow" traffic (2:00–3:00 and 21:00–22:00) with a high ratio of traffic quality levels A and B,
- "Rush hour" traffic (8:00–9:00 and 17:00–18:00) with a high ratio of traffic quality levels E and F,

- "Average" traffic (13:00–14:00) with similar ratio of traffic quality levels A/B and C/D.

Thursday is chosen as an exemplary day of the week featuring commuter traffic as well as sufficient availability of Taxi-FCD.

7.1.3 Information Models

Four information models are used to schedule shortest itineraries:

- *Roadmap travel time (RT).* The common way of finding time-shortest itineraries in city logistics is using average speeds. Estimated travel times derived from legal speeds serve as a benchmark for the following FCD-based travel times.
- *FCD averages (FA).* Travel times resulting from aggregating FCD to one average measure per link and day of the week.
- *FCD hourly averages (FH).* Travel times resulting from aggregating FCD to 24 averages per link, dependent on day of the week and time of the day.
- *FCD weighted averages (FW).* Travel times derived from speed reduction factors, weighted by an FA data set.

The three traveler scenarios (consisting of 100 OD pairs each), the five traffic scenarios, and the four information models lead to a sum of 3,600 genuine itineraries being subject of realization by simulation. Note that the time-dependent information models (FH, FW) establish one genuine itinerary for each traffic scenario, corresponding to a total of $2 \times 300 \times 5 = 3,000$ itineraries, whereas RT and FA information models lead to only one itinerary (in total: 2×300), independently of particular traffic scenarios. For all genuine itineraries, shortest paths are computed using Dijkstra's algorithm (cf. Sect. 6.3.1) due to every itinerary beginning on the hour and definitely not taking longer than 60 min.

7.2 Simulation and Evaluation of Shortest Paths

To determine the information model most suited for routing in city logistics, itineraries are scheduled for different traffic scenarios and then "realized" by simulation. This imitates a deterministic planning approach, where a dispatcher schedules itineraries for future realization. Simulation acts as a "feedback loop" for the evaluation of the reliability of travel time estimation. Different types of information models lead to different travel time estimations, which are compared to "true" travel times resulting from simulation.

Planning and simulation of itineraries are based on two distinct sets of FCD measurements. Planning data is derived from FCD recorded in even weeks of the year, whereas simulation data is derived from FCD measured in odd numbered weeks. Planning data sets are utilized in shortest path calculation, resulting in an information model-dependent shortest itinerary, its particular links, and the corresponding estimated travel time ("duration"). For subsequent computational experiments, simulation occurs by replacing the original travel times of a scheduled itinerary by time and date-specific travel times derived from simulation data. This simulates the situation where an itinerary would be realized at a certain time and date.

For simulation, a scheduled itinerary is segmented into its particular links. For each link, FCD measurements are derived from the FCD database according to the *simulation context*. A simulation context is defined by the time and date of simulation and the corresponding time bin (e.g., 15/05/2003, 8:00–9:00). If there is more than one FCD measurement for a link within the simulation context, simulated travel time is derived from the mean of FCD measurements. In the end, the simulated duration of a scheduled itinerary results from the sum of (mean) travel times of corresponding links. Then, the *simulated* duration can be compared to the *scheduled* duration to draw conclusions on planning quality, i.e., the reliability of travel time estimations. In case of FCD not being available for all links of the scheduled itinerary, no simulation occurs.

More formally, a scheduled itinerary is defined by its links resulting from shortest path computation and by the corresponding itinerary duration d^p. Simulation of a scheduled itinerary leads to n simulated itinerary durations d_i^s, resulting from the simulation on specific dates and time bins. The evaluation works as follows:

- For each scheduled itinerary, the corresponding mean simulated duration $\overline{d^s}$ is calculated by $\overline{d^s} = \frac{1}{n}\sum_{i=1}^{n} d_i^s$, n representing the number of simulations carried out.
- The comparison of scheduled and simulated durations is done in terms of the *relative absolute difference* $\text{RAD} = \sum_{i=1}^{n} \frac{|d_i^s - d^p|}{d^p}$. RAD allows for conclusions on the planning quality with respect to travel time estimation. The smaller the RAD, the more reliable is the scheduled duration.
- The *variation coefficient* $c = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(d_i^s - \overline{d^s})^2}}{\overline{d^s}}$ is considered for each scheduled itinerary. It is consulted to draw conclusions on the relative robustness of scheduled itineraries in terms of travel time variation. The smaller the variation coefficient, the more reliable is the selected route with respect to robustness of travel times.

RAD and variation coefficient are calculated for each scheduled itinerary, i.e., for each set of n simulated itineraries that corresponds to the scheduled itinerary. For overall analysis, RADs and variation coefficients are averaged according to the specified dimension, for example, the average RAD of all itineraries scheduled by RT in the 2:00–3:00 traveler scenario.

In the following, an example for the simulation and evaluation of an itinerary is provided. For instance, FA-based planning results in a scheduled itinerary d with several links and a scheduled duration of $d^p = 13.5$ min. Simulation occurs between 8:00 and 9:00 at 15/05/2003, i.e., FCD measurements from this time period are averaged for each link of the scheduled itinerary. For the first link, five measurements are available within this time period, resulting in an average speed of 19.8 km/h or an average travel time of 23.1 s for traversing the entire link, respectively. The sum of the average link travel times of this time period is then raised to denote the simulated duration $d_1^s = 17.6$ min for 8:00–9:00, 15/05/2003. Evaluation of the scheduled itinerary considers the average of all n simulated durations, i.e., simulations resulting from all Thursdays in uneven numbered weeks with a sufficient amount of FCD measurements for the 8:00–9:00 time bin. Here, $n = 16$ simulations can be summed up, resulting in a mean simulated duration of $\overline{d^s} = 16.2$ min with RAD = 23.3% and $c = 14.7\%$.

7.3 Computational Results

In this section, computational results for the evaluation of time-dependent information models for the area of Stuttgart are presented. First, routing quality is discussed for example itineraries. Then, an overall evaluation of RT, FA, FH, and FW information models allows for general statements on reliability and robustness of information models. Simulation results are evaluated in terms of traveler scenarios, traffic scenarios, and the combination of both.

7.3.1 Evaluation of Example Itineraries

First, the planner's view is taken by investigating one example itinerary out of each traveler scenario to demonstrate planning and simulation based on RT, FA, FH, and FW information models. Example itineraries allow for detailed insights into specific planning situations. The corresponding itineraries are depicted in Fig. 7.1. The black box indicates the border between inner city and outskirts, defining the structure of the simulation experiment in terms of traveler scenarios. Note that there is (almost) no variation in chosen routes due to the simplified structure of the core network.

In Fig. 7.2, RADs of scheduled and simulated durations are depicted for each example itinerary in terms of information models and traffic scenarios. Detailed numerical results are available in Table 7.1. Note that in the RT and FA case there is only one scheduled duration in contrast to time-dependent routing based on FH and FW data, which establish an individual travel time estimation for each traffic scenario.

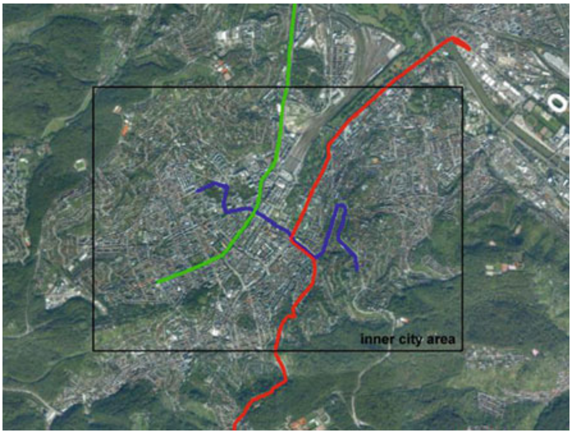


Fig. 7.1 Example itineraries (aerial view with kind permission of GeoContent GmbH, Magdeburg, Germany)

The first itinerary (Traveler Scenario 1) starts in the outskirts in the north of the city and terminates downtown, comprising a length of 7.1 km. Here, typical commuter traffic flows occur, leading to relatively high RADs concerning the afternoon rush hour (e.g., up to 76% in case of RT). FA-based planning is able to reduce RADs to a maximum of 40%. FH and FW provide the best travel time estimations, resulting in relatively small RADs of 4–21%. Mean simulated durations are nearly the same regarding different information models. They are slightly overestimated by FA in the evening and at night, whereas they are underestimated for all other traveler scenarios.

The example itinerary defined by Traveler Scenario 2 leads directly through the heart of the city (length: 4.1 km). FH and FW feature itineraries that are more reliable in terms of travel time anticipation, resulting in RADs of 4–12%, whereas static planning leads to significantly higher RADs (FA up to 25%, RT up to 61%). Mean simulated durations are similar within all information models, but differ a lot regarding different traffic scenarios (free flow: 6.4 min; rush hour: 9.2 min).

The third itinerary connects two suburbs (north and south) by traveling through the inner city area (12.0 km). Results indicate relatively good travel time estimations for all information models during free flow and average traffic scenarios (RAD: up to 12%) in contrast to high RADs during rush hours, reflecting typical commuter traffic (up to 50%). Once more, FH and FW provide relatively lower RADs in rush hours, but FW is performing worse than FH. As in Traveler Scenario 1, mean simulated route durations are the same throughout all traffic data sets due to the same scheduled route structure (e.g., free flow: 13.2 min; evening rush hour: 21.7 min).

In sum, RT leads to very poor travel time estimations throughout all time bins of Traveler Scenarios 1 and 2, especially in rush hours. To a large extent, FA-based itineraries provoke considerably smaller RADs than the RT-based itineraries. This is mainly due to FCD-based information models incorporating perturbations of city traffic, leading to more realistic travel time anticipations. FH improves travel time estimation in all traveler scenarios, leading to smaller RADs of scheduled and

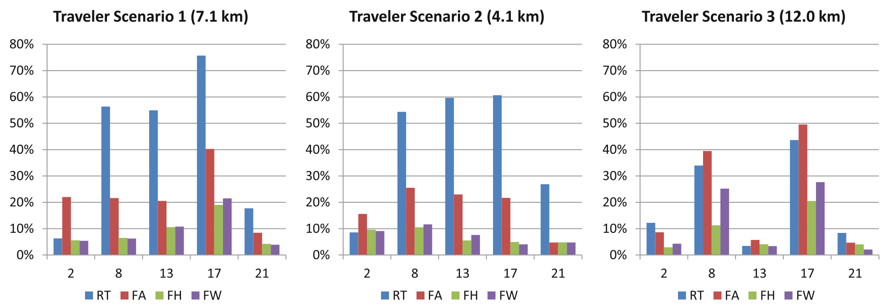


Fig. 7.2 RADs for example itineraries

Table 7.1 Detailed results for example itineraries

		Roadmap Travel Time (RT)					FCD Averages (FA)					FCD Hourly Averages (FH)					FCD Weighted Averages (FW)				
		sched (min)	sim (min)	diff (%)	varc (%)	count (no)	sched (min)	sim (min)	diff (%)	varc (%)	count (no)	sched (min)	sim (min)	diff (%)	varc (%)	count (no)	sched (min)	sim (min)	diff (%)	varc (%)	count (no)
Traveler Sc. 1	2		9.7	6.3	7.5	10		9.7	22.0	7.5	10	9.9	9.8	5.6	6.4	6	9.8	9.8	5.4	6.4	6
	8		15.1	56.4	7.8	11		15.1	21.6	7.8	11	14.7	15.1	6.5	7.8	11	14.9	15.1	6.2	7.8	11
	13	9.7	15.0	54.9	19.8	9	12.4	15.0	20.5	19.8	9	14.0	15.0	10.6	19.8	9	14.2	15.0	10.7	19.8	9
	17		17.0	75.7	23.3	9		17.0	40.2	23.3	9	17.9	17.3	19.0	25.6	6	16.0	17.0	21.5	23.3	9
	21		11.4	17.7	4.4	13		11.4	8.4	4.4	13	11.1	11.4	4.2	4.4	13	11.7	11.4	3.9	3.8	11
Traveler Sc. 2	2		6.3	8.5	8.2	10		6.4	15.6	8.5	6	5.9	6.4	9.6	8.5	6	6.0	6.4	9.1	8.5	6
	8		9.2	54.3	13.2	10		9.5	25.5	10.4	6	9.2	9.5	10.5	10.4	6	8.9	9.5	11.6	10.4	6
	13	5.9	9.5	59.7	11.8	12	7.5	9.3	23.0	5.3	9	8.9	9.3	5.5	5.3	9	8.7	9.3	7.6	5.3	9
	17		9.6	60.7	7.9	7		9.2	21.7	4.4	6	8.8	9.2	4.9	4.4	6	9.3	9.2	4.0	4.4	6
	21		7.5	26.9	4.4	8		7.2	4.7	4.8	5	7.1	7.2	4.8	4.8	5	7.1	7.2	4.7	4.8	5
Traveler Sc. 3	2		13.2	12.2	0.0	2		13.2	8.6	0.0	2	12.9	13.2	2.9	0.0	2	12.7	13.2	4.3	0.0	2
	8		20.2	34.0	13.7	12		20.2	39.5	13.7	12	19.6	20.2	11.3	13.7	12	16.2	20.2	25.2	13.7	12
	13	15.1	15.3	3.4	3.5	10	14.5	15.3	5.7	3.5	10	14.7	15.3	4.1	3.5	10	15.4	15.3	3.4	3.5	10
	17		21.7	43.6	21.9	8		21.7	49.5	21.9	8	18.6	21.7	20.5	21.9	8	17.1	21.7	27.7	21.9	8
	21		13.8	8.4	2.3	4		13.8	4.6	2.3	4	13.3	13.8	4.0	2.3	4	13.9	13.8	2.1	2.3	4

Key
sched: scheduled duration
varc: variation coefficient of sim
sim: mean simulated duration
diff: mean RAD between sched and sim
count: number of simulations

simulated route durations. Especially in congested time bins like morning and evening rush hours, RADs can be reduced significantly. At the most, RADs resulting from FW data are comparable to RADs resulting from FH data. The need for time-dependent data, especially in rush hours, is confirmed throughout all traveler scenarios.

Regarding robustness of scheduled itineraries, variation coefficients are almost coincident for particular traffic scenarios due to the simplified structure of the underlying core road network (cf. Fig. 7.3). For each time bin, scheduled itineraries do not differ a lot from each other with respect to chosen links. Variation coefficient increases throughout the day in Traveler Scenario 1 to a maximum of about 26% in the evening rush hour due to commuter traffic meeting recreational traffic. In Traveler Scenario 2, variation coefficients are relatively stable. In Traveler Scenario 3, higher variation coefficients in rush hours are clearly visible, especially in the afternoon, implicating that itineraries scheduled are relatively unreliable during rush hours.

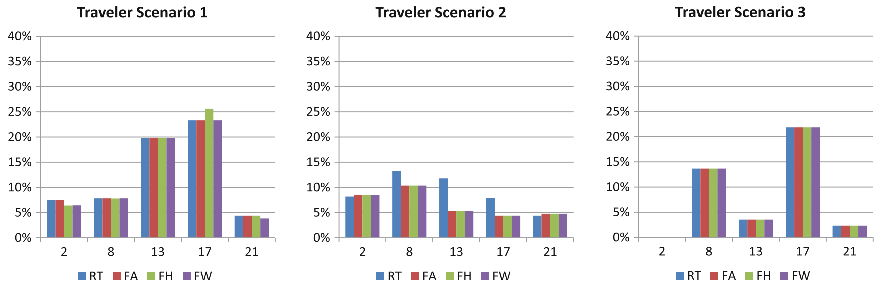


Fig. 7.3 Variation coefficients of example itineraries

Table 7.2 Overall results

Information model	Sched (min)	Sim (min)	Diff (%)	Varc (%)	Realized (%)	Count (no)
RT	8.2	10.9	38	10	64	5788
FA	9.1	10.2	20	11	62	5751
FH	9.5	9.9	11	10	61	5675
FW	9.7	10.1	11	11	62	5809

In the following, the evaluation of information models is conducted based on a large number of OD pairs.

7.3.2 Overall Evaluation

The overall evaluation of all 3,600 scheduled itineraries allows for general statements on the reliability of investigated information models in a metropolitan area. First, average results of all simulations are reported, allowing for general conclusions on the quality of routing. Then, results on the level of traveler scenarios as well as on the level of traffic scenarios are discussed.

7.3.2.1 Overall Results

In Table 7.2, overall results of all 3,600 scheduled itineraries and 23,023 simulations are reported in terms of mean scheduled duration (*sched*), mean simulated duration (*sim*), mean RADs of scheduled, and simulated duration (*diff*), mean variation coefficient of simulated durations (*varc*), the share of scheduled itineraries simulated (*realized*) and the total number of underlying simulations (*count*).

Generally, RT establishes most optimistic mean scheduled durations (8.2 min) in contrast to the other information models, resulting in the worst mean simulated route durations, though (10.9 min). Accordingly, mean RADs are high (38%) and

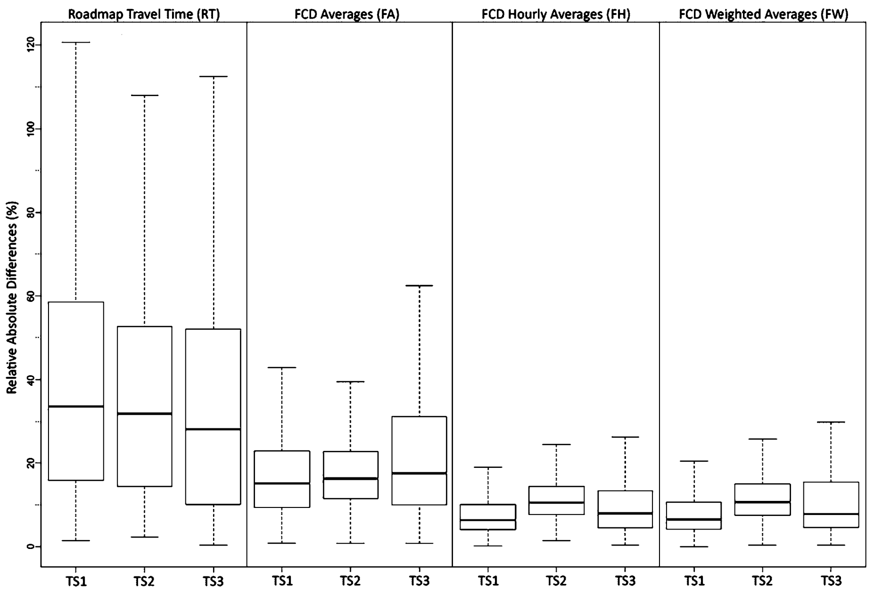


Fig. 7.4 Box-and-whisker plots of RADs for each traveler scenario

decrease a lot in the FA (20%) and FH/FW case (11%). Mean variation coefficients of simulated routes durations are alike, since itineraries scheduled by different information models are often the same due to the simplified structure of the core network. The share of scheduled itineraries is about the same (61–64%) throughout all information models, reflecting that not every chosen OD pair could be simulated with sufficient data availability for all traffic scenarios at all dates and time bins.

In sum, travel time estimation based on RT is not well suited for the estimation of urban travel times. FA-, FH-, and FW- based itineraries result in tours that are about 1 min shorter on average. Regarding quality of estimation, FA is a better guess in general, but suffers from over- and underestimations induced by static information models. FH and FW lead to a further reduction of 9% in terms of RADs due to incorporation of time dependence, thus providing more realistic itineraries. FH and FW perform on the same overall level regarding quality of travel time estimation.

7.3.2.2 Analysis of Traveler Scenarios

More insights into the performance of information models can be drawn from the analysis of simulated itineraries with respect to particular traveler scenarios. In Fig. 7.4, medians, quartiles and whiskers of RADs are depicted for each information model in the context of the three traveler scenarios. Outliers have been omitted. Note that the average length of itineraries is larger in case of Traveler

Table 7.3 Detailed results for all itineraries

		Roadmap Travel Time (RT)					FCD Averages (FA)					FCD Hourly Averages (FH)					FCD Weighted Averages (FW)				
		sched (min)	sim (min)	diff (%)	varc (%)	real (%)	sched (min)	sim (min)	diff (%)	varc (%)	real (%)	sched (min)	sim (min)	diff (%)	varc (%)	real (%)	sched (min)	sim (min)	diff (%)	varc (%)	real (%)
	overall	8.2	10.9	37.6	10.3	64	9.1	10.2	20.3	10.9	62	9.5	9.9	10.5	10.4	61	9.7	10.1	11.4	10.7	62
	Traveler Sc. 1	10.1	13.9	39.6	7.8	60	12.1	13.3	17.8	7.9	52	12.6	13.1	7.7	6.9	52	12.7	13.1	8.6	7.4	53
	Traveler Sc. 2	3.8	5.0	36.9	12.6	78	4.4	4.8	19.9	12.9	80	4.6	4.8	12.3	12.8	80	4.7	4.8	12.8	12.9	81
	Traveler Sc. 3	12.5	16.2	36.5	9.1	54	13.4	15.3	23.2	10.2	53	14.1	14.9	10.7	9.3	51	14.3	15.3	11.9	9.8	53
Traffic scenarios	2	8.3	8.5	12.1	7.2	65	8.9	7.6	15.7	7.9	61	7.5	7.8	8.4	7.5	64	7.5	7.8	8.4	7.5	64
	8	8.3	12.1	48.0	11.5	63	9.4	11.6	24.1	12.2	62	11.0	11.3	10.9	11.3	58	10.8	11.5	14.1	12.2	61
	13	8.8	12.0	40.5	9.9	71	9.4	10.6	15.3	10.2	65	10.3	10.7	9.5	10.1	67	10.5	10.7	9.2	10.1	67
	17	8.0	13.0	65.5	15.5	63	9.4	12.6	35.5	15.8	65	10.8	11.5	14.8	15.2	61	11.3	12.2	16.4	15.7	65
	21	7.6	8.8	20.8	7.5	57	8.6	8.3	8.8	8.0	55	7.8	8.2	9.0	8.1	54	8.1	8.3	8.4	7.9	55
Traveler Scenario 1	2	9.9	10.4	10.5	5.0	56	11.7	9.9	15.0	5.1	51	9.8	10.2	5.7	4.6	57	9.8	10.1	5.7	4.5	56
	8	10.0	15.1	53.0	8.1	57	12.2	15.0	23.7	9.1	50	14.6	15.2	8.4	7.9	48	14.0	15.0	12.0	8.6	50
	13	10.4	14.6	42.8	8.7	69	12.2	13.7	12.8	8.0	55	13.0	13.6	7.9	7.9	58	13.3	13.5	7.3	7.9	58
	17	10.3	16.9	65.3	12.2	65	12.3	15.7	28.4	11.7	58	14.4	15.1	10.2	9.5	52	14.7	15.2	11.2	11.2	57
	21	9.9	11.6	19.5	3.6	51	12.0	11.7	7.1	4.4	45	11.3	11.9	6.1	4.2	46	11.2	11.5	6.9	4.3	43
Traveler Scenario 2	2	3.8	3.9	14.8	10.0	78	4.4	3.7	16.5	10.6	79	3.5	3.7	10.9	10.5	79	3.6	3.7	10.9	10.5	81
	8	3.8	5.3	42.3	13.7	79	4.3	5.0	20.0	13.8	79	4.9	4.9	12.0	13.1	79	5.0	5.0	13.6	13.6	79
	13	3.9	5.3	41.3	12.3	80	4.4	5.0	16.5	12.3	81	4.9	5.0	11.0	12.5	83	5.0	5.0	10.9	12.4	83
	17	3.8	6.2	62.4	17.5	79	4.5	6.0	35.6	18.3	82	5.5	6.0	17.2	18.2	84	5.5	6.0	19.0	18.3	84
	21	3.8	4.4	23.0	9.5	76	4.3	4.2	9.9	9.6	77	4.0	4.1	9.7	9.6	76	4.1	4.2	9.4	9.6	77
Traveler Scenario 3	2	12.8	12.5	10.0	5.3	61	12.9	11.1	15.1	6.1	52	10.7	11.2	7.7	6.0	57	10.7	11.3	7.6	5.9	56
	8	13.1	19.1	51.3	11.4	54	14.1	17.9	30.1	12.3	57	17.3	17.8	11.6	11.0	48	16.5	18.1	16.6	12.9	53
	13	13.2	17.4	37.2	7.8	65	13.5	15.3	15.8	9.0	59	15.1	16.0	9.0	8.6	60	15.4	15.9	8.9	8.9	59
	17	11.7	19.1	71.0	15.7	46	13.6	19.1	42.8	15.7	55	16.1	17.5	15.3	14.9	47	16.6	18.6	18.0	15.3	53
	21	11.4	12.9	18.5	7.3	45	12.6	12.3	8.7	7.8	43	11.1	11.8	10.8	8.1	41	12.0	12.3	8.2	7.6	44

Key

sched: scheduled duration *sim:* mean simulated duration *real:* share of genuine itineraries simulated

varc: variation coefficient of sim *diff:* mean RADs between sched and sim

Scenarios 1 and 3 (8.6 versus 11.4 km), whereas inner city routes in Traveler Scenario 2 comprise an average length of 2.5 km only. In the following, the focus is on the comparison of mean RADs. Detailed figures on mean simulated durations, mean RADs, and mean variation coefficients can be found in Table 7.3.

In general, mean RADs and variation of RADs are large in the RT case throughout all traveler scenarios (large spans of boxes). Here, scheduled travel times are too optimistic (e.g., mean scheduled duration of 10.1 min compared to 13.9 min mean simulated duration in Traveler Scenario 1). FA leads to smaller RADs and a reduction in variation. When changing from static (RT/FA) to time-dependent planning (FH/FW), mean RADs and their variation are reduced once more throughout all traveler scenarios. The boxes become shorter due to decreasing variation of RADs, and medians make toward zero, for example, for Traveler Scenario 1 (TS1), mean RADs decrease from 39.6% (RT) via 17.9% (FA) to 8–9% (FH/FW). Furthermore, the range between upper and lower whisker is decreasing, indicating superior travel time anticipation. This is also the case for the remaining traveler scenarios. It is clearly visible that FH and FW lead to similar results in terms of mean RADs and simulated travel times (here: except Traveler Scenario 3, which shows a small increase in mean simulated durations).

In sum, downtown and tangential routes (Traveler Scenarios 1 and 3) seem to benefit a little more from time-dependent planning than inner city routes (Traveler Scenario 2). This may result from a general higher noise in travel times of inner city areas.

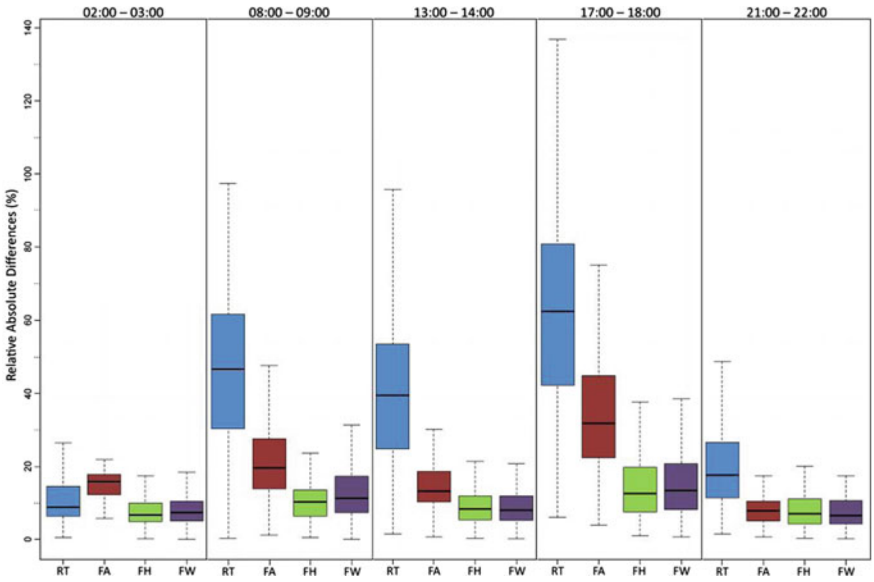


Fig. 7.5 Box-and-whisker plots of RADs for each traffic scenario

7.3.2.3 Analysis of Traffic Scenarios

Regarding different traffic scenarios, Fig. 7.5 gives detailed insights into the quality of travel time estimations. As expected, static information models lead to very high mean RADs and high variation of RADs in rush hours (e.g., FA in the afternoon: average RAD of 35%, mean scheduled duration 9.4 min, mean simulated duration 12.6 min). FH and FW are able to reduce mean RADs to a level of about 12–16%, settling on an inherent level of "system noise."

In the remaining traffic scenarios, FH and FW are able to keep RADs on a level of approximately 8–10%. Again, FH and FW perform on the same level of quality concerning travel time estimation while limiting variation simulated itinerary durations. In the evening, day specific FA is a very good guess, prevailing FH and FW. At night, RT works relatively well due to free flow in city traffic.

7.3.2.4 Analysis of Traveler and Traffic Scenarios

In Fig. 7.6, mean RADs per traveler scenario are depicted in the context of traffic scenarios. All traveler scenarios benefit from time-dependent information models as RADs decrease. At night and in the evening, mean RADs are relatively small throughout all information models. Focusing time bins during the day, rush-hour effects are clearly visible, resulting in poor performance of RT and FA information models throughout all traveler scenarios. For Traveler Scenario 2, there is no

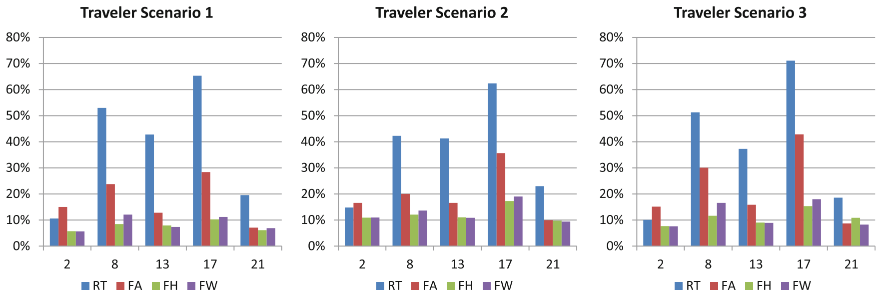


Fig. 7.6 Mean RADs for each traveler and traffic scenario

general difference between Traffic Scenarios "8" and "13" probably due to the high level of noise of inner city travel times. In any case, FH and FW are able to reduce RADs down to a maximum of 19%. Reliability guarantee of FA and RT is only 43 and 71%, respectively, depending on the particular traveler and traffic scenario.

All in all, FH and FW information models lead to an increase of planning reliability, which is underlined by a significant reduction of RADs compared to static traffic data sets. RT results in far too optimistic durations in all traffic scenarios except night hours. FA allows for a good guess at free flow times, whereas FH and FW ensure consistent levels of planning quality by RADs of about 19% in rush hours and 11% in free flow times. In the FW case, the required amount of data is tremendously reduced, keeping the level of FA efforts, but still allowing for the consideration of time dependence.

In Chap. 9, FA and FW information models serve as a fundament for the comparison of static and time-dependent optimization models. Based on FA and FW data, distance matrices are computed, establishing input data for static and time-dependent optimization procedures.

Chapter 8
Routing in City Logistics

In this chapter, optimization models for routing in city logistics are introduced and evaluated. A planning framework is established that features the provision of cost-efficient as well as customer-oriented delivery tours as required by demanding routing applications introduced in Part I. To this end, an overview on problem formulations, optimization models, and solution procedures for "offline" routing is given, i.e., the most relevant information for the optimization of transportation resources is assumed to be available the day before realization. The focus is especially on the incorporation of information concerning the expected state of the city road network in terms of information models presented in Part II. In Chap. 9, the planning framework is evaluated by means of computational experiments.

Contrasting the majority of similar work, planning problems from the area of offline routing are handled by *dynamic-deterministic* optimization models (Ghiani et al. 2003). Here, deterministic refers to that transportation requests and the expected course of travel times are known beforehand. However, travel times are time-dependent, which requires optimization models to be capable of handling dynamic input data. Time-dependent input data distinguishes this type of optimization problems from well-known, static-deterministic optimization models, since dynamic-deterministic optimization precludes the efficient extension of most of the heuristics for the solution of static-deterministic optimization models. Thus, static-deterministic optimization procedures are introduced and then extended to the dynamic-deterministic case, considering time-dependent travel times. Pseudo code clarifies the functionality of the presented algorithms. Time-dependent distance matrices serve as data source for the routing of a single vehicle as well as for the routing of a fleet of vehicles.

The routing of a single vehicle is investigated in Sect. 8.1, which refers to the well-known Traveling Salesman Problem (TSP). First, static-deterministic TSP formulations and solution approaches are introduced and then extended to the dynamic-deterministic, time-dependent case (TDTSP). From this point, the extension to routing of a fleet of vehicles is obvious (VRP and TDVRP in Sect. 8.2).

J. F. Ehmke, *Integration of Information and Optimization Models for Routing in City Logistics*, International Series in Operations Research & Management Science 177,
DOI: 10.1007/978-1-4614-3628-7_8,

In Sect. 8.3, the discussion on optimization models is concluded by incorporation of customer time windows for the static as well as for the time-dependent case of the VRP (VRPTW and TDVRPTW). These problem formulations correspond to increasing requirements resulting from demanding routing applications in city logistics.

8.1 Routing of a Single Vehicle

In the following sections, optimization models and optimization procedures for the computation of cost-efficient delivery tours are discussed. In this context, two problems occur: (1) the assignment of customers to individual tours, and (2) the determination of the efficient order of customers' visits. In this section, the focus is on the *order of the customers*. This corresponds to the routing of a single vehicle, which is a prerequisite for the routing of a fleet of vehicles (cf. Sect. 8.2). Routing of a single vehicle is well-known as the TSP. Whereas the static-deterministic TSP belongs to the most propagated optimization problems, the time-dependent and thus more realistic extension receives less attention.

8.1.1 Traveling Salesman Problem

The TSP deals with the computation of the optimal tour for a vehicle considering a given set of customers, i.e., the desire is to find the tour that results in the lowest costs visiting all customers exactly once. The TSP is one of the most famous and most investigated combinatorial optimization problems. Lawler et al. (1985) give a comprehensive overview on the history of the TSP. It can be seen as a special case of the VRP, which enhances the problem to a fleet of vehicles.

In order to describe TSP and VRP optimization models, some fundamental definitions are required. A standard problem formulation may be as follows. Vehicles are positioned at a single *depot*. A *tour* comprises the set of customers that are served by a vehicle on a specific trip; the order of customer visits is denoted by a *route*. More formally, let $G = (V, E, D)$ be a complete, directed, evaluated graph consisting of customer and depot nodes $V = \{v_0, v_1, \ldots, v_n\}$ and edges $E = \{(i,j) | i,j \in V, i \neq j\}$. v_0 represents the depot node, whereas remaining nodes represent customer nodes. A distance matrix D contains the costs $d_{i,j}$ that arise when traveling from v_i to v_j. The entries of D represent static costs such as shortest distances or average travel times. Then, the TSP aims at the determination of the optimal tour by minimization of costs resulting from visiting each customer exactly once and starting and terminating the tour at the depot. If $d_{i,j} = d_{j,i} \; \forall \, i,j$, the problem is said to be *symmetric*; otherwise, it is *asymmetric*. Applegate et al. (2007) give a comprehensive overview on the history of the symmetric TSP, its applications, solution approaches, and recent successes in solution quality and tractable problem size. In the following, the asymmetric variant of the TSP is focused, which is more reasonable for city road networks.

In the Euclidean plane, the entries of the distance matrix D can be derived from Euclidean distances $d_{i,j} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}$, (x_i, y_i) and $\left(x_j, y_j\right)$ being coordinates of depot and customer locations. For routing in city logistics, SPP computation is required in order to give a realistic estimation of distances or travel times, respectively (cf. Sect. 6.3). SPP computation may be based on RT, RD, or FA information models, providing input for the entries of the distance matrix D. D features a compact representation of costs arising from traveling in the underlying road network and allows for the efficient computation of the cost-optimal order of delivery.

The TSP may be described in terms of a mathematical model. The mathematical formulation for the asymmetric TSP is as follows, distinguishing variables, the objective function and constraints (Domschke 1997):

Variables: $$x_{i,j} = \left\{ \begin{matrix} 1 \text{ if tour includes edge from } i \text{ to } j \\ 0 \text{ else} \end{matrix} \right\}$$

Objective function: $$\sum_{i=1}^{n} \sum_{j=1}^{n} d_{i,j} x_{i,j} \rightarrow \min \tag{1}$$

Constraints: $$\sum_{j=1}^{n} x_{i,j} = 1 \; \forall \, i = 0, \ldots, n \tag{2}$$

$$\sum_{j=1}^{n} x_{i,j} = 1 \; \forall \, i = 0, \ldots, n \tag{3}$$

$$x_{i,j} \text{ do not form a sub tour} \tag{4}$$

$$x_{i,j} \in \{0,1\} \tag{5}$$

The decision variable $x_{i,j}$ denotes if a connection between v_i and v_j is included in the optimal delivery tour. The *objective function* (1) evaluates the costs of the resulting tour, i.e., the tour that visits every customer node exactly once, which is to be minimized. Constraints (2) and (3) ensure that every customer is visited *exactly once* by allowing only one edge leaving and one edge arriving at each customer. Elimination of *sub tours* occurs by an instantiation of (4). Subtour elimination causes the high level of complexity of TSP solution, cf. Richter (2005); Domschke (1997). Since the focus is on heuristics in the following, subtour elimination is not discussed in detail here.

Although the problem formulation of the TSP is rather simple, computational effort for the generation of the optimal solution grows rapidly with increasing problem size. From an enumeration point of view, $(n - 1)!$ tours are required in order to represent all possible customer sequences (n denoting the overall number of customers). Thus, the TSP belongs to the group of NP complete optimization problems, i.e., no algorithm is known that is able to find an optimal solution within polynomial bounded time (Lawler et al. 1985).

8.1.1.1 Solution of TSP

The TSP can be solved by *exact* or *heuristic* optimization approaches. As an example for the exact case, Applegate et al. (2007) report on a (symmetric) TSP consisting of 85,900 cities that has been solved to optimality, consuming 136 CPU years on a total of 64 Opteron processors and 192 Xeon processors. In general, most effective exact algorithms are quite complex, with codes on the order of 10,000 lines (Helsgaun 2000). Such algorithms are very demanding of computer power.

Whereas exact approaches solve an optimization problem to optimality in bounded time, heuristics facilitate an approximate solution within reasonable time. To this end, heuristics utilize rules according to the structure of the underlying optimization problem in order to find or improve a solution. Heuristics obtain good solutions, but do not guarantee that the optimal solution is found. They are usually simple and feature (relatively) short run times. Some of the algorithms facilitate solutions that differ only by a few percent from the optimal solution. One of the current challenges, for example, is the solution of the 1,904,711-city instance of locations throughout the world (Cook 2011), which can only be achieved by heuristics.

TSP heuristics can be distinguished into *tour construction* and *tour improvement* heuristics. Tour construction algorithms gradually build a tour by incremental addition of further customers. Tour improvement algorithms improve upon a tour by performing various changes of customer sequences. In the following, a selection of well-known TSP heuristics is presented, being able to calculate a feasible solution for problems of practical size in very limited run time. These heuristics will serve as basic ingredients for the subsequent solution of the time-dependent TSP variant.

```
Algorithm 5: Nearest-neighbor heuristic (NN)
Input:        G = (V,E,D) (v_0 ∈ V is the depot node), result tour P = φ
Start:        v_current ← v_0;
              V ← V\{v_0};
              P = {v_0};
Processing:   while V ≠ ∅ do
                  v_j ← v_j | min{d_current,j} ∀ j ∈ V;
                  P = {v_0, ..., v_j};                  // append v_j
                  V ← V\{v_j};
Result:       P = {v_0, ..., v_j, ..., v_0};            // append v_0
```

8.1.1.2 Nearest-Neighbor

An example of a very greedy tour construction approach is represented by the NN heuristic. NN determines the order of customers by performing the relative best decision. The corresponding pseudo code is reported in `Algorithm 5`. Here, a tour is initialized by the depot node v_0. Then, remaining customer nodes are appended

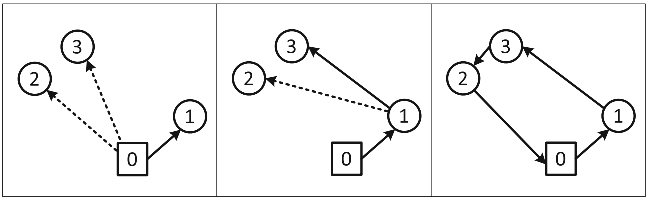


Fig. 8.1 Processing of NN

to the tour so that the distance of the current and the following node is minimized. Finally, the tour is completed by attaching the depot node to the end of the tour.

Exemplary processing of NN is depicted in Fig. 8.1. Possible decisions during processing are depicted by dotted arrows, whereas realized decisions are shown by solid arrows. In the beginning, v_1 is identified as the customer node with the lowest costs starting from the depot node, followed by visits of v_3 and v_2 due to their relatively short distance to the previous customer node.

Obviously, NN is based on myopic decisions and leads to relatively bad approximations of optimal solutions, since integration of "expensive" customers is postponed to the end. Solutions usually differ about 13–17% from the optimum (Golden and Stewart 1985).

8.1.1.3 Insertion Heuristic

A more forward-looking way of the construction of tours is pursued by insertion heuristics. Contrasting NN, the IH heuristic evaluates *insertion costs* of customers, i.e., the costs of a detour resulting from the integration of a further customer. When a customer node v_k is inserted between customer nodes v_i and v_j, the insertion costs $d_{i,k} + d_{k,j} - d_{i,j}$ are calculated. Taking into account all remaining insertion candidates, that v_k is chosen that features the lowest insertion costs. This is referred to as *cheapest insertion* (Solomon 1987). The selection of the next customer can be varied by choosing a random customer (*random insertion*) or by insertion of the customer that is farthest from the current partial tour (*farthest insertion*). In computational experiments, farthest and random insertion feature the best results (Rosenkrantz et al. 1977).

Algorithm 6: Insertion heuristic (IH, cheapest insertion)	
Input:	$G = (V, E, D)$ ($v_0 \in V$ is the depot node), result tour $P = \phi$
Start:	$V \leftarrow V \backslash \{v_0\}$;
	$v_k \leftarrow v_k \big\vert \min\{d_{0,k} + d_{k,0}\} \forall k \in V$;
	$V \leftarrow V \backslash \{v_k\}$;
	$P = \{v_0, v_k, v_0\}$; // insert v_k
Processing:	while $V \neq \phi$ do // investigate remaining nodes
	$v_k \leftarrow v_k \big\vert \min\{d_{i,k} + d_{k,j} - d_{i,j}\} \forall k \in V,\ \forall i \in P$;
	$P = \{v_0, \ldots, v_i, v_k, v_j, \ldots, v_0\}$; // insert v_k

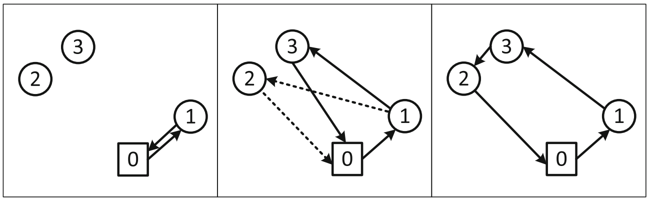


Fig. 8.2 Processing of IH

The pseudo code of the cheapest insertion variant is shown in `Algorithm 6`. A tour is initialized by the insertion of the customer node v_k that features the lowest costs with respect to the depot. Remaining customers are inserted at that position of the tentative tour where they effect a cost-minimal detour.

Processing of IH is shown in Fig. 8.2. The exemplary tour is initialized by a pendulum tour to customer node v_1 (v_0, v_1, v_0). During processing, v_3 is chosen as cheapest insertion candidate due to $d_{1,3} + d_{3,0} - d_{1,0} < d_{1,2} + d_{2,0} - d_{1,0}$. In general, IH features a superior solution quality than NN due to its more intelligent construction process.

8.1.1.4 Savings Algorithm

SAV, in its basic form also known as Clarke and Wright heuristic, is one of the most popular route construction heuristics. It is widely used in practice for TSP as well as for VRP solution. SAV follows the idea of serving each customer individually by a pendulum tour first (Clarke and Wright 1964). Pendulum tours are then merged according to potentially high savings denoted by a precalculated savings list.

In particular, a merger of two tours leads to the saving $s_{i,j} = d_{i,0} + d_{0,j} - d_{i,j}$, i.e., a reduction of costs due to disappearing depot connections minus a detour to the new partial tour. In asymmetric TSPs, the merger of pendulum tours $\{v_0, \ldots, v_k, \ldots, v_0\}$ and $\{v_0, \ldots, v_l, \ldots, v_0\}$ results either in $\{v_0, v_1, \ldots, v_k, v_l, \ldots, v_m, v_0\}$ or $\{v_0, v_l, \ldots, v_m, v_1, \ldots, v_k, \ldots, v_0\}$ due to the direction of subtours. In general, savings are larger when customers i and j are closer to each other and far away from the depot.

The pseudo code of SAV is shown in `Algorithm 7`. In the beginning, the k pendulum tours P_k are initialized. Then, savings $s_{i,j}$ are calculated, sorted in descending order and stored in the savings list S. Afterwards, savings are processed, aiming at the realization of potential high savings first. If tours can be merged, the genuine tours are omitted and the merged tour is retained. In the end, a tour comprising all customer nodes remains.

Processing of SAV is exemplified in Fig. 8.3. In the beginning, three pendulum tours exist. $s_{3,2} > \max(s_{1,2}, s_{1,3}, s_{2,1}, s_{2,3}, s_{3,1})$ leads to the merge of $\{v_0, v_2, v_0\}$

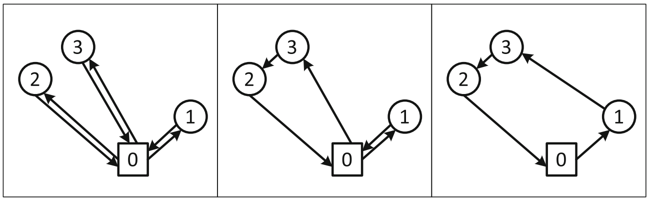


Fig. 8.3 Processing of SAV

and $\{v_0, v_3, v_0\}$. The two remaining partial tours $\{v_0, v_3, v_2, v_0\}$ and $\{v_0, v_1, v_0\}$ are merged afterwards.

```
Algorithm 7: Savings heuristic (SAV)
Input:        G = (V,E,D) (v_0 ∈ v is the depot node),
              L set of tentative tours, S list of savings values
Start:        V ← V\{v_0};
              For each v_0 ∈ V do
                 P_k = {v_0, v_k, v_0};        // initialize pendulum tour
                 L ← L ∪ P_k;
                 For each P_i, P_j ∈ P|i ≠ j
                    s_{i,j} = d_{i,0} + d_{0,j} − d_{i,j};
                    S ← S ∪ s_{i,j};
                 Sort S in descending order;
Processing:   While S ≠ φ
                 Read first s_{i,j} from S;
                 If v_i ∈ P_l and v_j ∈ P_m are on the ''border'' of
                    P_l and P_m
                    P_q ← P_l ∪ P_m;            // merge tours as shown above
                    L ← L ∪ P_q;                // insert new tour
                    L ← L\{P_l, P_m};           // delete individual tours
                 S ← S\{s_{i,j}};
Result:       L contains the result tour;
```

SAV exists in a number of variants. Paessens (1987), for example, introduces a parameterized form of savings calculation by means of

$$s_{i,j} = d_{i,0} + d_{0,j} - g \times d_{i,j} + f \times \left|d_{i,0} - d_{0,j}\right| \text{ with } 0 < g \leq 3 \text{ and } 0 \leq f \leq 1.$$

Thus, the impact of the distance between customers or the distance between depot and customers is adjusted. If $g < 1$, the likelihood of combination of close customers decreases with decreasing g. If $f < 0$, savings decrease with increasing difference in customers' distances from the depot. With $g = 1$ and $f = 0$, we have the classical form of savings calculation.

In general, SAV is easy to code and calculations are fast (Cordeau et al. 2002). Sophisticated implementation variants feature an average deviation of 6.71% from the best known solution. However, the algorithm is based on a greedy principle and contains no mechanism to undo early, possibly deficient route mergers.

8.1.1.5 2-Opt

As an example for an improvement heuristic, the 2-opt approach is introduced. Like every improvement heuristic, the 2-opt heuristic expects a feasible tour as input. Then, this tour is optimized by local search algorithms, i.e., simple changes of the order of customer nodes. 2-opt belongs to the group of *n*-opt heuristics, which improve a solution by exchange of *n* edges. Actually, 2-opt and 3-opt implementations are used in order to limit computational efforts.

2-opt examines the neighborhood of a given tour *P*. The 2-opt neighborhood is defined by all tours that differ in exactly two edges from *P*. The idea is to systematically replace two edges with two other edges so that the resulting tour features lower overall costs. *P* is then replaced by the improved tour and investigations are continued.

An exemplary 2-opt move of edges $(i,i+1)$ and $(j,j+1)$ is depicted in Fig. 8.4. Here, the order of nodes is reversed between v_{i+1} and v_j. Note that for asymmetric TSPs the reversal of edge directions may induce a change of edge costs. The functionality of 2-opt is reported by pseudo code description in `Algorithm 8`.

`Algorithm 8: 2-opt heuristic`

`Input:`	`tour` $P = \{v_0, \ldots, v_i, v_{i+1}, \ldots, v_j, v_{j+1}, \ldots, v_0\}$
`Processing:`	`While exchange is possible`
	`select two non-adjacent edges` $(i, i+1)$ `and` $(j, j+1)$ `from` P;
	`If` $d_{i,i+1} + \sum_{n=i+1}^{j-1} d_{n,n+1} + d_{j,j+1} > d_{i,j} + \sum_{n=i+1}^{j-1} d_{n+1,n} + d_{i+1,j+1}$
	$P = \{v_0, \ldots, v_i, v_j, \ldots, v_{i+1}, v_{j+1}, \ldots, v_0\}$;
	// `(with reversed order between` v_j `and` v_{i+1}`)`
`Result:`	`2-optimized tour` P;

---

n-opt variants with larger *n* are able to generate superior solutions than the 2-opt heuristic, but suffer from a complexity of $O(m^n)$. Lin and Kernighan (1973) introduce a variable *n*-opt algorithm that examines at each iteration step if an interchange of *n* edges may result in an improved tour. Helsgaun (2000) sums up the basic ideas of the Lin-Kernighan heuristic and reports on an implementation for a symmetric TSP.

Static TSP formulation and corresponding solution approaches have set the stage for advanced routing in city logistics. In the following, the time-dependent variant is introduced, i.e., static heuristics are transferred to the time-dependent TSP.

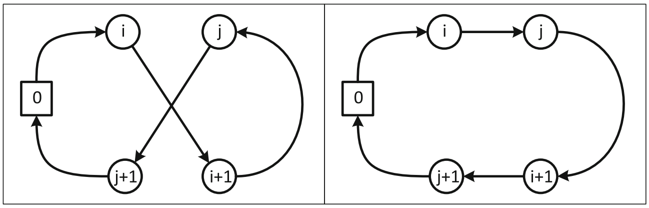


Fig. 8.4 2-opt move

8.1.2 Time-Dependent Traveling Salesman Problem

Time-dependent optimization models represent an urban, congested environment more accurately than their static counterparts, but they are more difficult to solve. The capability of taking into account time-dependent travel times is extremely valuable, not only because varying speeds affect the objective function, but also because the best solutions for a static problem are suboptimal when realized in a dynamic context (van Woensel et al. 2008). In the following, the TDTSP is formulated mathematically, and heuristics for the static TSP are enhanced to the dynamic, time-dependent case.

The TDTSP is defined as follows: Let $G = (V, E, D(t))$ be a complete, directed, and evaluated graph consisting of customer and depot nodes $V = \{v_0, v_1, \ldots, v_n\}$ and edges $E = \{(i,j)|i,j \in V, i \neq j\}$. v_0 represents the depot node, whereas remaining vertices represent customer nodes. A time-dependent distance matrix $D(t)$ contains the costs $d_{i,j}(t)$ that arise when traveling from customer node v_i to customer node v_j at time t. For routing in city logistics, the entries of $D(t)$ correspond to time-dependent costs in terms of time-dependent travel times. Time-dependent distance matrices are precalculated by TDSPP (cf. Sect. 6.3.2).

The TDTSP aims at the determination of the optimal route by minimization of costs $d_{i,j}(t)$ resulting from visiting each customer node exactly once, starting at the depot at a given time t. The main difference between TDTSP and TSP is the consideration of time-dependent distance matrices that feature different travel times for an edge according to a set of discrete time bins. Here, the arrival time at customer v_n depends on the departure time t at customer v_{n-1}.

8.1.2.1 Solution of TDTSP

Literature on the TDTSP is rather scarce. The focus is usually either on the time-independent variant of the TSP, or time dependence is discussed in the context of the VRP (cf. Sect. 8.2). Compact formulations of the TDTSP are discussed by Fox et al. (1979). Gouveia and Voß (1995) model time-dependent machine sequencing by TDTSP formulations. Bentner et al. (2001); Schneider (2002); Li et al. (2005)

focus on time-dependent variants of the “Bier127” problem, investigating the optimal delivery of 127 beer gardens in and around Augsburg, Germany. Bentner et al. (2001); Schneider (2002) generate tours using simulated annealing, whereas Li et al. (2005) adapt complex local search procedures for the heuristic solution of the problem. Haghani and Jung (2005) refer to a genetic algorithm in order to solve a TDTSP in a dynamic context of stochastic customer demands and time-dependent travel times. Computational experiments based on an artificial road network show that time-dependent travel times are very useful in a dynamic context.

Most of the literature on TDTSP computation focuses on well-defined, artificial networks. Comparison of individual results is difficult, because solution approaches differ with regard to problem instances and investigated networks. Thus, basic principles of static TSP computation are resumed in the following, and the extension to TDTSP solution is outlined. The focus is on the suitability of well-known heuristics for optimization of delivery tours in city road networks. In addition to TSP heuristics already presented, also a new composite heuristic is introduced.

8.1.2.2 Time-Dependent Nearest-Neighbor

The TDNN heuristic represents a natural extension of NN, considering a time-dependent distance matrix for the construction of a tour. In TDNN computation, the key point is the determination of the customer node causing the shortest delay at the tentative end of the tour. The delay can be derived from the travel times provided by the entries of the time-dependent distance matrix corresponding to the desired time bin.

For TDNN solution, the general process as described in `Algorithm 5` is still valid (except for the structure of the underlying distance matrix). Well-known advantages of NN such as its speed and simplicity remain. Also disadvantages persist: TDNN results in rather inefficient delivery tours due to expensive customers being postponed to the end of a tour. In sum, TDNN is able to immediately react to time-dependent travel times, but tour construction suffers from the myopic approach.

8.1.2.3 Time-Dependent Nearest-Neighbor Based on Dynamic Programming

Malandraki and Dial (1996) incorporate the principle of TDNN and improve its solution quality. The search process is intensified and good solutions are retained by means of a Dynamic Programming (DP) approach. To alleviate exponential explosion of time and storage requirements, they introduce a DP-oriented heuristic that computes a TDTSP tour with the earliest return time to the depot considering the desired departure time.

DP refers to an exact optimization method which aims at the optimization of multi-level decision problems (Bellman and Dreyfus 1962). Decisions are based on *states,* which describe the amount of information that is relevant for decision making. The number of possible decisions determines the size of the state space. A transition between two states leads to transition costs. DP aims at the determination of decisions that minimize overall transition costs.

The TSP can be modeled as multi-level decision problem. Corresponding decisions regard the customer that is to be visited next, defining the order of delivery. In case of the TDTSP, transition costs are given by time-dependent travel times. Malandraki and Dial (1996) define the minimum transition costs T^* of n states recursively by

$$T^* = \min_{p\in\{1,\ldots,n-1\}}\left[T(\{v_1,\ldots,v_{n-1}\},p) + d_{p,0}(t_p)\right],$$

which corresponds to the earliest return time to the depot given a departure time T_0. T^* depends on the transition costs of predecessor states

$$T(V,k) = \min_{p\in V\setminus\{k\}}\left[T(V\setminus\{v_k\},p) + d_{p,k}(t_p)\right],$$

which define a state space as possible transition costs raised by immediate visit of k after p based on previous calculations. The origin state referring to the first decision after departing from the depot is given by

$$T(\{v_k\},k) = T_0 + d_{0,k}(T_0)\forall\, k = 1,\ldots,n-1,$$

where T_0 is the departure time at the depot, and $d_{0,k}(t_0)$ is the time-dependent travel time from v_0 to v_k, denoted as a function of the time-dependent departure time from v_0. Hence, $T(\{v_k\},k)$ represents the transition costs in terms of the arrival time at v_k.

The DP formulation presented above leads to an exponential explosion of time and storage requirements. Thus, Malandraki and Dial (1996) introduce a "restricted" DP heuristic that retains only the H most promising partial tours at each state. In particular, only the H partial tours with the smallest arrival times $T(V,k)$ with respect to the last node v_k of each partial tour are retained for the next stage. If $H = 1$, the heuristic corresponds to TDNN.

The functionality of the algorithm is depicted in Fig. 8.5. In this example, investigations are limited to the $H = 3$ best partial tours of each state. States are separated by dashed lines. The third state, for example, has a state space size of 6, represented by six partial tours. Only the three partial tours with the relatively earliest arrival times are retained (marked by gray shadowing of tentatively last nodes), avoiding the complete enumeration of all possible solutions.

Pseudo code of TDNNDP is given in `Algorithm 9`. In the starting phase, $T(\{v_k\},k)$ is calculated, which corresponds to arrival times P_k^{arr} of all tours P_k that depart from the depot at time T_0 in order to visit the first customer v_k. These k partial tours are then processed for the computation of each state space. Each

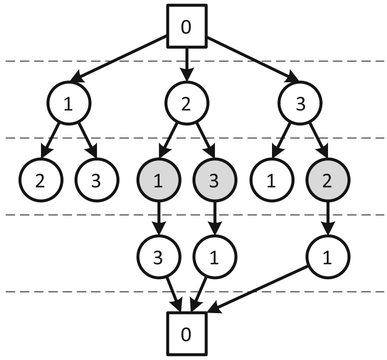


Fig. 8.5 Concept of TDNNDP for an example with $H = 3$

partial tour P_k is amended by one more unrouted customer, leading to a state space size $|T| = |T_{old}| \times (|V| - 1 - |L|)$, with $|L|$ denoting the number of existing partial tours. Partial tours are sorted according to $T(V,k)$, and only the H best partial tours are retained. When all states have been explored, partial tours are completed by the depot node, and sorted. The best partial tour denotes T^*, the earliest arrival time at the depot.

```
Algorithm 9: Time-Dependent Nearest-Neighbor DP heuristic (TDNNDP)
Input:        G = (V, E, D(t)), T_0 = departure time, list of tours L
Start:        V ← V\{v_0};
              For each v_k ∈ V do
                  Initialize partial tour P_k = {v_0, v_k};
                  P_k^arr = d_{0,k}(T_0);                      // T({v_k}, k)
                  L = L ∪ P_k;
Processing:   For each k ∈ |V| do
                  For each P_j ∈ L do                          // T(V, k)
                      v_l ← last node of P_j;
                      v_n ← v_n | min{ d_{l,n}(P_j^arr) } ∀ n ∈ V\P_j;
                      P_j = {v_0, ..., v_l, v_n};
                      P_j^arr = d_{l,n}(P_j^arr);
                  Maintain H shortest tours P_j ∈ L;
Result:       For each P_j ∈ L do
                  P_j = {v_0, ..., v_l, v_0};
                  Sort P according to P_j^arr;
              Return first element of L;                      // T*
```

TDNNDP offers the user to choose some “middle ground” between an optimal DP solution and computational effort required for heuristic solution. It does not guarantee optimality, because only a user specified number of partial tours is retained at each state, featuring a trade-off between solution quality and increasing computational effort. In contrast to the fast computation of TDNN, TDNNDP demands for a vast number of travel time lookups in order to calculate transition costs. Malandraki and Dial (1996) state computational complexity with $O(n^2 H \log_2 H)$. H can be set dynamically according to available RAM and run time limits based on the overall number of customers (e.g., $H = 1,000,000/(|V| - 1)$). In an example of 20 customers, this would lead to 20 partial tours in the start phase, followed by a state space size of 380, 6,840, and 116,280 until $H = 50,000$ would limit the size of the state space to a maximum of 50,000 $\times$ 16 partial tours.

8.1.2.4 Time-Dependent Insertion Heuristic

The technical adaptation of IH to the time-dependent case is as straightforward as the adaptation of NN. Instead of static costs, time-dependent distance matrices provide travel times required for the determination of insertion costs. Nonetheless, due to the time dependence of travel times, the insertion of a customer node may lead to subsequent changes of travel times. Hence, a simple comparison of insertion costs ("delta comparison") is not sufficient in order to evaluate the customer node that causes the cheapest detour. Instead, the insertion of a customer requires the calculation of the shortest delay with respect to the arrival time at the depot. Therefore, travel times of all subsequent edges must be updated.

Although the technical adaptation of IH might be simple, the suitability of the algorithm for time-dependent networks is questionable. The evaluation of detours occurs in a static way, i.e., the evaluation of an insertion is based on travel times of the current time bin only. Since scheduled departure times are subject to change in the course of tour construction, a customer visit might be realized in an unfavorable time bin. The explicit anticipation of favorable time bins is not supported.

8.1.2.5 Combination of TDNNDP and TDIH

TDNNDP performs an exhaustive search in the solution space according to the extended requirements of time-dependent input data, whereas TDIH follows a more intelligent search principle, ignoring potentially favorable time bins. Advantages of both algorithms are now combined in terms of the TDIHDP heuristic.

TDIHDP is based on the improvement of pendulum tours according to the travel times of the current time bin. In particular, all possible insertion positions of an unrouted customer are investigated, which corresponds to the algorithmic concept of TDIH. In order to limit computational efforts, only the H best tours are retained for further improvement, which matches up with the TDNNDP principle. In terms of DP, transition costs are defined by time-dependent insertion costs known from TDIH, whereas a state refers to all possibilities of insertion of an unrouted customer into a set of tentative tours. TDIHDP results in a large variety of tentative tours, ensuring the diversification of the search process.

The pseudo code of TDIHDP is given in `Algorithm 10`. For initialization, n pendulum tours $\{v_0, v_k, v_0\}$ are generated and stored in a set of tentative tours P. The set of tentative tours is improved iteratively. In every step, all possibilities of inserting one more customer node in each tentative tour P_j are investigated, resulting in a new set of tentative tours that have been enhanced by one unrouted customer. Since the investigation of all insertion possibilities leads to a huge number of tentative tours, only the H best tentative tours are maintained. Tours are retained according to their tentative arrival time P_k^{arr} at the depot.

`Algorithm 10: Combination of TDIH and TDNNDP heuristic (TDIHDP)`

`Input:`	$G = (V, E, D(t))$, T_0 = `departure time`, L `set of tentative tours` P_k
`Start:`	$V \leftarrow V \backslash \{v_0\}$;
	`For each` $v_k \in V$ `do`
	$\quad P_k = \{v_0, v_k, v_0\}$;
	$\quad P_k^{arr} = T_0 + d_{0,k}(T_0) + d_{k,0}(T_k)$;
	$\quad L = L \cup P_k$;
`Processing:`	`For` $k = 0,\ k < \lvert V \rvert$ `do` `// for all states`
	$\quad$ `For each` $P_j \in L$ `do // enhance existing partial tours`
	$\qquad L = L \backslash \{P_j\}$;
	$\quad$ `For each` $v_n \notin P_j$ `and all insertion positions do`
	$\qquad P_q = \{v_0, \ldots, v_n, \ldots, v_0\}$; `// insert` v_n
	$\qquad P_q^{arr} = T_0 + \sum_{i=0}^{n-1} d_{i,i+1}(T_{i-1}), n = \lvert P_q \rvert - 1$
	$\quad L = L \cup P_q$;
	$\quad$ `Sort` L `in descending order;`
	`Maintain first` H `tours` $P_q \in L$;
`Result:`	`Return best` $P_j \in L$;

TDIHDP benefits from its convenient adaptation to time dependence by following the TDNN principle, simultaneously considering a multitude of tour variants. Computational effort increases significantly due to the large number of insertion possibilities. Thus, the maximum number of tour variants H should be chosen with care.

8.1.2.6 Time-Dependent Savings

Similar to IH, SAV follows a successful principle of tour construction. Though, SAV cannot be extended to the time-dependent case without modifications. As described above, SAV expects the precalculation of a savings list which denotes attractive mergers. This approach cannot represent time-dependent variation of mergers.

For TDTSP computation, an adapted version of TDSAV is implemented instead. Here, average savings are considered, which in fact corresponds to a static solution of the optimization problem. Average savings are calculated for an estimated planning horizon, which can be provided by a simple heuristic such as TDNN. TDNN generates a tour that stretches over n different time bins. Savings are then calculated based on the average savings corresponding to these n time bins, i.e.,

$$\overline{s_{i,j}} = \frac{1}{n}\sum\nolimits_{m=1}^{n} \left(d_{0,i}(T_m) + d_{j,0}(T_m) - d_{i,j}(T_m)\right),$$

with T_m denoting the departure time according to the boundary of time bin m.

Nonetheless, savings should adapt to the time bin they are expected to be realized, i.e., they should be valid for the time bin when the corresponding merger is expected to be realized. However, we do not know in advance when a merger finally occurs, since this decision is interlinked with the savings value itself. This might lead to the situation where a merger with an estimated high savings value is unlikely to be realized, preventing the determination of a more favorable tour. This problem may be alleviated by consideration of information on the time a merger is likely to be realized, for example, when customer time windows are known. Gietz (1994); Fleischmann et al. (2004) discuss the implementation of SAV in the context of time-dependent travel times and customer time windows (cf. Sect. 8.3.2).

8.1.2.7 Time-Dependent 2-opt

In order to complete the discussion on time-dependent heuristics for the routing of a single vehicle, the adaptation of 2-opt is sketched. As in the static case, improvement heuristics may be quite useful in order to improve relatively poor solutions facilitated by TDNN and TDIH. Again, the technical adaptation is straightforward, but the evaluation of an exchange becomes more expensive.

Especially, an exchange of two (or more) edges cannot be evaluated by a simple delta comparison, leading to an increase of computational efforts. An exchange of two edges $(i, i+1)$ and $(j, j+1)$ inevitably results in a change of travel times between node i and the depot node at the end of a given tour. As in the static case, this is due to costs raised by new edges (i,j) and $(i+1, j+1)$ as well as the reversed edges between $(i+1)$ and $(j+1)$. Also travel times of edges being positioned behind the exchange operation, i.e., behind customer node v_{j+1}, have to be recalculated due to possibly changing departure and arrival times, leading to increasing computational efforts compared to the static variant.

8.2 Routing of a Fleet of Vehicles

Routing of a fleet of vehicles expects both the efficient assignment of customers to delivery tours and the cost-optimal ordering of customers according to distances or travel times. Based on the previous section, the VRP is discussed in the following, which comprises the integrated assignment and ordering of customers to (freight) vehicles. First, background on the static variant of the VRP is given, which is then extended to the time-dependent case. VRP and TDVRP optimization models are briefly summarized. Subsequent discussions on customer time windows will give more details with respect to particular algorithms and time-dependent extensions.

8.2.1 Vehicle Routing Problem

The VRP aims at the generation of a set of tours for a fleet of vehicles under various constraints. Each customer is visited exactly once by a specific vehicle which delivers the demanded amount of goods to the customer. The VRP usually minimizes the overall distance or the overall travel time of a set of tours. In static-deterministic variants of the VRP, demand and location of customers as well as distances and travel times between them are assumed to be known in advance.

Similar to the TSP, the VRP is one of the most studied combinatorial optimization problems. Many different variants of this problem have been formulated to ensure a suitable application to a number of real world cases. To this end, a large variety of constraints and details of the problem have been taken into account. Application areas concern pickup and delivery problems, transport of people, dial-a-ride problems, heterogeneous transport fleets, multiple depots, and the consideration of customer time windows, time-dependent travel times or planning in a dynamic context due to stochastic travel times, or stochastic customer demands. In the following, the focus is on a standard definition, which is extended to the time-dependent case subsequently.

Let $G = (V, E, D)$ be a complete, directed, evaluated graph consisting of nodes $V = \{v_0, v_1, \ldots, v_n\}$ and edges $E = \{(i,j)|i,j \in V, i \neq j\}$. v_0 represents the depot, whereas remaining nodes represent customers. A distance matrix D contains the costs $d_{i,j}$ that arise when traveling from customer node v_i to customer node v_j. The entries of D represent static costs $d_{i,j}$ such as distances or average travel times. Then, the VRP aims at the determination of the optimal tour plan where (1) every tour is starting and terminating at the depot, (2) each customer is visited exactly once by one vehicle, and (3) the tour plan is at optimal cost. If $d_{i,j} = d_{j,i} \forall\, i,j$, the problem is said to be symmetric; otherwise, it is asymmetric.

According to Toth and Vigo (2002), the mathematical formulation for the asymmetric VRP is as follows. Here, A denotes a subset of V, facilitating subtour elimination. Q represents the maximum capacity of a vehicle, and b_i denotes the demand of customer v_i.

$$\textit{Variables}: \quad x_{i,j} = \left\{ \begin{array}{c} 1 \text{ if tour plan includes edge from } i \text{ to } j \\ 0 \text{ else} \end{array} \right\}$$

$$\textit{Objective function}: \sum\nolimits_{i=0}^{n-1} \sum\nolimits_{j=i+1}^{n} d_{i,j} x_{i,j} \rightarrow \min \tag{1}$$

$$\textit{Constraints}: \quad \sum\nolimits_{h=0}^{i-1} x_{h,i} + \sum\nolimits_{j=i+1}^{n} x_{i,j} = 2 \, \forall \, i = 1, \ldots, n \tag{2}$$

$$\sum\nolimits_{i \in A} \sum\nolimits_{\substack{j \in V \backslash A \\ i<j}} x_{i,j} + \sum\nolimits_{i \in V \backslash A} \sum\nolimits_{\substack{j \in A \\ i<j}} x_{i,j} \geq 2r(A) \forall A \in$$

$$V \backslash \{v_0\} \text{ and } |A| \geq 2 \; with \; r(A) = \left\lceil \frac{\sum_{i \in A} b_i}{Q} \right\rceil \tag{3}$$

$$x_{i,j} \in \{0, 1\} \forall \, i, j = 1, \ldots, n; i<j \tag{4}$$

$$x_{i,j} \in \{0, 1, 2\} \forall \, i = 1, \ldots, n \tag{5}$$

The objective function (1) leads to the tour plan with the lowest overall costs by minimization of the sum of edge costs. Constraint (2) ensures that each customer node is visited exactly once. Condition (3) prevents subtours except for pendulum tours and ensures that the maximum capacity Q of a vehicle is not exceeded. This occurs by consideration of the demand b_i for each customer $v_i \in A$. (4) and (5) denote the domain of the variables.

8.2.1.1 Solution of VRP

Lots of algorithms and solution methodologies for the static VRP exist. The interested reader is referred to Laporte (1992), who presents a survey on VRP formulations as well as exact and heuristic solution approaches. A more recent overview on exact and heuristic solution methods as well as application areas can be found in Toth and Vigo (2002). Recent advances are available in a handbook by Golden et al. (2008). A comparison of classical and recent heuristics with regard to accuracy, speed, simplicity, and flexibility is provided by Cordeau et al. (2002). In the following, a brief overview on solution approaches is given.

Similar to the TSP, the VRP can be solved exactly or by heuristics (cf. Fig. 8.6). DP and branch and bound algorithms establish an optimal solution in a limited number of steps. The concept of *DP* with respect to the solution of the TSP has already been discussed in Sect. 8.1.2. Solomon (1987) presents a DP framework for the solution of VRPs. *Branch and bound* algorithms divide a problem into partial problems ("branching") and compute bounds for the partial problems ("bounding", Domschke 1997). Partial solutions are ordered within a tree. One branch of this tree (and corresponding partial solutions) can be precluded from subsequent search if no solution of this branch can lead to an improvement of

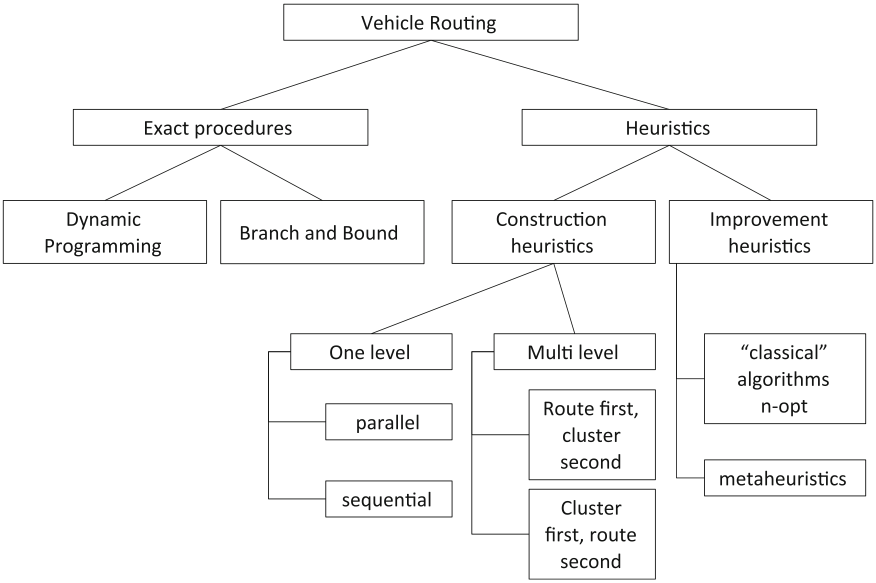


Fig. 8.6 Solution methods for VRP (following Richter 2005; Aberle 2009)

solutions already available. Exact approaches like DP and branch and bound require lots of computational effort and are not suitable for the solution of problems of practical size. Thus, the focus is on heuristics in the following.

VRP heuristics can be distinguished into *construction* and *improvement heuristics*. Construction heuristics select nodes (or edges) sequentially until a feasible solution is created. Nodes are chosen according to some cost minimization criterion, often subject to the restriction that the selection does not create a violation of vehicle capacity or temporal constraints. If one aims at the simultaneous solution of customer assignment and customer ordering, algorithms are assigned to the group of *one level* procedures. One level procedures can be discerned into *parallel* and *sequential* algorithms. Sequential methods construct one route at a time, while parallel methods build several routes simultaneously. Depending on the order of solutions, *multi-level* approaches distinguish *route-first—cluster-second* and *cluster first—route second* procedures. Route-first—cluster-second algorithms determine a "giant route" by solving of a TSP for all customers first. In the second step, capacity and time restrictions are considered. This leads to a split-up into feasible tours, which are then reoptimized individually. Cluster-first—route-second algorithms first group customers in subsets according to capacity and time restrictions. Subsequently, a TSP is solved for each subset.

Contrasting the TSP, *improvement heuristics* do not only improve existing solutions by reordering of customers within a particular tour ("intra tour procedures"). Instead, they alter the assignment of customers by exchanging customer nodes or partial tours. *n-opt improvement* exchanges n edges within a given tour plan by new edges as long as an improvement of the objective function value is achieved. If no improvement occurs, the procedure stops, often remaining in a local optimum. This disadvantage can be overcome by *metaheuristics*, which permit a temporary decline of the objective function value.

Metaheuristics are general solution procedures that explore the solution space to identify good solutions. They often embed some of the standard route construction and improvement heuristics. Solution quality of metaheuristics is usually better at the cost of higher computational effort due to a more thorough search in the solution space. They comprise a diverse set of methods such as simulated annealing, genetic algorithms, *Tabu Search* (TS), ant colony optimization, and constraint programming. Within metaheuristics for the VRP, TS clearly dominates other solution strategies (cf. Sect. 8.3.2). The idea behind TS is to perform a local search by moving from one solution to the best solution of the neighborhood. Since this move may cause the objective function to deteriorate, an anti-cycling mechanism declares some previous solutions forbidden ("tabu") for a number of iterations.

Many of the algorithms suited for TSP optimization can easily be adapted to VRP solution. One of the most prominent examples is SAV. The basic principle has already been introduced in Sect. 8.1.1. In case of VRP solution, individual tours arise by consideration of temporal or capacity restrictions such as the maximal duration of a tour or the maximum capacity of a vehicle. Tours are only merged if a capacity or a temporal restriction can be complied with. SAV belongs to the group of parallel construction heuristics, since customer assignment and ordering of customers is solved simultaneously.

In the following, the focus is on the enhancement of the static VRP to the time-dependent VRP.

8.2.2 Time-Dependent Vehicle Routing Problem

Static VRP models are not able to consider time-dependent variations of edge costs. For routing in city logistics, a time-dependent formulation is required, ensuring adequate incorporation of time-varying travel times between customer locations. Ignoring time dependence might lead to a suboptimal solution with a different route structure and a different number of required vehicles than would result from a time-dependent optimal solution. The degree of infeasibility of static solutions rises with the increase of the degree of time dependence, and static solutions, even if they might seem to be superior, are usually infeasible (Donati et al. 2008). If the static solutions are feasible, they are often suboptimal in time-dependent contexts.

Contrasting the vast number of publications on the static VRP, the literature on the time-dependent variant is still rather scarce. On the one hand, adequate travel time data was not available in the past, which has recently changed due to ongoing advances in telematics (cf. Chap. 5). On the other hand, the TDVRP is harder to model and harder to solve than the VRP, because a time-dependent topology is required and because well-known principles known from non-temporal optimization do not hold (Ichoua et al. 2003; Malandraki and Daskin 1992). So far, most research in this area has focused on dynamic routing and scheduling considering variation in customer demands instead of dynamic travel times.

The TDVRP can be defined as follows: Let $G = (V, E, D(t))$ be a complete, directed, evaluated graph consisting of nodes $V = \{v_0, v_1, \ldots, v_n\}$ and edges $E = \{(i,j)|i,j \in V, i \neq j\}$. v_0 represents the depot, whereas remaining nodes represent customers. A time-dependent travel time matrix $D(t)$ contains the costs $d_{i,j}(t)$ that arise when traveling from customer node v_i to customer node v_j using the corresponding edge at time t. The entries of $D(t)$ denote time-dependent costs in terms of time-dependent travel times. They are generated by TDSPP computation (cf. Sect. 6.3.2). Once the time bin during which an edge is traversed is known, the travel time for this edge becomes a known constant. The TDVRP aims at the determination of the optimal tour plan where (1) every tour is starting at the depot at a given time T_0 and is terminating there, (2) every customer is visited exactly once by one vehicle, (3) the number of tours as well as total travel time is minimized.

Based on Figliozzi (2009), the TDVRP can be formulated as follows, considering a set of vehicles K, a set of customers $D = V \backslash \{v_0\}$, and a customer demand b_i for each customer node v_i:

$$\textit{Variables}: \quad x_{i,j}^k = \begin{Bmatrix} 1 \text{ if vehicle } k \text{ travels between customers } i \text{ and } j \\ 0 \text{ else} \end{Bmatrix}$$

$$\textit{Objective function}: \quad \sum\nolimits_{k \in K} \sum\nolimits_{j \in D} x_{0,j}^k \rightarrow \min \tag{1}$$

$$\sum\nolimits_{k \in K} \sum\nolimits_{i,j \in E} d_{i,j}^k(t) x_{i,j}^k \rightarrow \min \tag{2}$$

$$\textit{Constraints}: \quad \sum\nolimits_{i \in D} b_i \sum\nolimits_{j \in V} x_{i,j}^k \leq Q \, \forall \, k \in K \tag{3}$$

$$\sum\nolimits_{k \in K} \sum\nolimits_{j \in V} x_{i,j}^k = 1 \, \forall \, i \in D \tag{4}$$

$$\sum\nolimits_{i \in V} x_{i,l}^k - \sum\nolimits_{i \in V} x_{l,j}^k = 0 \, \forall \, l \in D, \; \forall \, k \in K \tag{5}$$

$$\sum\nolimits_{j \in V} x_{0,j}^k = 1 \, \forall \, k \in K \tag{6}$$

$$\sum\nolimits_{j \in V} x_{j,0}^k = 1 \, \forall \, k \in K \tag{7}$$

$$x_{i,j}^k \in \{0,1\} \forall \, (i,j) \in E, \forall k \in K \tag{8}$$

The primary and secondary objectives are defined by (1) and (2), respectively. Constraints are defined as follows: the sum of customer demand delivered by vehicle k must comply with the maximum vehicle capacity (3), all customers must be served exactly once by one vehicle (4), if a vehicle arrives at a customer it must also depart from that customer (5), tours must start and end at the depot (6–7), the decision variable is binary (8). Figliozzi (2009) also considers customer time windows, which have been omitted here. The TDVRP is usually handled as an asymmetric problem.

8.2.2.1 Solution of TDVRP

Due to the VRP being an optimization problem of high complexity, the solution of the TDVRP also occurs by heuristics for problems of practical size. In the following, a literature overview on TDVRP solution is provided. The solution of the TDVRP is strongly related to the modeling of the underlying topology, i.e., computational effort for solution as well as solution quality depend on complexity and accuracy of time-dependent information models. Due to the impact of time-dependent travel times on costs and service quality of resulting tours, the TDVRP is often discussed in the context of customer time windows. Approaches with focus on customer time windows are brought up in Sect. 8.3.2.

Beasley (1981) discusses a very simple variant of the TDVRP, considering two periods of the day with different travel times. He adapts SAV correspondingly. Hill et al. (1988); Hill and Benton (1992) provide the nodes of a road network with time-dependent, piecewise-constant speeds and derive the travel time of an edge from the average speeds of incident nodes. They report on an application for a courier service for 15 branches of a bank.

Malandraki and Daskin (1992) introduce a sequential route construction heuristic for the TDVRP. Starting with an empty vehicle at the depot, they append customers in a nearest-neighbor manner until temporal requirements or capacity restrictions come into effect. If a vehicle is "full", it returns to the depot and a new vehicle is filled until all customers are served. They also discuss a simultaneous version of this route construction heuristic, where that unrouted customer is added to a tour that features the smallest travel times, considering temporal requirements and capacity restrictions. Computational tests are based on problems with 10–25 customers and two or three time bins per edge.

Gietz (1994); Fleischmann et al. (2004) are the first who utilize time-dependent travel times from a traffic information system. They model piecewise time-dependent travel time functions for usage in adapted construction heuristics similar to SAV and IH. Solutions are improved by a 2-opt approach. Comparisons of static travel times and time-dependent travel times show that static optimization underestimates travel times in a range of 10%. With only five time bins, a rather good approximation of the "true" travel times is achieved, which is not significantly improved by 10 time bins. Computational efforts grow modestly for SAV and increase by a factor of 2–3 for IH combined with local improvement algorithms. A growth of the number of time bins does not lead to higher computational efforts; instead, limits are set by the required storage space for distance matrices.

Ichoua et al. (2003) refer to a parallel TS heuristic in order to solve the TDVRP. They enhance an approach by Taillard et al. (1997) who partition the VRP by geographical and distance-based criteria. Partial problems are then solved by a cluster of computers. A starting solution is generated by an insertion algorithm by random selection of nodes. Underlying problems are derived from variants of Solomon's Euclidean 100 customer problem (Solomon 1987), enhanced by FIFO consistent, time-dependent travel times arranged in three time bins for three different road types. Computational experiments on static as well as on time-dependent travel times show that consideration of time dependence is beneficial.

Van Woensel et al. (2008) also refer to a TS approach. Their time-dependent travel times result from queuing models. They implement a TS based on Gendreau et al. (1994), Gendreau et al. (1996); Hertz et al. (2000). Solutions are calculated by 2-opt and adapted improvement heuristics that shift the departure time at the depot in order to achieve further improvements. Based on problems by Augerat et al. (1998), which contain between 32 and 80 customers, results show that total travel times can be improved significantly by time-dependent optimization. A larger number of time bins enhances solution quality. They state that the "impact of dynamic components will be even more important when relating the approach to urban contexts".

Similar to TDTSP evaluation, the comparison of the solution quality of TDVRP approaches is rather difficult, since problems vary in network structure and underlying data. Figliozzi (2009) is the first who proposes a set of time-dependent benchmark problems based on Solomon's VRP benchmarks in order to enable comparisons of different optimization procedures for TDVRP solution.

8.3 Customer Time Windows

Routing in city logistics requires the consideration of customer time windows in the optimization of delivery tours, especially for innovative applications such as attended home delivery (cf. Chap. 3). On the one hand, customers suppose narrow and reliable time windows. On the other hand, customer time windows exacerbate optimization due to additional constraints. Here, a compromise between service quality and service costs must be found. In the following, the VRP with customer time windows (VRPTW) is introduced and formulations and heuristics for the consideration of time-dependent travel times are discussed (TDVRPTW). In particular, the implementation of a state-of-the-art metaheuristic is reported. This completes the time-dependent optimization framework for routing in city logistics.

8.3.1 Vehicle Routing Problem with Time Windows

Service quality of delivery services strongly depends on the reliability of time windows assured to customers. From an optimization point of view, this is reflected by VRPTW optimization models, which are one of the most important extensions of the VRP. For the VRPTW, customer time windows are modeled as additional constraints. The VRPTW has been widely studied in the last 20 years, see Desrochers et al. (1988); Solomon and Desrosiers (1988); Koskosidis et al. (1992); Savelsbergh (1992); Desrosiers et al. (1995); Potvin and Bengio (1996); Taillard et al. (1997). It has a wide range of applications such as bank and postal deliveries, school bus routing, industrial refuse collection, franchise restaurant services, security patrol services, and just-in-time manufacturing. Especially heuristic and metaheuristic approaches have been investigated intensively; see Bräysy and Gendreau (2005b); Hashimoto et al. (2008).

The definition of the VRPTW is similar to the VRP, since the VRPTW may be seen as a special case of the VRP with additional constraints. Additionally, for each customer node v_i, a customer time window $[e_i, l_i]$ may be defined, with e_i being the earliest time and l_i being the latest time for start of service at customer v_i. Then, the VRPTW aims at the determination of the optimal tour plan where (1) every tour departs and terminates at the depot, (2) every customer is visited once by one vehicle, (3) the number of tours is minimal, and (4) every customer is served within its time window. A secondary objective is to either (5) minimize the total distance traveled or the total travel times of the particular routes. Results are ranked according to a hierarchical objective function, where the number of tours is considered as the primary objective, and for the same number of tours, the secondary objective is either the total distance traveled or the total duration of tours. A solution requiring fewer tours is often considered better than a solution with more tours, regardless of the total distance traveled.

Customer time windows can be *soft* or *hard*. For the case of soft customer time windows, the limits of a time window may be exceeded, which is then penalized in the objective function. For the hard case, limits of a customer time window must be strictly complied with. Here, customer time windows are modeled in terms of additional constraints. For the latter, even just finding a feasible schedule with a given number of vehicles is known be to a complex, NP complete optimization problem (Hashimoto et al. 2008).

A mathematical formulation for the VRPTW can be found in Figliozzi (2010). Since the VRPTW is an extension of the VRP, similar solution approaches apply. Bräysy and Gendreau (2005a, b) give comprehensive surveys on route construction, local search, and metaheuristic approaches for VRPTW solution. In the following, principles of most important heuristics are sketched. They serve as ingredients for the subsequent implementation of a time-dependent VRPTW solution approach.

8.3.1.1 Construction Heuristics

Solomon (1986) solves the VRPTW by a route-first cluster-second approach. First, the order of customers is optimized in a giant tour, which is then divided into a number of smaller tours.

Solomon (1987) introduces several heuristics for the VRPTW:

- SAV is enhanced by parameters describing the temporal and spatial closeness of customers.
- A variant of NN is proposed, which initializes every tour by determination of the closest unrouted customer with respect to the depot. In each iteration, the heuristic appends the customer closest to the last customer of a tour. A new tour is initialized whenever the search fails to compute a feasible insertion position.
- The sequential insertion heuristic *I1* initializes every tour with a "seed" customer. Unrouted customers are inserted into this tour until it is "full" with respect to capacity or temporal constraints. The selection of customers occurs according to criteria c_1 and c_2. c_1 denotes the best feasible position of insertion for an unrouted customer. c_2 evaluates the benefit of insertion, which is derived from the difference of the distance between a customer and the depot and overall insertion costs. A detailed discussion of such an insertion mechanism is conducted in Sect. 8.3.2.
- The sweep heuristic by Gillett and Miller (1974) is extended. To this end, the VRPTW is decomposed into a clustering stage and a scheduling stage. During clustering, customers are assigned to vehicles similar to the original sweep heuristic. A center of gravity is computed and customers are positioned according to their polar angle. In the scheduling phase, the order of customers is optimized based on I1 functionality.

Potvin and Rousseau (1993) present a parallel insertion variant of Solomon's I1 heuristic, where a set of routes is initialized simultaneously. They generate pendulum tours based on the customers that are the farthest from the depot ("seed customers"). The selection of the customer to be inserted next is based on a *regret measure*. A large regret measure means that there is a large gap between the best insertion place for a customer and the best insertion places in the other tours, enforcing immediate processing. Different parameter sets allow for the generation of a set of solutions. The minimal number of tours is found by iteratively reducing the number of tours based on the initial solution. Details for a time-dependent variant of this algorithm are presented in Sect. 8.3.2.

8.3.1.2 Improvement Heuristics

Improvement heuristics enhance a solution by exploring neighboring ones. A multitude of neighborhood operators for VRP and VRPTW solution exist, see Bräysy and Gendreau (2005b). In the following, the most prominent ones are

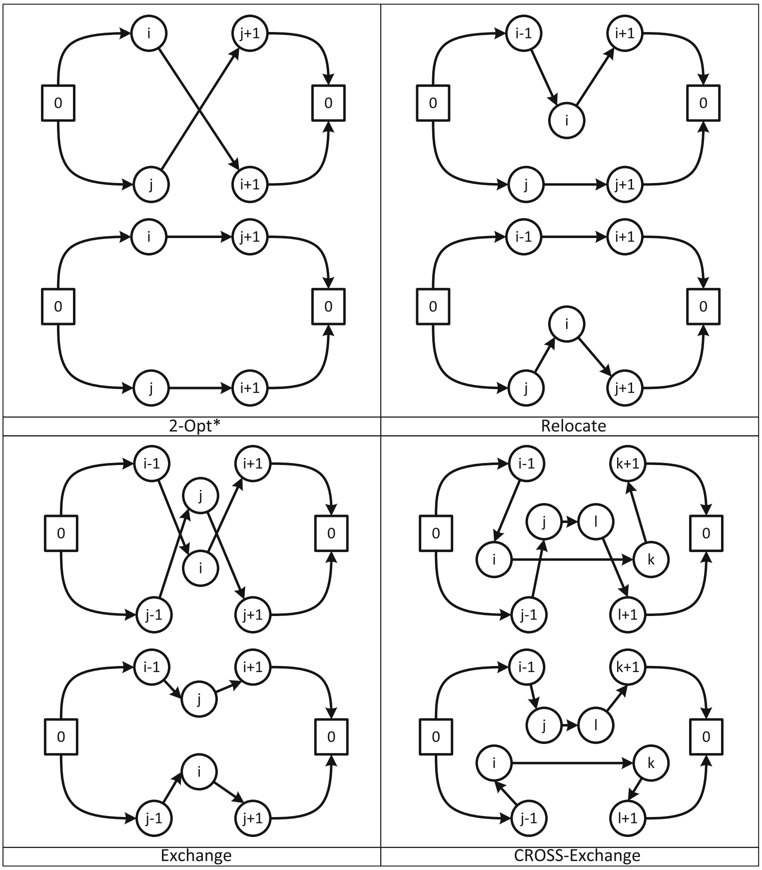


Fig. 8.7 Common neighborhood operators for VRPTW solution (adapted from Bräysy and Gendreau 2005b)

summarized. They mainly differ in the size and complexity of resulting neighborhoods:

- The most common solution approach for intra-tour improvements are n-opt neighborhoods, which replace a set of n edges by another set of n edges. Verifying n-optimality demands $O(m^n)$ computational time. Feasibility has to be checked after each improvement. The *Or-opt operator*, for example, replaces three edges in the original tour by three new ones without modifying the orientation of the route (Or 1976). This kind of operators is also suited for TSP optimization, cf. Sect. 8.1.
- Improvement by an exchange of customers between several tours is induced by inter-tour operators (cf. Fig. 8.7). Potvin and Robillard (1995) introduce the *2-opt** exchange operator, which combines two tours so that the last customer of a given tour is shifted after the first customer of another tour. *Relocate* shifts a

customer from one tour between two nodes of another tour. The *Exchange operator* swaps two nodes in different tours. The *CROSS-Exchange* operator removes two edges $(i-1,i)$ and $(k,k+1)$ from a first tour, while two edges $(j-1,j)$ and $(l,l+1)$ are removed from a second tour. Then the segments $i-k$ and $j-l$, which may contain an arbitrary number of customers, are swapped by introducing the new edges $(i-1,j),(l,k+1),(j-1,i)$ and $(k,l+1)$.

8.3.1.3 Metaheuristics

Contrasting previous approaches, metaheuristics are more solution concepts than solution procedures. They allow objective function values to deteriorate and even infeasible intermediate solutions in the course of the search process, and incorporate common route construction and improvement heuristics. Bräysy and Gendreau (2005a) review prominent metaheuristics in terms of TS, genetic algorithms and a variety of other metaheuristics. Metaheuristics comprise (1) a representation of the problem, (2) the generation of an initial solution, (3) the computation of new solutions based on a neighborhood of a solution, (4) the evaluation of solutions, and (5) a control structure that coordinates the single components. In the following, the focus is on TS, since it serves as a framework for subsequent time-dependent optimization.

The general idea of TS is depicted in Fig. 8.8. TS builds on an initial solution which is produced by an arbitrary heuristic, for example, a local search procedure. In each iteration, TS explores the solution space by moving from the current solution to the best solution in a subset of its neighborhood. The neighborhood of a solution is induced by a *neighborhood operator*. Neighborhood operators correspond to simple local search heuristics as presented above. Contrary to common heuristics, the current solution may deteriorate comparing an iteration with the next. Inferior solutions are temporarily accepted in order to avoid solutions already investigated. This ensures that new areas of the solution space are considered in order to avoid local minima and to ultimately find an improved solution. To avoid cycling, solutions or moves equal to recently explored solutions are temporarily declared tabu or forbidden. *Tabu tenure* denotes the duration a solution or a move remains tabu. The tabu status can be overridden if certain conditions are met; this is called the *aspiration criterion*. This may happen if a tabu declared solution is superior to any previous solution. The *termination criterion* defines when the search is terminated, for example, after a number of iterations or a run time limit.

A first TS approach for VRPTW solution has been established by Garcia et al. (1994). They create an initial solution by Solomon's I1 insertion heuristic and refer to 2-opt* and Or-opt exchanges for improvement. Since then, authors have presented numerous TS approaches involving sophisticated diversification and intensification techniques, strategies for minimizing the number of routes, complex post-optimization techniques, hybridizations with other search techniques such as simulated annealing and genetic algorithms, parallel implementations, and allowance of

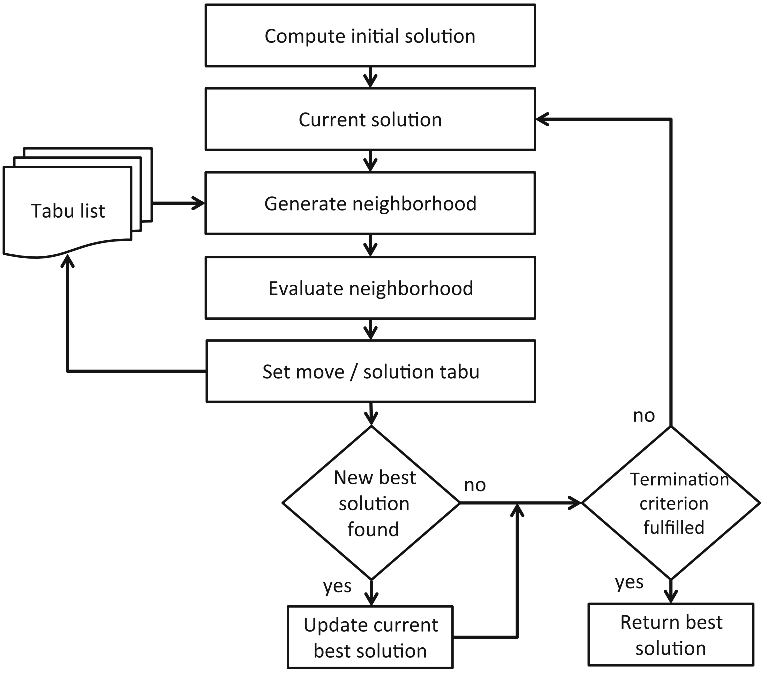


Fig. 8.8 Functionality of Tabu Search

infeasible solutions in the search process. The quality of solutions obtained with different metaheuristic techniques is often superior to traditional construction heuristics and local search algorithms. Though, parameterization and control of single components are very complex with respect to algorithm usage and evaluation.

8.3.2 Time-Dependent Vehicle Routing Problem with Time Windows

Time-dependent travel times require significant modifications to local search approaches and metaheuristics that have been successfully applied to the static VRPTW (Figliozzi 2009; Maden et al. 2010). The insertion of a customer or a local improvement step does not only affect the arrival and departure times of a local subset of customers, but it may also significantly change travel times among other customers. Additionally, the impact of altering a route is not just local but potentially affects all subsequent travel times. Changes in travel times have an impact on the feasibility of solutions, especially with respect to feasibility of customer time windows.

The TDVRPTW can be defined as follows: Let $G = (V, E, D(t))$ be a complete, directed, evaluated graph consisting of nodes $V = \{v_0, v_1, \ldots, v_n\}$ and edges

$E = \{(i,j)|i,j \in V, i \neq j\}$. v_0 represents the depot, whereas remaining nodes represent customers. For each customer node v_i, a time window $[e_i, l_i]$ may be defined, with e_i being the earliest time and l_i being the latest time for start of service at customer v_i. A time-dependent travel time matrix $D(t)$ represents the costs $d_{i,j}(t)$ that arise when traveling from customer node v_i to customer node v_j at time t. The entries of $D(t)$ denote time-dependent costs in terms of time-dependent travel times. Based on the formulation of the TDVRP (cf. Sect. 8.2.2), Figliozzi (2009) presents an arc-flow formulation of the TDVRPTW with hard time windows, considering a set of vehicles K, a set of customers $D = V \backslash \{v_0\}$, a service time g_i for and a demand b_i required by customer v_i. The destination depot is referenced as v_{n+1}. A simplified variant ignoring distance costs is as follows:

$$Variables: \quad x_{i,j}^k = \begin{Bmatrix} 1 \text{ if vehicle } k \text{ travels between customers } i \text{ and } j \\ 0 \text{ else} \end{Bmatrix}$$

$$y_{i,j}^k = \textit{service start time for customer } i \textit{ served by vehicle } k$$

$$Objective\,function: \sum\nolimits_{k \in K} \sum\nolimits_{j \in D} x_{0,j}^k \rightarrow \min \tag{1}$$

$$\sum\nolimits_{k \in K} \sum\nolimits_{j \in D} \left(y_{n+1}^k - y_0^k\right) x_{0,j}^k \rightarrow \min \tag{2}$$

$$Constraints: \sum\nolimits_{i \in D} b_i \sum\nolimits_{j \in V} x_{i,j}^k \leq Q \,\forall\, k \in K \tag{3}$$

$$\sum\nolimits_{k \in K} \sum\nolimits_{j \in V} x_{i,j}^k = 1 \,\forall\, i \in D \tag{4}$$

$$\sum\nolimits_{i \in V} x_{i,l}^k - \sum\nolimits_{i \in V} x_{l,j}^k = 0 \,\forall\, l \in D, \ \forall\, k \in K \tag{5}$$

$$x_{i,0}^k = 0, \ x_{n+1,i}^k = 0 \,\forall\, i \in V, \ \forall\, k \in K \tag{6}$$

$$\sum\nolimits_{j \in V} x_{0,j}^k = 1 \,\forall\, k \in K \tag{7}$$

$$\sum\nolimits_{j \in V} x_{j,n+1}^k = 1 \,\forall\, k \in K \tag{8}$$

$$e_i \sum\nolimits_{j \in V} x_{i,j}^k \leq y_i^k \,\forall\, i \in V, \ \forall\, k \in K \tag{9}$$

$$l_i \sum\nolimits_{j \in V} x_{i,j}^k \geq y_i^k \,\forall\, i \in V, \ \forall\, k \in K \tag{10}$$

$$x_{i,j}^k \left(y_i^k + g_i + \mathrm{d}_{i,j}\left(y_i^k + g_i\right)\right) \leq y_i^k \,\forall (i,j) \in E, \ \forall\, k \in K \tag{11}$$

$$x_{i,j}^k \in \{0,1\} \forall (i,j) \in E, \ \forall\, k \in K \tag{12}$$

$$y_j^k \in R, \forall\, i \in V, \ \forall\, k \in K \tag{13}$$

The primary and secondary objectives are defined by (1) and (2), respectively, ensuring the minimization of the number of tours (1) as well as the minimization of total travel times (2). Constraints are as follows: the maximum capacity of a vehicle Q must be complied with (3), and all customers must be served exactly once by one vehicle k (4). If the vehicle k arrives at a customer l, it must also depart from customer l (5). Tours must start and end at the depot (6-8). The most important constraints regard the satisfaction of service times (9,10). At this point, the earliest start time y_i of service at customer i must be later or equal to e_i (9), and the latest start time y_i must be earlier or equal to l_i (10). Service times y_i must allow for traveling between customers (11). Decision variables' type and domain are indicated in (11) and (12). Figliozzi (2009) also presents a formulation for the consideration of soft time windows.

8.3.2.1 Solution of TDVRPTW

The TDVRPTW is usually solved by heuristics. In the following, most important approaches are summarized. For a more comprehensive review, Flatberg et al. (2005) distinguish the recent literature on dynamic, stochastic, and time-dependent TDVRP formulations.

Ahn and Shin (1991) present implementations of insertion and local improvement algorithms and report on computational experiments for up to 200 customers. Ichoua et al. (2003) refer to a parallel variant of TS and the utilization of the exchange operator. They approximate the lateness and its impact on travel times, since an exact delta evaluation is expensive in time-dependent contexts. The approximation is used to quickly reject all but k moves from the neighborhood, followed by exact evaluation of the remaining moves. Modified Solomon problems serve as a test bed for the evaluation of their heuristic. For the case with highest degree of time dependence, the objective value is reduced by $9-18\%$ compared to the usage of static travel times. Almost all constant speed solutions turn out to be infeasible in the time-dependent context.

Fleischmann et al. (2004) assume travel times to be known between all pairs of relevant locations and constant within individual time bins. Transitions between time bins are smoothed to ensure a FIFO behavior of travel time functions (cf. Sect. 6.2.2). They discuss efficient feasibility checks, evaluations, and updates of tours within time-dependent contexts. Based on traffic data collected in Berlin, Germany, they calculate tours by adapted SAV, IH, and 2-opt heuristics.

Haghani and Jung (2005) refer to a genetic algorithm to solve a dynamic VRP with time-dependent travel times. The problem is formulated as mixed integer linear program. Solution methods are tested on small random test problems. The genetic algorithm is also tested on a more realistic case with a simulated network considering variation of travel times and dynamic orders arriving regularly. New plans are generated at regular intervals based on time-dependent travel times. Results show that the advantage of a dynamic routing strategy increases as the dynamics in the problem increase.

Donati et al. (2008) rely on a "Multi Ant Colony System" published by Gambardella et al. (1999). Here, two ant colonies are combined to find a TDVRPTW solution. The first colony minimizes the number of tours, whereas the second colony minimizes overall distance and overall travel times. Pheromones are modeled with relation to time, reflecting the time-dependent travel time of an edge. Evaluation and pheromone updates are based on the overall travel times of a tour instead of distances. Whenever one of the colonies has found a new best solution, a global pheromone update adjusts the edge-related information, ensuring that both colonies benefit from new information. Underlying travel time data originate from empirical traffic data of northern Italy. On average, the framework leads to an improvement of 7.58% compared to the static case.

Hashimoto et al. (2008) refer to a local search algorithm. They modify standard neighborhoods in terms of 2-opt*, cross exchange and Or-opt and implement a DP heuristic that utilizes information from past recursions to evaluate the current solution. The search space is limited by a filtering method in order to avoid solutions having no prospect of improvement.

Figliozzi (2009) is the first who publishes benchmark problems dedicated to the TDVRPTW. He investigates the benchmark problems by a route construction and a route improvement heuristic. An arc-flow formulation of the TDVRPTW with hard and soft time windows is given. To a certain degree, soft time windows ameliorate the computational burden and loss of efficiency introduced by time-dependent travel times. Hard time windows are more difficult to accommodate.

The most recent paper dealing with TDVRPTW solution has been published by Maden et al. (2010). Their *LANTIME heuristic* corresponds to a metaheuristic approach. The general functionality is depicted in Fig. 8.9. An initial solution is created using the *parallel insertion heuristic* by Potvin and Rousseau (1993), which is based on *Solomon's I1 heuristic* (Solomon 1987). The I1 algorithm computes an estimation of the number of tours and produces parameters for the initialization of the parallel insertion heuristic. Parallel insertion then generates an initial solution, which is improved by the LANTIME heuristic.

In the following, the approach by Maden et al. (2010) is adapted to the requirements of routing in city logistics. Therefore, the single components are introduced in detail. They are implemented into the time-dependent optimization framework and instantiated with time-dependent information models.

8.3.2.2 I1 Heuristic (Adapted from Solomon 1987)

The sequential insertion heuristic I1 initializes a pendulum tour, considering the customer that is farthermost at the time of departure from the depot. As long as unrouted customers exist, the best position of insertion is determined for each unrouted customer u in terms of insertion costs between customers i and j:

$$c_1(i,u,j) = \min\left[c_1\left(i_{p-1},u,i_p\right)\right], p = 1,\ldots,m.$$

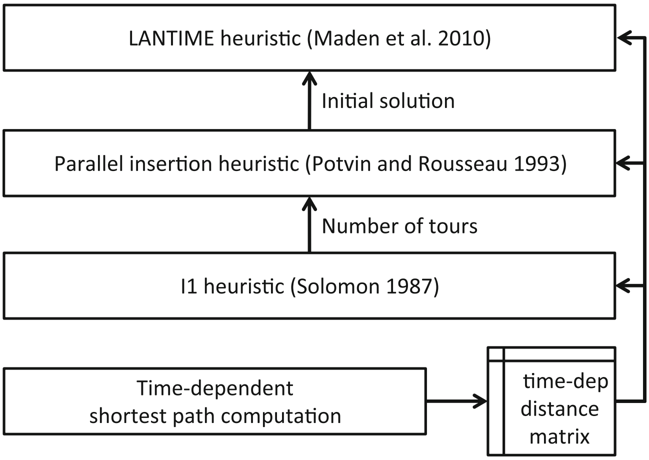


Fig. 8.9 Functionality of LANTIME heuristic

Insertion costs c_1 arise as linear combination of c_{11} and c_{12}, denoting the costs of a detour due to insertion of customer u, and the costs of the resulting delay:

$$c_1(i,u,j) = \alpha_1 c_{11}(i,u,j) + \alpha_2 c_{12}(i,u,j), \alpha_1 + \alpha_2 = 1, \alpha_1, \alpha_2 \geq 0.$$

In contrast to static environments, the *costs of the detour* c_{11} depend on the sum of travel times between the insertion position and the depot, compared to the previous sum of travel times, i.e.,

$$c_{11}(i,u,j) = d_{i,u}(T_i) + d_{u,j}(T_u) + d_{j,x}(T_j) + \ldots - d_{i,j}(T_i) - d_{j,x}(T_j) - \ldots$$

with T_i denoting the end of service, i.e., departure time at customer i. Here, a local investigation of insertion costs would be inadequate, since the insertion of a customer node may lead to changing departure times at the following customer locations. Increasing travel times, however, may be absorbed by waiting times at other customer locations.

Delay costs c_{12} are computed as

$$c_{12}(i,u,j) = b_{k_u} - b_k,$$

with k being the first customer featuring a customer time window after j, and b_k denoting the service start time at i.

Subsequently, that customer is chosen that features the largest insertion benefit

$$c_2(i,u,j) = \max[c_2(i,u,j)] | u \textit{ is feasible and unrouted}.$$

The insertion benefit may be quantified as follows, resulting in a higher benefit the larger the distance to the depot and the smaller c_1:

$$c_2(i,u,j) = \lambda d_{0,u}(T_0) - c_1(i,u,j), \lambda \geq 0.$$

The corresponding pseudo code is shown in `Algorithm 11`.

```
Algorithm 11: Time-dependent I1 heuristic
Input:        G = (V, E, D(t)); set of tours L
Start:        V ← V\{v_0};
              L = φ;
Processing:   While V ≠ ∅ do
                  P_k = {v_0, v_k, v_0};        // Initialize partial tour
                  Repeat
                      Determine insertion position based on c_1 ∀u ∈ V;
                      Select insertion candidate based on c_2 ∀u ∈ V;
                      if insertion of v_u is possible
                          insert v_u at best position;
                          V ← V\{v_u};
                  Until no more insertion is possible into P_k;
                  L = L ∪ P_k;
Result:       Return L;
```

8.3.2.3 Parallel Insertion Heuristic (Adapted from Potvin and Rousseau 1993)

The parallel insertion heuristic produces a number of tours simultaneously. Therefore, an estimation of the overall number of tours is required. Potvin and Rousseau (1993) utilize Solomon's I1 heuristic to estimate the overall number of tours and to generate seed customers. Seed customers denote the farthest customer in each tour with respect to the depot. They are used to initialize pendulum tours which are then subject to parallel insertion of unrouted customers. Similar to the I1 heuristic, insertion costs for each unrouted customer are computed, considering the possibility of insertion in all existing tours. Subsequently, that customer is chosen that features the largest benefit of insertion. For evaluation, the sum of differences of insertion costs within a tour is compared to insertion costs into the best tour over all tours except the best tour. Pseudo code is given in `Algorithm 12`.

Algorithm 12: Parallel insertion heuristic

Input: $G=(V,E,D(t))$;

Start: $V \leftarrow V \backslash \{v_0\}$;

$I \leftarrow$ solution from I1 heuristic with $\lambda=1, \alpha_1=1, \alpha_2=0$;

$n_r \leftarrow$ number of tours in I;

$S_{seed,n_r} \leftarrow$ customer of each tour in I with largest distance to depot;

Processing:

While $n_{r\ \min}$ is not known do

  Initialize $S_{(\alpha_1,\alpha_2),n_r} \leftarrow \{(0.5,0.5),(0.75,0.25),(1.0,1.0)\}$;

  $(\alpha_1,\alpha_2) \leftarrow$ Random parameter set removed from $S_{(\alpha_1,\alpha_2),n_r}$;

  $FS \leftarrow$ DetermineFeasibleSolution$\left(n_r, \alpha_1, \alpha_2, S_{seed,\, n_r}\right)$;

  If $FS \neq \emptyset$ then

    $FS_{best} \leftarrow FS$;

    If $n_r \leq$ number of tours in FS then

      $n_r \leftarrow n_r - 1$

      $S_{seed,n_r} \leftarrow S_{seed,n_r+1} - s^*$;

    else

      $n_{r\ \min} \leftarrow n_r$;

  else

    If $S_{(\alpha_1,\alpha_2),n_r} = \phi$ then

      $n_{r\ \min} \leftarrow n_r + 1$;

      else

        $S_{seed,\, n_r+1} \leftarrow S_{seed,\, n_r} + u^*$;

        $n_r \leftarrow n_r + 1$;

While $S_{(\alpha_1,\alpha_2),n_{r\min}} \neq \phi$ do

  $(\alpha_1,\alpha_2) \leftarrow$ Random parameter set removed from $S_{(\alpha_1,\alpha_2),n_{r\ \min}}$;

  $FS \leftarrow$ DetermineFeasibleSolution$(n_{r\ \min}, \alpha_1, \alpha_2, S_{seed,\, n_{r\ \min}})$;

  If $FS \neq \phi$ and FS is superior to FS_{best}

    $FS_{best} \leftarrow FS$;

Return: FS_{best};

In particular, the parallel insertion heuristic builds on an initial solution provided by the adapted I1 algorithm with $\lambda=1$, $\alpha_1=1, \alpha_2=0$. The number of tours and corresponding seed customers are required in the initialization phase. In the processing phase, the minimal number $n_{r\ \min}$ is determined by checking three parameter sets for the computation of insertion costs with respect to n_r tours. If a feasible solution is found, it is checked if also a smaller number of tours was sufficient. If no feasible solution can be found, a larger number of tours is investigated instead. Seed customers are adjusted by removing s^* or amending u^* correspondingly. When the appropriate number of tours has been established, remaining parameter sets are checked with regard to a potentially superior solution.

Algorithm 13: Parallel insertion heuristic: determine feasible solution	
Start:	Initialize n_r pendulum tours P_k based on $S_{Seed,\,n_r}$;
Processing:	While $V \neq \phi$ do
	Determine insertion position based on $c_1 \forall u \in V$;
	Select insertion candidate based on $c_2 \forall u \in V$;
	if insertion of v_u is possible
	insert v_u at best position;
	else return no solution;
Return:	List of tours L;

The computation of a feasible solution occurs as shown in Algorithm 13. For each unrouted customer, the best insertion position with respect to all tours is computed. Then, the appropriate customer is selected for insertion. The determination of the best position of insertion consists of two steps. First, the best position of insertion of customer u between i^r and j^r is computed for each tour r individually:

$$c_1^r(i^r, u, j^r) = \min_{p=1,\ldots,m}\left[c_1^r\left(i_{p-1}^r, u, i_p^r\right)\right], r = 1, \ldots, n_r$$

Then, a tour r' is selected which would provide the lowest insertion costs of customer u with respect to all tours:

$$c_1^{r'}\left(i^{r'}, u, j^{r'}\right) = \min_{r=1,\ldots,n_r}\left[c_1^r(i^r, u, j^r)\right]$$

Finally, that customer is selected for insertion that features the largest benefit of insertion over all unrouted customers, i.e., $c_2(u) = \max_u[c_2(u)]$. The benefit of an insertion of customer u is quantified as

$$c_2(u) = \sum\nolimits_{r \neq r'}\left[c_1^r(i^r, u, j^r) - c_1^{r'}\left(i^{r'}, u, j^{r'}\right)\right].$$

The more the overall insertion costs of customer u differ from insertion costs into its best tour, the more beneficial an insertion becomes. Thus, the insertion of customers that are potentially difficult to handle is favored in order to avoid infeasible solutions.

8.3.2.4 LANTIME Heuristic (Adapted from Maden et al. 2010)

The LANTIME heuristic refers to a metaheuristic based on a TS approach. To this end, LANTIME expects an initial solution, which is produced by the adapted parallel insertion heuristic presented above. This solution initializes x^*, the best feasible solution known. Then, a tabu list, a long-term memory and an iteration count are initialized. During processing, a potentially superior solution with respect to x^* is investigated as long as the *termination criterion* is not accomplished with.

In every iteration, a *neighborhood operator* is selected randomly. It generates the neighborhood of x_{now}, which is subject to subsequent investigation. As a result, x_{now} represents the best solution within the neighborhood, which may be infeasible. The applied move is set tabu on the *tabu list* and noticed in the *long-term memory*. If x_{now} is feasible and superior to x^*, x^* is overridden.

The main functionality of LANTIME is denoted in `Algorithm 14`. In the following, the core parts of the metaheuristic are presented in more detail.

`Algorithm 14: LANTIME heuristic`

`Start:`	`Compute initial solution` x^*;
	$x_{now} \leftarrow x^*$;
	`Initialize tabu list, iteration counter, tally count;`
`Processing:`	`While termination is not complied with`
	`Randomly select a neighborhood operator;`
	$N \leftarrow$ `list of moves resulting from the neighborhood of` x_{now};
	$x_{now} \leftarrow$ `InvestigateNeighborhood(`N`);`
	`Update tally count, tabu list and iteration counter;`
	`if` x_{now} `is feasible and superior to` x^* `then`
	$x^* \leftarrow x_{now}$;
`Return:`	x^*;

The *tabu list* is a core feature of the LANTIME algorithm. It enables the shifting from a local optimum of a neighborhood to a potentially superior solution, preventing the oscillation back to the local optimum in one of the next iterations. Therefore, the tabu list stores information about every move performed. The duration for which a move is set tabu (*tabu tenure*) is set randomly from a predefined interval. Corresponding to Gendreau et al. (1994), a stochastic duration prevents cycling between solutions. If the aspiration criterion is complied with, the tabu list is ignored.

Tabu lists are implemented in a solution-based and in a move-based way. Move-based tabu lists store significant features of the move, for example, customers affected by a move. Solution-based tabu lists store initial solutions that have led to a move, requiring more computational time and more storage space than move-based ones. Nonetheless, solution-based tabu lists may lead to inconsistencies regarding the uniqueness of a move.

Different neighborhoods are generated by *neighborhood operators*. A neighborhood operator induces a list of all possible moves that lead to a neighboring solution. Corresponding to Maden et al. (2010), the following neighborhood operators are considered; graphic examples are presented in Fig. 8.10:

- The *Adapted Cross-Exchange* operator exchanges a number of consecutive customers in a tour with a number of consecutive customers in the same or another tour. This corresponds to a relatively comprehensive neighborhood.

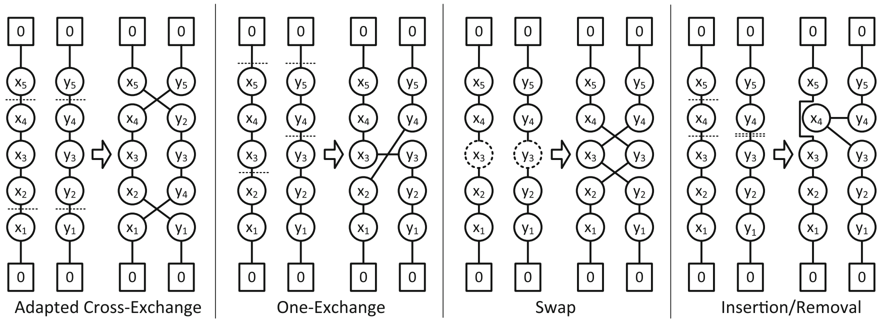


Fig. 8.10 LANTIME neighborhood operators

When a move is executed, it is checked if the reversal of the customer sequence is beneficial for those moved. In Fig. 8.10, the customer sequence $x_2 \ldots x_4$ is exchanged with the customer sequence $y_2 \ldots y_4$, leading to a reversed customer sequence $y_4 \ldots y_2$. It must be ensured that the exchange of customer sequences does not lead to the vanishing of tours.

The following operators can be considered as special cases of the Adapted Cross-Exchange operator:

- The *One-Exchange* operator exchanges all customers behind a certain point of a tour with all customers behind a certain point of another tour. This is exemplified in Fig. 8.10, where the customer sequences $x_3 \ldots x_5$ and y_4, y_5 are swapped. Shifts of customer sequences within a single tour are not allowed. Reversal of customer sequences is checked similar to the Adapted Cross-Exchange operator. Again, it must be ensured that the exchange of customer sequences does not lead to the vanishing of tours.
- The *Swap* operator alternates two customers in a single tour or in two different tours. In Fig. 8.10, customers x_3 and y_3 are exchanged.
- The *Insertion/Removal* operator shifts a customer to another position in the same tour or in another tour. In Fig. 8.10, x_4 is shifted in between y_3 and y_4, leading to a shortening of the first and to an extension of the second tour.

The *investigation of a neighborhood* of x_{now} occurs as presented in `Algorithm 15`. Here, the list N comprises all neighborhood moves generating the neighborhood of x_{now}. When the neighborhood has been investigated, the (possibly infeasible) best solution x_{best} is returned. In particular, a move M is randomly selected and deleted from N, leading to a new solution x_{trial}. If x_{trial} is superior to x^*, $x_{best} = x_{trial}$ is returned. Otherwise, if M has not been set tabu, the following condition will be checked:

$$M(x_{trial})f(x_{trial}) + \alpha P(x_{trial}) < M(x_{now})f(x_{now}) + \alpha P(x_{now}).$$

$M(x)$ refers to a function that evaluates the long-term memory with respect to a given solution, whereas $f(x)$ evaluates a solution in terms of the overall sum of travel times. $P(x)$ corresponds to a measure of infeasibility, whose impact is parameterized by α. If this condition is valid, $x_{best} = x_{trial}$ will be returned, i.e., x_{trial} is evaluated as superior to x_{now}. Otherwise, x_{trial} is compared to the best current solution of the neighborhood in terms of x_{best}:

$$M(x_{trial})f(x_{trial}) + \alpha P(x_{trial}) < M(x_{best})f(x_{best}) + \alpha P(x_{best}).$$

If that is true, the investigation of the neighborhood will be continued with $x_{best} = x_{trial}$.

```
Algorithm 15: LANTIME heuristic: investigate neighborhood
Start:       x_best ← null;
Processing: While N ≠ ϕ do
                M ← Randomly chosen move from N;
                x_trial ← M executed on x_now;
                If x_best = null then
                   x_best ← x_trial;
                If x_trial is feasible and superior to x*
                   Return x_best ← x_trial;
                Else if M or x_trial is not tabu then
                   If M(x_trial)f(x_trial) + αP(x_trial) < M(x_now)f(x_now) + αP(x_now) then
                       Return x_best ← x_trial;
                   Else if M(x_trial)f(x_trial) + αP(x_trial) < M(x_best)f(x_best) + αP(x_best)
                       Then x_best ← x_trial;
Return:      x_best;
```

The LANTIME heuristic also features a *long-term memory* in order to support diversification in the search process. The long-term memory is a data structure that notices accomplished moves ("tally count"), and a function for the evaluation of the tally count with respect to a given move. For each customer, a distinct tally count denotes the sum of partial moves. The evaluation of $M(x_{trial})$ ensures that a solution resulting from an "unconventional" move is preferred, although the solution might be inferior compared to the known best solution. In relation with the measure of infeasibility $P(x)$, intermittent acceptance of inferior solutions is enabled, featuring the diversification of the search process. The parameter α defines the intensity to which $P(x)$ is considered. It is initialized with $\alpha = 1$ and then dynamically adjusted. If all solutions of the last n iterations have been feasible, then $\alpha = \frac{\alpha}{2}$, i.e., feasibility will become relatively unimportant. If all solutions of the last n iterations have been infeasible, then $\alpha = 2\alpha$, i.e., feasibility will become relatively more important.

For the configuration of the *termination criterion,* a number of options exist. In the optimization framework, three variants are implemented:

- The *number of operations* condition allows for the definition of the desired number of iterations of the metaheuristic. It is well suited for the evaluation of a neighborhood operator's performance with respect to average solution quality after a fixed number of operations, for example.

- The *consecutive unsuccessful* condition cancels investigations after a number of unsuccessful operations, i.e., a number of neighborhood investigations leading to any improvement of results. Thus, computational efforts adapt dynamically to the quality of the search process.
- The *timeout* condition allows for the definition of a maximum limit for run time.

The metaheuristic presented above concludes the discussion on time-dependent optimization procedures for routing of a single vehicle, routing of a fleet of vehicles, and for consideration of customer time windows in optimization procedures. Based on this optimization framework, computational experiments for vehicle routing in city logistics are conducted in the following chapter.

Chapter 9
Evaluation of Optimization Models

In this chapter, the usage of optimization procedures for the routing of a single vehicle as well as for a fleet of vehicles is investigated. Based on the evaluation of information models in Chap. 7, FA and FW data serve as input for (time-dependent) heuristics introduced in Chap. 8. The analysis reveals the different potential of heuristics with respect to adaptation to time-dependent information models.

First, the experimental setup for a fictitious logistics service provider in the area of Stuttgart is presented (Sect. 9.1). Experiments focus on the inherent spatio-temporal relations between depot and customers, which is reflected by the capability of heuristics to react to time-dependent input data. The routing of a single vehicle is investigated by several TDTSP heuristics. Solutions are compared with regard to their efficiency and spatio-temporal structure (Sect. 9.2). Computational experiments are then enhanced to the TDVRP and TDVRPTW case (Sect. 9.3). The impact of time-dependent travel times on the reliability of customer time windows and corresponding delivery costs is analyzed in Sect. 9.4.

9.1 Experimental Setup

The evaluation of optimization models requires an adequate experimental setup. Different sets of customer locations are defined by customer scenarios, reflecting the spatio-temporal structure of experiments. Static and time-dependent information models establish input data for (time-dependent) optimization of delivery tours. Results are evaluated regarding solution quality, especially with respect to adaptability to time-dependent input data.

J. F. Ehmke, *Integration of Information and Optimization Models for Routing in City Logistics*, International Series in Operations Research & Management Science 177,
DOI: 10.1007/978-1-4614-3628-7_9,

9.1.1 Customer Scenarios

Corresponding to the experimental setup for the evaluation of information models, the city road network is divided into an inner city area and an outer city area (cf. Sect. 7.3.1; Fig. 7.1). Areas define potential customer locations. Thus, different traffic flow patterns and local variability of urban traffic flows can be considered. Customers are randomly selected from the link table of the core road network as follows:

- Customer Scenario 1 (*inner city*) refers to customers that are situated in the inner city area. This scenario allows for the analysis of time dependence within the fine-grained inner city road network. Here, a rather high customer density is expected.
- Customer Scenario 2 (*outer city*) consists of customers that are situated outside the inner city district, featuring a number of alternatives for passing or avoiding the city center. Vast parts of the routes are expected to follow roads in the outskirts. Since customers may be spread throughout the whole metropolitan area, longer tours accompanied by a rather low customer density are expected.
- Customer Scenario 3 (*entire city*) focuses on the situation where customers are situated in the entire conurbation, i.e., it corresponds to a city logistics service provider delivering to all locations contained in the core road network.

An exemplary delivery tour for Customer Scenario 2 is depicted in Fig. 9.1. An aerial view of the metropolitan area shows depot and customer locations (yellow pins). The tour exemplifies delivery in the afternoon, departing on Tuesday at 13:30. The order of delivery is indicated by round brackets, for example, the depot is denoted by “(0)”, whereas estimated arrival times are highlighted by squared brackets. The inner city area is marked. Since this delivery tour arises from Customer Scenario 2, no customers are located in the inner city area. First, the southern part of the area is visited, followed by a northeast and a final northwest cycle to customers near the depot. The inner city area is passed two times.

9.1.2 Information Models

Information models are selected based on the results of their evaluation in Chap. 7:

- FA information models produce sufficient input data for the static solution of TSP and VRP. As shown above, FA performs on a level of approximately 20% mean RAD between planned and realized tour durations, representing the most reliable static information model for routing in city logistics. For computational experiments, FA data is used by means of a static distance matrix for each day of the week (cf. Sect. 6.1.2).
- FW information models establish input data for the solution of TDTSP and TDVRP. FW performs on a level of 11% mean RAD between planned and

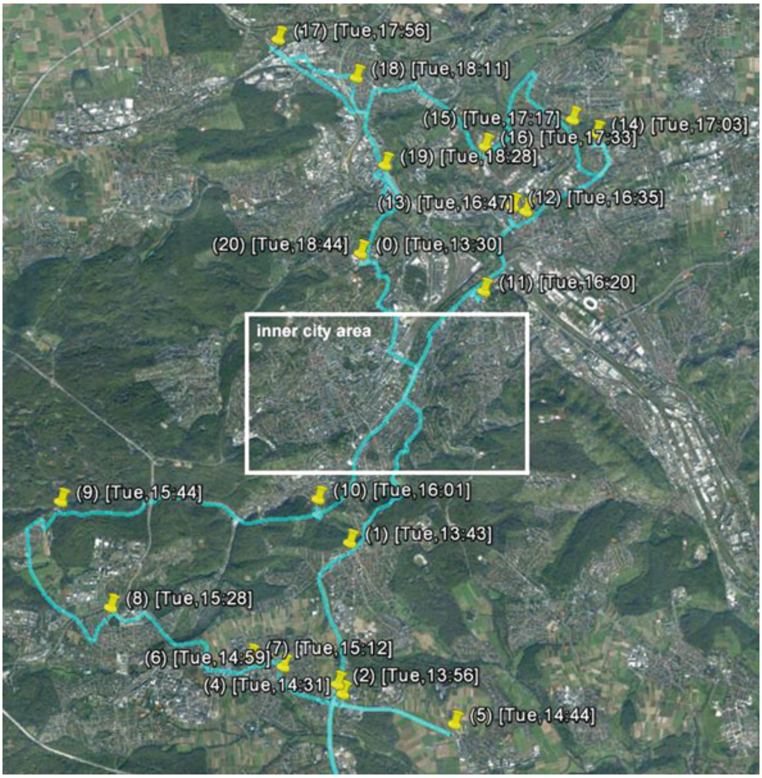


Fig. 9.1 Example delivery tour for Customer Scenario 2 (aerial view with kind permission of GeoContent GmbH, Magdeburg, Germany)

realized tour durations, representing time-dependent variations of travel times. For computational experiments, FW data is used by means of a time-dependent distance matrix featuring 672 time slots (cf. Sect. 6.2.2).

9.1.3 Evaluation of Heuristics

According to Barr et al. (1995) and Cordeau et al. (2002), evaluation of any heuristic method is subject to the comparison of a number of criteria that relate to various aspects of algorithm performance. Examples of such criteria are run time, quality of solution, ease of implementation, robustness, and flexibility. For the analysis of time-dependent heuristics, the following measures are taken into account:

- Different heuristics are compared to each other in terms of overall travel time and the required number of vehicles for a tour plan with respect to 168 consecutive departure times in the course of the week. Also run time is considered in order to relatively determine the most efficient algorithm.
- Since the optimal solution is not known, the best known solution is compared to different individual solutions for a given departure time.
- The impact of time-dependent travel times is analyzed by depicting the evolution of overall travel times in the course of the week (FW data). Results are compared to static routing based on FA data.

Next to well-known performance measures, also the adaptability of individual heuristics to time-dependent travel times is analyzed. Time-dependent travel times might induce altering of a route structure according to their evolution, if optimization procedures are capable of adapting to dynamic input data. For instance, the optimal order of customers in the morning rush hour might completely differ from the order of customers in the evening rush hour.

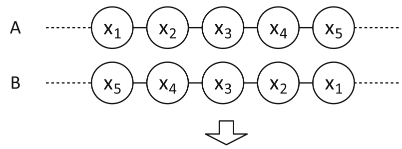
Especially, routes are investigated by comparison of customer sequences. Variation of customer sequences can be quantified by a meaningful comparison measure. Bierwirth et al. (1995) introduce the *Hamming distance* as a measure for the ratio of precedence relations being preserved comparing two routes. Following their presentation for the Job Shop Scheduling problem, a TSP solution can be transformed into a bit-string representing whether a customer v_i is visited previously to customer v_j (i.e., $v_i < v_j \rightarrow 1$) or not (i.e., $v_j < v_i \rightarrow 0$). The inverse relation $v_j < v_i$ is redundant and therefore omitted, i.e., for n customers, a bit string with the number of $(n^2 - n)/2$ bits is computed. Then, the normalized Hamming distance between the bit strings x and y is calculated by $d_{x,y} = \frac{1}{l}\sum_{i=1}^{l} \mathrm{xor}(x_i, y_i)$, where l denotes the length of the bit strings. The Hamming distance denotes the change rate in the order of customers, abstracting from edges between them. Hence, structural evolution of routes can be assessed in order to evaluate the impact of time-dependent input data.

Two examples for the comparison of precedence relations are depicted in Fig. 9.2. Here, the consecutive routes A and B as well as A and C are given, respectively. Precedence relations are noted in a bit field. In Example (1), the order of customer visits is reversed, which leads to the altering of all precedence relations; the corresponding Hamming sum is $d_{A,B} = 1$. In Example (2), the comparison of routes A and C reflects that only customers x_2 and x_4 have been switched, which leads to a change of 30% of precedence relations or a Hamming sum of $d_{A,\,C} = 0.3$.

In the following, optimization models are evaluated in the order of increasing complexity. The main focus is on the analysis of presented heuristics for the routing of a single vehicle (TDTSP). Then, incorporation of time-dependent information models in TDVRP and TDVRPTW optimization procedures is exemplified. For evaluation of the reliability of customer time windows and resulting costs of delivery, an analysis of the impact of time-dependent input data on resulting service quality is conducted.

Fig. 9.2 Comparison of route structures by Hamming distance

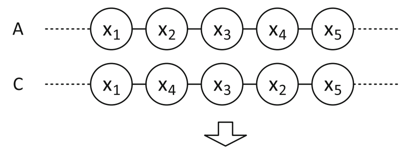
(1)



	$x_1<x_2$	$x_1<x_3$	$x_1<x_4$	$x_1<x_5$	$x_2<x_3$	$x_2<x_4$	$x_2<x_5$	$x_3<x_4$	$x_3<x_5$	$x_4<x_5$
A	0	0	0	0	0	0	0	0	0	0
B	1	1	1	1	1	1	1	1	1	1

$$d_{A,B} = \frac{1}{10}(10) = 1$$

(2)



	$x_1<x_2$	$x_1<x_3$	$x_1<x_4$	$x_1<x_5$	$x_2<x_3$	$x_2<x_4$	$x_2<x_5$	$x_3<x_4$	$x_3<x_5$	$x_4<x_5$
A	0	0	0	0	0	0	0	0	0	0
C	0	0	0	0	1	1	0	1	0	0

$$d_{A,C} = \frac{1}{10}(3) = 0.3$$

9.2 Routing of a Single Vehicle

In this section, static as well as time-dependent routing of a single vehicle are illustrated by computation of the optimal order of customers. For each customer scenario, distance matrices arising from FA and FW information models establish input data for the cost-efficient as well as customer friendly optimization of customer visits. Time-dependent evolution of overall travel times results from modification of the departure time at the depot in 1-hour steps from Monday, 00:30, to Sunday, 23:30.

The following heuristics are used (cf. Sect. 8.1.2):

- TDNN computes the order of customers by a greedy, myopic approach in terms of the currently best decision.
- TDNNDP enhances the simple idea of TDNN by investigation of a ranked order of best decisions. The number of best alternatives H is set to $H = 50000/n$, n representing the number of customers.
- TDIH refers to a cost-efficient insertion of unrouted customers.
- TDIHDP consolidates the principles of TDNNDP and TDIH. The number of best alternatives H is set to $H = 1000000/n$, n representing the number of customers.
- TDSAV refers to preferably good mergers of pendulum tours based on savings values for an estimated planning horizon.
- LANTIME represents a simplified variant of the metaheuristic presented for TDVRPTW solution (cf. Sect. 8.3.2). For TDTSP computation, Adapted

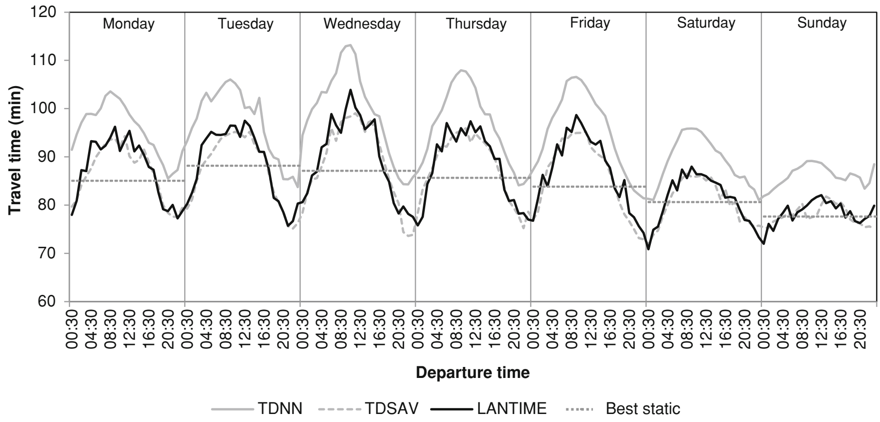


Fig. 9.3 Evolution of overall travel times for Customer Scenario 1

Cross-Exchange, Swap, and Insertion/Removal operators are used. The timeout condition is set to a maximum run time of 10 s.

Results are improved by a time-dependent 2-opt heuristic.

9.2.1 Customer Scenario 1

The investigation of a typical inner city delivery tour occurs in Customer Scenario 1. Here, the optimal order of 40 inner city customer visits is determined for 168 departure times at the depot. Customer service time is set to 10 min per customer.

In Fig. 9.3, overall travel times resulting from 168 departure times are depicted in the course of the week. Different solution quality of heuristics is reflected by three curves, denoting the best (TDSAV), second best (LANTIME) and worst results (TDNN) for this customer scenario. For instance, a departure on Wednesdays at 4:30 leads to a tour with an overall travel time of 107 min, calculated by TDNN. This is contrasted by shorter overall travel times of 92 min (LANTIME) and 88 min (TDSAV). Addition of customer service time (40 × 10 min) would result in the overall tour duration. Since customer service time is fixed, it is omitted here. The dotted gray line denotes the best static solution resulting from weekday specific FA data.

At a first glance, time-dependent variation of overall travel times is clearly visible, underlining the impact of time-dependent FW data. Travel times vary between an overall minimum of 70 min (departure on Saturdays, 0:30) and a maximum of 100 min (departure on Wednesdays, 9:30). They reflect typical variations in urban road traffic, resulting in significantly longer travel times during the day, and relatively short travel times in the evening and at night. At weekends, no extensive variations of tour durations are visible. The average length of a tour is 46.7 km.

Table 9.1 Performance of heuristics (Customer Scenario 1)

Algorithm	Ratio of best known solution found (%)	Average deviation from best known solution (%)	Average run time (s)
TDNN	0	11.3	0.0
TDNNDP	0	9.5	416.8
TDIH	0	6.2	0.2
TDIHDP	1	6.0	483.0
TDSAV	65	0.5	0.1
LANTIME	34	1.6	10.0

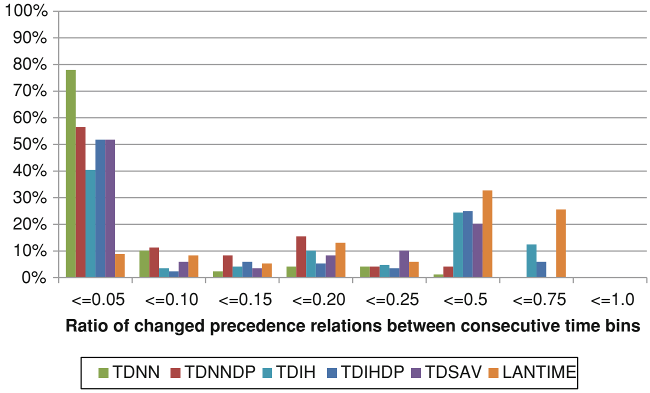


Fig. 9.4 Analysis of precedence relations (Customer Scenario 1)

Performance of algorithms with respect to relative solution quality and run time is presented in Table 9.1. For each heuristic, the ratio of best known solutions found with respect to all 168 solutions is presented. Furthermore, the average deviation from the best known solution is given as well as the average run time, derived from the execution on an AMD Athlon 64 X2 Dual Core Processor PC with 2.4 Ghz and 5 GB RAM. TDNN and TDNNDP perform poorly. TDIH and TDIHDP also never find the best known solution, but are able to decrease the average deviation. TDSAV and modified LANTIME feature the best known solutions. TDSAV may be advantageous due to the depot being situated in the outskirts, which fosters the algorithmic principle of merging pendulum tours in this customer scenario.

In Fig. 9.4, the ratio of alterations in route structures is depicted for each heuristic, distinguished by different colors. TDNN is limited to rather small changes of route structures: for almost 80% of consecutive time bins, precedence relations remain almost the same, i.e., the next best decision does not lead to fundamental changes in route structures. TDNNDP is able to alter the route structure on a higher level than TDNN, which is also reflected by superior solution quality. TDIH and LANTIME are capable of shifting the route structure to a large extent; especially the LANTIME heuristic benefits from the metaheuristic

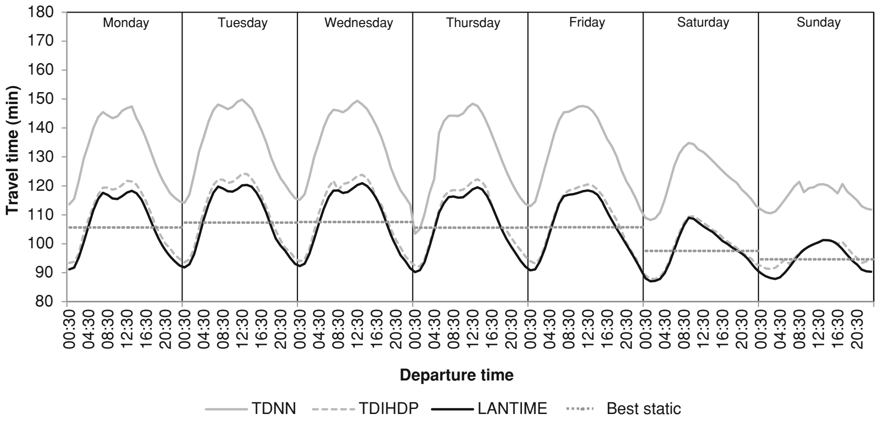


Fig. 9.5 Evolution of overall travel times for Customer Scenario 2

principle and is able to modify up to 75% of all precedence relations. TDSAV either preserves the route structure (in about 50% of all cases) or alters up to 50% of a route structure, respectively.

For inner city tours, TDSAV and LANTIME establish the best results with regard to relative solution quality and computational effort. Evaluation of resulting route structures shows the impact of dynamic input data on tour determination, especially for LANTIME, TDSAV, TDIH, and TDIHDP. Greedy procedures such as TDNN/TDNNDP are not able to react adequately to time-dependent input data in a fine-grained inner city road network.

9.2.2 Customer Scenario 2

Time-dependent routing of 20 outer city customers is conducted in Customer Scenario 2. In Fig. 9.5, results of TDNN, TDIHDP, and LANTIME are presented. As in Customer Scenario 1, time-dependent variations of overall travel times are clearly noticeable. Travel times vary between an overall minimum of 87 min (departure on Saturdays, 2:30) and a maximum of 121 min (departure on Wednesdays, 14:30). The average length of a tour is 88.6 km, which is twice as much as in Customer Scenario 1, although the number of customers is halved. Contrasting Customer Scenario 1, two small peaks are visible at weekdays during morning and afternoon rush hours due to commuter traffic that cannot be bypassed at all.

Considering the performance of individual heuristics (cf. Table 9.2), TDNN is even more distant from the other heuristics' solution quality than in Customer Scenario 1. This might result from rather large distances in Customer Scenario 2, which leads to increasing impact of expensive customers being postponed to the end of a route. Solution quality does not differ a lot for the remaining heuristics. The TDTSP variant of LANTIME is able to find the relatively best solution in all

Table 9.2 Performance of heuristics (Customer Scenario 2)

Algorithm	Ratio of best known solution found (%)	Average deviation from best known solution (%)	Average run time (s)
TDNN	0	23.8	0.0
TDNNDP	3	2.2	110.3
TDIH	18	3.1	0.0
TDIHDP	5	1.9	71.7
TDSAV	33	2.0	0.0
LANTIME	100	0.0	10.0

cases at short run times. TDNNDP features a significant improvement compared to results of Customer Scenario 1, which might be caused by the small size of the problem. TDIH also performs better, finding the best known solution in 18% of all cases.

The analysis of precedence relations reveals the simplicity of route structures within the outer city road network (cf. Fig. 9.6). Almost all heuristics refer to the route structure they have constructed once, i.e., only 5% or less precedence relations are modified. The design of the underlying road network seems to imply corridors that are suited for longer distances, for example, freeways or through roads. If a completely modified tour structure may lead to improvements, it is not found by TDNN or TDNNDP due to their simple evaluation criteria.

Commuter traffic clearly impacts delivery tours in conurbations. Longer distances are covered in a road network of a higher priority than city roads; in rush hours, there is almost no possibility of bypassing congested trunk roads. Thus, variation of chosen routes is very low. Individual heuristics, especially LANTIME and TDIH, are able to produce solutions at low computational efforts. The metaheuristics approach also shows its potential here, but insertion heuristics work quite well, too. This might result from the limited choice of beneficial routes in this scenario.

9.2.3 Customer Scenario 3

This scenario comprises delivery tours for a mixture of 20 inner city and 10 outer city customers, i.e., customers may be located in the entire conurbation. Again, overall travel times of single tours vary a lot. In Fig. 9.7, the results of TDNN, TDSAV, and LANTIME are presented. The weekly minimum arises from departing on Saturdays at 2:30 (91 min), whereas the weekly maximum corresponds to departing on Wednesdays at 11:30 (126 min). The variation of tour durations is alike within weekdays; at weekends, a decrease in travel times as well as in travel time variations occurs. Static planning does not result in a huge variation of tour durations except for the weekend. The static estimation is

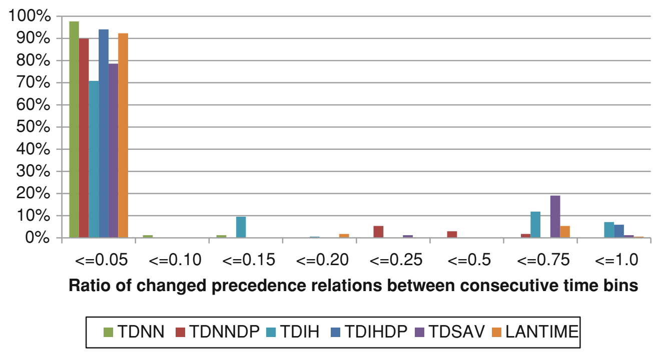


Fig. 9.6 Analysis of precedence relations (Customer Scenario 2)

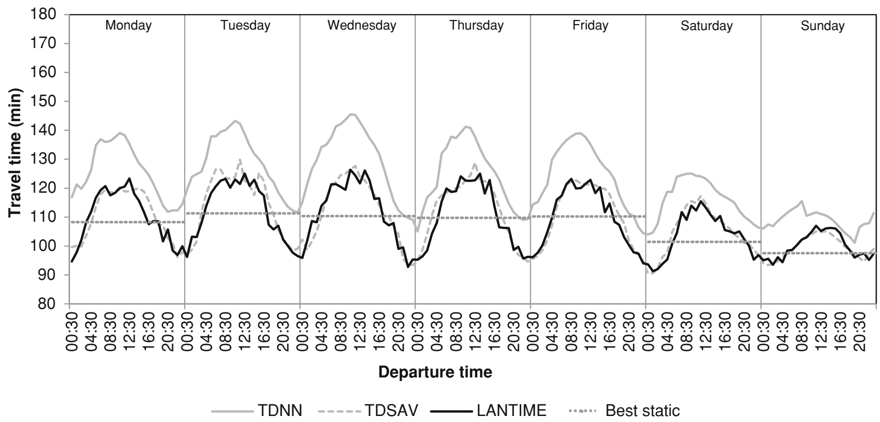


Fig. 9.7 Evolution of overall travel times for Customer Scenario 3

approximately 109 min on weekdays, which underestimates tour durations during the day up to about 14%.

In general, the performance of the heuristics corresponds to observations made in Customer Scenario 2. In Table 9.3, it is reported that TDNN again features relatively poor solutions, whereas LANTIME is able to find the best known solution in 57% of all tours with only a small average deviation from the best known solution. TDSAV follows closely. TDNNDP and TDIHDP again benefit from the relatively small problem size.

Following the lines of Customer Scenario 1, the structural analysis reveals the variety of routes within the whole conurbation, especially in the city road network (cf. Fig. 9.8). LANTIME and insertion heuristics TDIH and TDIHDP are capable of modifying large parts of a route, which is also reflected by relatively superior solution quality. In the case of LANTIME, this leads to the generation of the best known solution found in the majority of all cases. TDSAV also seems to be well suited for the computation of time-dependent routes; it is capable of changing

Table 9.3 Performance of heuristics (Customer Scenario 3)

Algorithm	Ratio of best known solution found (%)	Average deviation from best known solution (%)	Average run time (s)
TDNN	0	14.0	0.0
TDNNDP	1	5.5	224.3
TDIH	1	7.4	0.1
TDIHDP	0	7.4	212.5
TDSAV	40	1.5	0.0
LANTIME	57	0.8	10.0

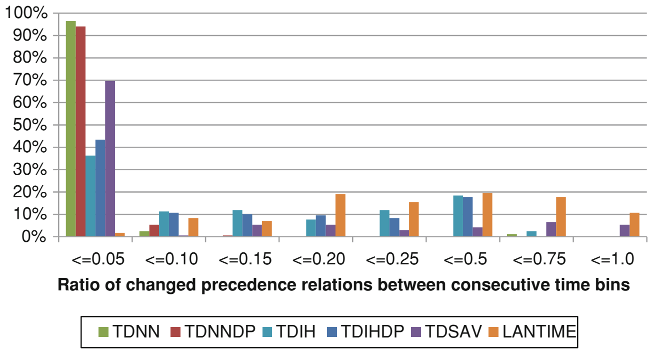


Fig. 9.8 Analysis of precedence relations (Customer Scenario 3)

up to all precedence relations, making it to the second best heuristic of this experiment.

Commuter traffic as well as disturbances of inner city traffic interact in Customer Scenario 3. The stochastic combination of neighborhood operators within the metaheuristic approach ensures that beneficial alternatives can be found, even in the environment of inevitable commuter traffic on access roads. Though, also simple approaches such as TDSAV are able to compute reasonable results at low computational efforts.

The consideration of time dependence in routing of a single vehicle is crucial for the determination of efficient and customer friendly delivery tours in conurbations. Overall travel times vary significantly in the course of the day. As highlighted by the customer scenarios above, time-dependent heuristics react to dynamic input data to a differing extent. From an algorithmic point of view, LANTIME and TDSAV provide superior or the best routes known throughout all customer scenarios, especially with increasing problem size. For smaller problems, TDIHDP is also able to produce competitive solutions, but at high computational efforts. Impact of time-dependent travel times also depends on the application scenario investigated. In inner city areas, the major difference is between "free flow" at night and congested roads throughout the entire weekday, whereas delivery tours on trunk roads mainly suffer from congestion during rush hours which cannot be bypassed at all.

9.3 Routing of a Fleet of Vehicles

This section considers both the efficient assignment of customers to delivery tours and the optimal ordering of customers with respect to time-dependent travel times. Therefore, the TDVRP is exemplarily solved for different sets of customers. Based on the three customer scenarios introduced above, the impact of time-dependent input data is demonstrated. For the computation of the optimal tour plan, it is referred to the LANTIME metaheuristic as introduced in Sect. 8.3.2. First, the performance of the proposed neighborhood operators is investigated. Then, optimal tour plans are generated for individual sets of customers according to customer scenarios.

9.3.1 Performance of Neighborhood Operators

Neighborhood operators are a key component of metaheuristics such as LANTIME. During processing, they are selected according to a probability distribution. Since neighborhoods induced by neighborhood operators may vary in size and complexity, the parameterization of the metaheuristic should consider the characteristics of each operator with respect to computational efforts and potential solution quality. A test environment allows for the investigation of each of the neighborhood operators.

Results for the performance of Adapted Cross Exchange are exemplarily shown in Fig. 9.9. Here, average solution quality of 25 runs is depicted in the course of increasing number of iterations. Box—whisker plots denote the variation of solution quality of overall tour durations resulting from the stochastic behavior of the metaheuristic. The curve denotes average run times in seconds. Compared to remaining neighborhood operators, Adapted Cross Exchange is rather slow; whereas other operators may perform up to 5,000 iterations in 70 s, Adapted Cross Exchange completes only 140 iterations, but shows large improvements of overall tour durations within this small number of iterations. For instance, the upper quartile of tour durations is at 260 min and below as early as 70 iterations. This makes Adapted Cross Exchange a powerful, but relatively slow neighborhood operator.

Although the speed of remaining neighborhood operators is superior, solution quality is worse. Solution quality of the One Exchange operator, for example, has been determined every 100 iterations up to 5,000 iterations (cf. Fig. 9.10). Results are characterized by high variations and only small improvements during the first 2,000 iterations. Then, improvements are only marginally. This surely results from the simplicity of the operator, which may be suitable for the relinking of individual tours, but not for the exchange of partial tours.

In sum, the Adapted Cross Exchange operator leads to significant improvements of overall travel times in a small number of iterations, but processing is relatively slow. To ensure an efficient as well as an effective search, it is proposed

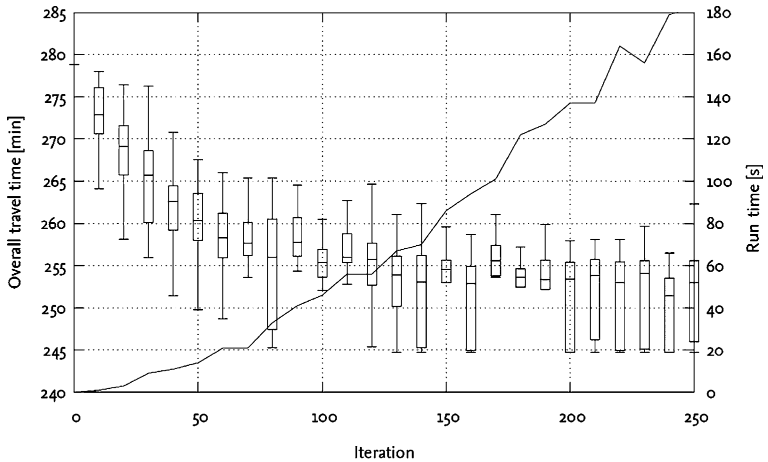


Fig. 9.9 Solution quality and temporal effort of Adapted Cross Exchange

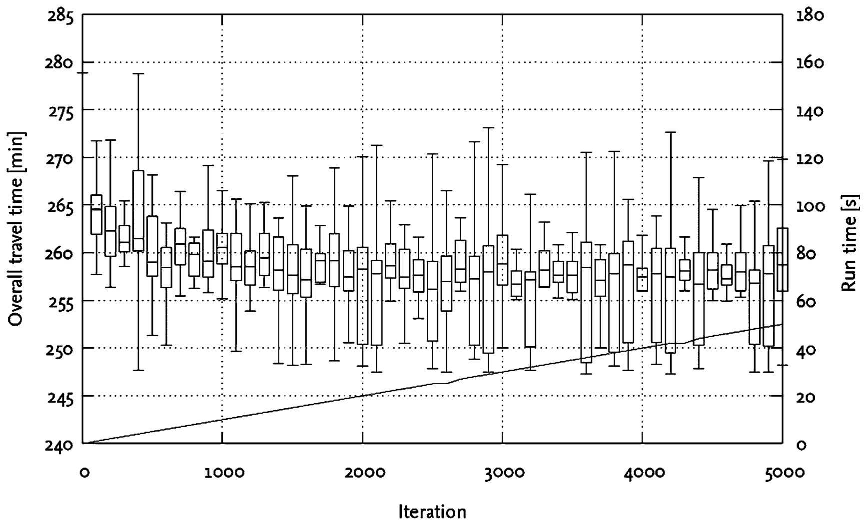


Fig. 9.10 Solution quality and temporal effort of One Exchange

to use the four neighborhood operators with different probabilities according to their speed and potential solution quality. Usage probabilities are estimated from their relative solution quality observed in this test environment. The powerful, but relatively slow Adapted Cross Exchange operator is used with a probability of 5%, whereas One Exchange and Swap Operator are used with a probability of 40%. Insertion/Removal is applied with a probability of 15%. Compared to a scenario with equal probabilities, preliminary experiments have shown a superior performance of the metaheuristic when setting the probabilities as described above.

9.3.2 Computational Results

First, tour plans for 60 inner city customers are computed according to Customer Scenario 1. Then, delivery to 60 outer city customers is investigated according to Customer Scenario 2. Finally, a mixture of 30 inner city customers and 30 outer city customers is analyzed (Customer Scenario 3). For each scenario, the maximum duration of an individual tour is set to 4 h (temporal constraint). LANTIME stops after a maximum computation time of 2 min (time out). The usage probability of individual neighborhood operators is as described above. The analysis of resulting tours occurs in terms of the sum of travel times and the sum of distances of particular tours of a tour plan, as well as the corresponding number of required vehicles. All figures are investigated in terms of 168 departure times at the depot in the course of the week. Customer service time is 10 min. Results presented are derived from the average of 10 runs of the LANTIME metaheuristic.

Results for Customer Scenario 1 are shown in Fig. 9.11. Overall travel times are reported by the dark gray curve. Corresponding distances are denoted as black curve, and the number of required vehicles is shown as light gray curve. As a benchmark, the dotted line denotes overall travel times resulting from FA data. Similar to TDTSP experiments, overall travel times vary in the course of the week, especially on weekdays. As a minimum, the overall travel time of vehicles amounts to 93 min when departing on Saturdays at 2:30. As opposed to this, departure on Wednesdays at 15:30 leads to an overall travel time of 141 min. The relative standard deviation of scheduled travel times is 11%. Especially for departures during the day, overall travel times are underestimated by static planning. The number of required vehicles is varying, too: at night, 3 instead of 4 vehicles are sufficient. Due to the relatively small distances in the inner city area, there is no remarkable variation in distances. Overall kilometers fluctuate in the range of 57.3–63.9 km with a relative standard deviation of only 2%.

The impact of commuter traffic in certain time bins is reflected by results for Customer Scenario 2 (cf. Fig. 9.12). The minimum overall travel time of 210 min occurs when departing on Saturday morning at 2:30, resulting in a requirement of four vehicles. The maximum overall travel time of 303 min corresponds to departure on Wednesdays at 14:30. This is 44% more travel time than in the minimum case or more than 14% compared to the day-specific FA estimation. FA data facilitates weekday specific solutions of the VRP, denoting longer travel times at weekdays and shorter travel times at weekends. FA based planning results in an estimation of four vehicles, whereas time-dependent planning sometimes demands for five vehicles to fulfill the given constraints. Hence, FA based planning may lead to driver overtimes, infeasible schedules, etc. The relative standard deviation of overall travel times is 11%. Kilometers traveled fluctuate in the range of 191–203 km with a relative standard deviation of only 1%.

Results for delivery to customers spreading in the entire conurbation are presented in Fig. 9.13. Four vehicles are required for all departure times. Similar to results above, a typical evolvement of travel times can be stated: relatively short

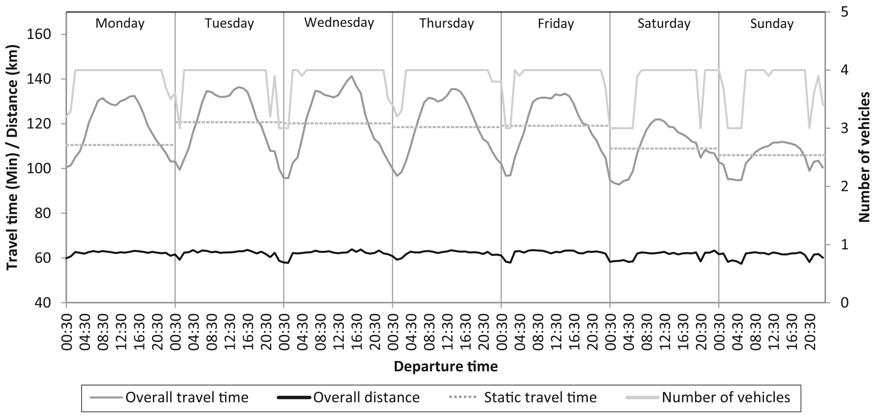


Fig. 9.11 Evolution of overall travel times, distances and number of vehicles for Customer Scenario 1

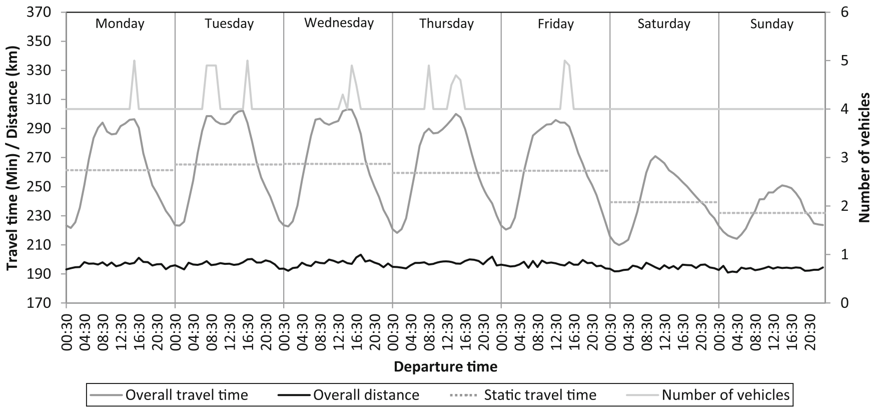


Fig. 9.12 Evolution of overall travel times, distances and number of vehicles for Customer Scenario 2

travel times in the night (minimum: 187 min, departure on Saturdays, 3:30), relatively long travel times during the day, especially in rush hours (maximum: 280 min, departure on Wednesdays, 14:30). Interestingly, afternoon peaks seem to have an even bigger impact on the extension of overall travel times than morning peaks. Time-dependent travel time estimations feature up to 18% longer or 17% shorter travel times compared to the static FA benchmark, depending on the specific departure time. Overall kilometers fluctuate between 150 km at night and 161 km during rush hours, mainly following the evolution of travel times. The relative standard deviation of travel times is 12%.

In Table 9.4, results for TDVRP optimization are consolidated. Here, minima and maxima of overall distances as well as overall travel times are denoted for each customer scenario. The relative standard deviation of overall travel times

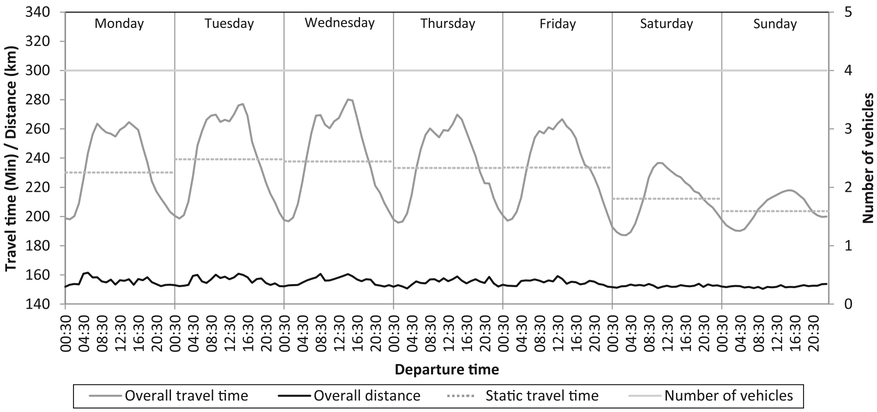


Fig. 9.13 Evolution of overall travel times, distances and number of vehicles for Customer Scenario 3

Table 9.4 Overall results for all customer scenarios

Customer Scenario	Min/Max overall distance	Standard deviation distance (%)	Min/Max overall travel time FW	Standard deviation travel time (%)	Vehicles required	Max deviation from FA
Inner city	57/64	2	93/141	11	3–4	−20%/+20%
Outer city	191/203	1	210/303	11	4–5	−16%/+16%
Entire city	150/161	2	187/280	12	4	−17%/+18%

underlines the impact of time-varying input data. Furthermore, the number of required vehicles is shown which is relatively larger in the outer city scenario due to relatively longer tours. Finally, the range of deviation of the shortest FW based delivery tour to its corresponding FA counterpart is reported. Static planning underestimates or overestimates tour durations on a level of 16–20%, depending on the customer scenario. These results roughly correspond to the evaluation of information models in Chap. 7.

In sum, the consideration of time-dependent information models has a significant impact on routing in city logistics. In TDVRP computation, results from TDTSP computation could be approved. Here, the consideration of spatio-temporal network behavior seems to be even more important, because resulting tour plans do have a major impact on service quality, as well as on delivery costs due to the increasing or decreasing number of required vehicles.

9.4 Customer Time Windows

Finally, the interaction between time-dependent travel times, reliability of customer time windows, and corresponding delivery costs is investigated. Therefore, the role of customer time windows in operational planning is clarified. Comparison of static and time-dependent planning leads to differences in costs and service quality, which is reflected by the number of required vehicles, overall duration of the tour plan, and the number of kept time customer windows.

9.4.1 Role of Customer Time Windows

The aim of operational planning is to produce a cost-efficient tour plan, ideally realized as estimated by the planning system. Among other things, a tour plan is defined by the number of required vehicles, expected travel times between customers, the overall duration, and the overall kilometers traveled. Whereas overall duration and overall kilometers traveled correspond to important economic and ecological measures, the difference between estimated and realized arrival times affects the service quality perceived by the customers. Logistics managers are aware of the fact that real world conditions have a significant impact on the reliability and feasibility of a tour plan. Especially inaccuracies of travel times may cause lateness or earliness of scheduled arrival and service times. Underestimation of travel times may lead to driver overtimes, violations of driving periods, and poor service quality due to violated customer time windows. Overestimated travel times, however, result in inefficient usage of transportation resources, reflected by unnecessary costs.

To produce more reliable tour plans, logistics managers can choose between different alternatives. First, *buffer times* may be added to automatically generated schedules, which increase the probability of timeliness (cf. Sect. 3.4). Though, buffer times also increase the probability of inefficient usage of transportation resources. Second, logistics managers may *constrain the number of customers* per tour, since the effect of lateness propagation may be limited in shorter tours. This leads to higher costs due to a higher number of required vehicles. Third, only *wide customer time windows* may be accepted, reducing the risk of time window violation and alleviating restrictions for operational planning, but also reducing perceived service quality. In the following simulation experiment, these effects are exemplified.

9.4.2 Simulation of Customer Time Windows

Simulation of customer time windows occurs by planning and subsequent simulation of delivery tours. For planning, the following information models are considered:

- Static RT and FA data serve as a benchmark for the most inferior and most superior static information model available.

- Computational experiments have shown that FA based planning may differ up to 18% from the "true," time-dependent travel time. To this end, FA based travel times are increased by 18%, representing a risk averse, static planning approach with buffer times.
- Static planning is contrasted by time-dependent planning based on FW data, which features the smallest RADs between scheduled and realized tour durations and is therefore treated as "true" travel time (cf. Chap. 7).

Tour plans comprise delivery to 100 customers in the environment of Customer Scenario 3 (entire city). To clarify the impact of different traffic states, three *time slots for delivery* are investigated:

- Delivery in the morning of a typical weekday (Thursday 8:00–10:00)
- After work delivery on a typical weekday (Thursday 16:00–18:00)
- Delivery on a Saturday morning (10:00–12:00).

Service quality is investigated in terms of three different options of nonoverlapping *customer time windows* with a length of 20, 10, and 5 min, respectively. For each option, specific time windows are randomly assigned to customers in terms of a uniform distribution in the range of the corresponding slot for delivery. Tour plans are then generated by the I1 heuristic (cf. Sect. 8.3.2) as well as LANTIME with parameters corresponding to the solution of the TDVRP (cf. Sect. 9.3).

After tour plan construction, each tour is simulated by replacing estimated travel times with time dependent, FW based travel times. Both the frequency and the sum of earliness and lateness are reported as: (1) a measure of efficiency and (2) as a measure of service quality. In case of *earliness*, the vehicle waits until the start of service becomes possible. Whereas earliness is mainly caused by the scheduled order of customers in a route, *lateness* arises from delays due to imprecise travel time estimation. If lateness occurs, service is started immediately on arrival.

9.4.3 Computational Results

Detailed results for the simulation of customer time windows are reported in Table 9.5. For each data set, the number of tours, the overall duration of the tour plan and the overall distance are given. All results are based on the I1 heuristic. For time-dependent planning, also results of LANTIME are given, underlining the superiority of the metaheuristic in terms of solutions with a smaller number of vehicles, significantly reduced tour durations and reduced sums of earliness. However, the main focus is on the relative comparison of results produced by the deterministic I1 heuristic, which allows for ignoring variation due to the stochastic nature of LANTIME.

Results of simulation are denoted in terms of the "true" tour plan duration and the corresponding frequency and sum of earliness and lateness. For FW based

Table 9.5 Simulation of customer time windows

	delivery slot		Thursday, 8 – 10			Thursday, 16 – 18			Saturday, 10 – 12		
	length of customer time window		5min	10min	20min	5min	10min	20min	5min	10min	20min
Static planning (RT)	number of tours	[#]	16.0	14.0	13.0	16.0	14.0	13.0	16.0	14.0	13.0
	tour plan duration	[h]	36.3	31.8	29.6	36.3	31.8	29.6	36.3	31.8	29.6
	overall distance	[km]	506.1	454.0	360.5	506.1	454.0	360.5	506.1	454.0	360.5
Simulation	tour plan duration	[h]	37.2	33.5	31.1	37.4	33.9	31.4	36.9	33.0	30.6
	number of earliness	[#]	33.0	10.0	7.0	29.0	9.0	6.0	34.0	12.0	9.0
	sum of earliness	[min]	213.2	65.5	35.6	192.7	61.2	32.3	244.9	76.8	42.1
	number of lateness	[#]	18.0	28.0	13.0	22.0	35.0	16.0	12.0	19.0	8.0
	sum of lateness	[min]	41.8	108.7	30.8	63.2	154.0	52.9	19.7	54.9	13.9
Static planning (FA)	number of tours	[#]	16.0	15.0	13.0	16.0	15.0	13.0	15.0	14.0	13.0
	tour plan duration	[h]	36.8	34.1	30.3	36.8	34.1	30.3	35.5	31.8	30.1
	overall distance	[km]	499.7	445.3	373.0	499.7	445.3	373.0	492.0	451.0	380.9
Simulation	tour plan duration	[h]	37.2	34.7	30.9	37.4	35.0	31.2	35.9	32.5	30.7
	number of earliness	[#]	44.0	18.0	13.0	41.0	16.0	12.0	38.0	20.0	14.0
	sum of earliness	[min]	244.2	133.1	74.1	221.8	124.3	68.0	240.0	81.6	70.5
	number of lateness	[#]	4.0	9.0	2.0	9.0	15.0	5.0	9.0	7.0	5.0
	sum of lateness	[min]	2.5	13.5	3.1	7.7	32.1	9.8	3.5	9.3	6.2
Buffer times (FA+18%)	number of tours	[#]	18.0	16.0	14.0	18.0	16.0	14.0	17.0	15.0	13.0
	tour plan duration	[h]	39.8	37.0	32.6	39.8	37.0	32.6	38.9	34.9	30.7
	overall distance	[km]	464.6	455.2	397.3	464.6	455.2	397.3	504.5	444.9	361.3
Simulation	tour plan duration	[h]	41.7	36.7	32.2	41.8	36.9	32.4	38.7	34.7	30.4
	number of earliness	[#]	53.0	23.0	17.0	52.0	21.0	15.0	49.0	26.0	18.0
	sum of earliness	[min]	429.3	173.6	65.2	408.4	158.0	55.2	337.2	131.5	65.1
	number of lateness	[#]	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
	sum of lateness	[min]	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0
Time-dep planning (FW)	number of tours	[#]	17.0	15.0	13.0	18.0	16.0	14.0	16.0	15.0	13.0
	tour plan duration	[h]	40.0	34.5	30.9	41.8	36.4	32.2	37.7	33.8	30.2
	overall distance	[km]	493.9	445.5	377.2	461.7	448.7	383.3	485.5	445.7	372.2
	number of earliness	[#]	49.0	25.0	18.0	54.0	21.0	21.0	55.0	20.0	15.0
	sum of earliness	[min]	400.2	128.0	63.3	416.5	146.1	76.2	340.9	138.4	51.9
Time-dep planning (FW) LANTIME	number of tours	[#]	16.0	15.0	13.0	17.0	15.0	14.0	15.0	14.0	13.0
	tour plan duration	[h]	35.0	32.2	29.0	36.6	32.6	30.0	34.1	31.3	28.7
	overall distance	[km]	526.8	455.6	373.0	544.5	455.5	373.9	546.3	438.3	379.1
	number of earliness	[#]	24.0	21.0	16.0	33.0	19.0	18.0	24.0	21.0	22.0
	sum of earliness	[min]	74.4	38.1	18.5	90.1	31.2	31.5	52.4	48.3	29.4

planning, no simulation occurs, since FW refers to the best known information model. Columns distinguish different lengths of delivery time slots, comparing the impact of very tight (5 min), tight (10 min) and relaxed (20 min) customer time windows on service quality and costs of delivery.

The estimated number of tours for different delivery time slots and different lengths of customer time windows is depicted in Fig. 9.14 (left). In general, the number of tours decreases with increasing length of customer time windows. For instance, time-dependent planning demands for 18 vehicles for very tight customer time windows in the afternoon rush-hour, contrasted by only 14 vehicles for 20 min customer time windows. This observation is accompanied by decreasing tour plan durations as well as decreasing overall distances. With regard to the particular information model, RT and FA based planning results in the most

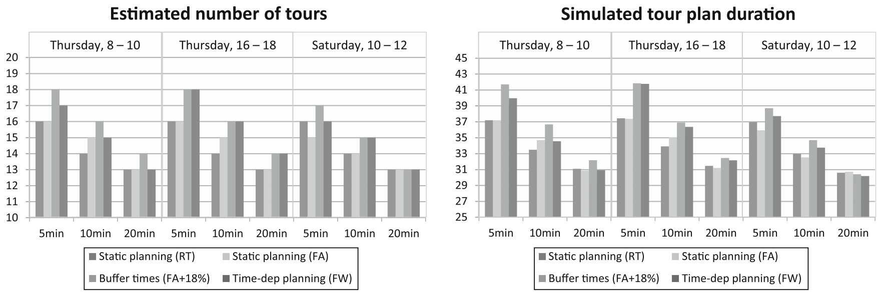


Fig. 9.14 Estimated number of tours and simulated tour plan durations

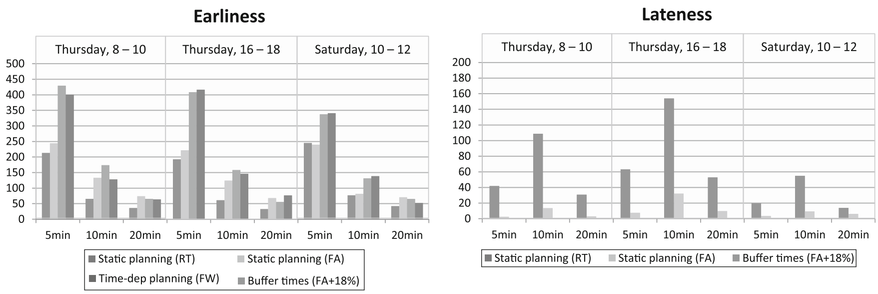


Fig. 9.15 Earliness and lateness resulting from simulation of delivery tours

optimistic estimation of required transportation resources; planning with buffer times as well as planning based on time-dependent travel times often recommends a larger number of tours, especially in the afternoon rush hour.

Simulation of static tour plans results in simulated tour plan durations deviating from estimated tour plan durations, depending on time of day and day of week (Fig. 9.14, right). As a benchmark, also results of time-dependent planning are depicted. In general, simulated tour plan durations are longer in the afternoon rush hour. Once again, increasing degrees of freedom in terms of relaxed customer time windows leads to decreasing simulated tour plan durations, i.e., decreasing costs of delivery. In some cases, buffer times and time-dependent planning result in longer simulated tour durations than RT and FA based planning. However, note that the latter cause violations of customer time windows, which is virtually impossible in case of buffer times and time-dependent travel times.

The impact of different planning data sets on the reliability of customer time windows and efficiency of delivery tours is reflected by the sum of earliness and lateness of simulated delivery tours (cf. Fig. 9.15). Here, earliness decreases with increasing degrees of freedom, i.e., with increasing lengths of customer time windows. RT based planning leads to rather small sums of earliness, contrasted by the largest sums of lateness, though. Earliness rises with the usage of FA data, accompanied by heavily subsiding lateness. However, in the afternoon rush hour, lateness is still a problem. Buffer times often effect the highest sum of earliness as well as the vanishing of lateness. Time-dependent travel times never lead to a delay per assumption, but are able to keep the level of earliness, accompanied by a smaller number of tours. For delivery on Thursday morning, for example, time-dependent solutions get along with one tour less.

In sum, RT and FA based planning produces tour plans that are not feasible and reliable when realized in a time-dependent context, inducing violations of customer time windows. Inclusion of buffer times may alleviate violation of customer time windows, but is accompanied by ineffective utilization of transportation resources, though. Tour plans are more efficient and reliable when derived from time-dependent travel times. Time-dependent planning is able to anticipate the number of required vehicles and corresponding tour durations more precisely, alleviating driver overtimes and customer dissatisfaction, as well as the overestimation of travel times in time slots of less congestion. Here, overall tour plan durations are shorter or at least similar to the measures known from planning with buffer times.

Chapter 10
Conclusions and Outlook

This work investigates the integration of information models and optimization models for routing in city logistics. The following areas have been discussed: recent challenges for city logistics service providers are presented in Part I. Attended home delivery exemplifies a time-critical application of routing in city logistics, underlining the necessity of time-dependent travel times. Technology and methodology for the collection, analysis and preparation of empirical travel times are presented in Part II. Resulting information models establish time-dependent travel time data sets of different volume, complexity, and reliability (Part III). Time-dependent optimization procedures incorporate time-dependent information models. They are investigated and evaluated by simulation of planning procedures for a fictitious logistics service provider (Part IV).

In particular, Chap. 1 introduces the structure of this work and highlights deficiencies of common planning systems. The idea is to improve common planning systems by an integrated approach of city logistics, for example, by consideration of congestion in city logistics routing in terms of time-dependent travel times.

Recent challenges for city logistics service providers are presented in Chap. 2. Increasing complexity of supply chains as well as rising congestion in metropolitan areas exacerbate planning of delivery tours. Planning tasks and functionality of common planning systems are presented, ignoring time-dependent variation of travel times in urban areas. City logistics concepts facilitate solutions considering comprehensive data collection for the provision of time-dependent travel times.

Within Chap. 3, planning of delivery tours for attended home delivery is exemplified. Attended home delivery is a challenging problem for city logistics service providers, because consumers expect tight delivery time windows, which counteract the efficiency of delivery tours. Support of common planning systems is rather limited, since congestion is not considered adequately due to missing data and potentially increasing complexity of computation.

J. F. Ehmke, *Integration of Information and Optimization Models for Routing in City Logistics*, International Series in Operations Research & Management Science 177,
DOI: 10.1007/978-1-4614-3628-7_10,

Enterprise-wide information systems and a large variety of sensors allow for the collection of vast amounts of operational data nowadays. Methodology for the aggregation and analysis of such mass data is introduced in Chap. 4. It is referred to the field of DM, which aims at the analysis of operational data for exploration of required information. KDD serves as a framework for the analysis of mass data. Preprocessing, DM and verification steps are introduced, comprising a multitude of techniques for data analysis. Cluster analysis and EDA are investigated with respect to subsequent analysis of empirical traffic data.

To derive main characteristics of typical traffic network behavior, the KDD is instantiated for the analysis and aggregation of historical traffic data in urban areas (Chap. 5). Historical traffic data are transformed into planning data sets according to different information models. Technology for data collection, analysis of historical traffic data by DM and verification of information models by EDA are presented. Taxi-FCD collects extensive amounts of sensor data for urban areas at low costs. Though, resulting time-dependent planning data sets are too voluminous for the efficient support of planning systems. Thus, second level aggregation facilitates a compact information model, clustering links according to their travel time variation. EDA serves as tool for the temporal and spatial validation of FH and FW information models.

Chapter 6 deals with the preparation of information models according to the requirements of optimization models. The integration of time-dependent information models expects a consistent, time-dependent representation of a digital roadmap. Time aggregated and time-expanded graphs are distinguished, and the FIFO property is introduced, which ensures consistent behavior of the time-dependent road network topology as required by shortest path computation. Shortest path computation produces time-dependent distance matrices as data input for complex routing problems.

In Chap. 7, information models of different complexity are evaluated with respect to reliability of travel time prediction and realization of the shortest itinerary. A comprehensive simulation study reveals that FH and FW information models lead to an increase of planning reliability compared to static information models. In the FW case, data volume is tremendously reduced, keeping the level of a static information model, but allowing for the consideration of time dependence.

Problem formulations, optimization models, and solution procedures for city logistics routing are presented in Chap. 8, reflecting OR methodology. Static and time-dependent variants of TSP and VRP are introduced and then enhanced to the time-dependent case. Besides, heuristics considering customer time windows in city logistics routing are discussed.

The impact of time-dependent information models on optimization procedures is analyzed in Chap. 9. Here, TDTSP and TDVRP are solved by different heuristics for different customer scenarios. A structural analysis of routes and tour plans gives insights into the effectiveness of underlying heuristics and their adaptability to time-dependent input data. Experiments show that metaheuristics and local search algorithms produce reasonable results throughout all customer scenarios. Besides, the impact of time-dependent travel times on customer time windows is exemplified, underlining the necessity of well prepared input data for improved service quality in city logistics routing.

In sum, the investigation of information models and optimization models for routing in city logistics has yielded the following contributions:

- A city logistics framework is presented and applied to understand the role and planning tasks of city logistics service providers. This becomes especially important in the light of increasing importance of e-commerce activities and more complex supply chains.
- The attended home delivery problem is identified as the most challenging delivery problem in the last mile of urban areas. The support of common planning systems is recognized as insufficient.
- The KDD is configured with respect to domain specific requirements of traffic data collection and city logistics routing, ensuring data provision for advanced city logistics planning systems.
- Model-based approaches for the determination of traffic quality are complemented by a data-driven approach from the area of DM.
- Based on a huge amount of empirical traffic data, information models of different volume and complexity are developed. Sophisticated aggregation by DM produces a competitive information model with a data volume comparable to static information models and a planning reliability comparable to extensive time-dependent information models.
- Challenges resulting from the extension of static optimization models to time-dependent optimization models are identified. Well-known static heuristics are presented and enhanced to the time-dependent case.
- Findings from the integration of information models and optimization models are considered in the implementation of a city logistics routing framework, which is able to efficiently solve and analyze routing problems for real road networks.
- Different information models are evaluated by simulation, revealing their strengths and weaknesses regarding reliability of travel time estimation in the environment of urban road traffic.
- Different heuristics for TDTSP and TDVRP solution are evaluated in a real world setting with respect to their ability for adapting to dynamic input data, computational efforts, and solution quality.

Although first attempts of a more realistic modeling of congestion for optimization procedures date back to the 1960s, research on underlying optimization models has been rather scarce. Availability of advanced information and communication technology has recently led to increasing interest in this area. However, the potential of mass data for the improvement of routing applications has not been exploited by far yet. In this work, a methodological framework has been presented which may serve as an origin for the enhancement of planning systems for advanced city logistics applications. Possible kinds of extensions may be as follows:

- The increasing number of sensors may allow for a direct feedback loop between operational planning and real-time operation. Here, dynamic methods from the young field of real-time DM should be considered, allowing for automated data collection, data processing, and dynamic adaptation of information models.
- Interaction of tactical and operational planning may offer enormous potential for cost savings and improvement of customer satisfaction. Time-dependent planning allows for the estimation of costs resulting from a specific design of customer time windows, which can be anticipated in tactical planning.
- Synergies between a compact representation of information on congestion and efficient time-dependent optimization procedures have only been investigated elementarily so far. More sophisticated approaches from the areas of databases, DM, and OR may result in even more sophisticated planning systems. To this end, the presented approach may serve as a framework.

Recent innovations in information and communication technology demand for advances in information modeling and optimization modeling, especially for applications that are characterized by limited resources (e.g., traffic infrastructure and transportation resources) in a dynamic context (e.g., recent applications of routing in city logistics). In the future, increasing economic and ecological requirements induce the integrated treatment of problems as exemplified in this work. Exploiting of the enormous potential of information and communication technology is crucial to ensure sufficient service quality at competitive costs.

References

Aberle, G. (2009). *Transportwirtschaft. Einzelwirtschaftliche und gesamtwirtschaftliche Grundlagen* (*Transport economy. Fundamentals*). München: Oldenbourg.

Agatz, N., Campbell, A. M., Fleischmann, M., & Savelsbergh, M. (2008). Challenges and opportunities in attended home delivery. In B. L. Golden, S. Raghavan & E. A. Wasil (Eds.), *The vehicle routing problem: latest advances and new challenges* (pp. 379–396). Boston: Springer.

Agatz, N., Campbell, A. M., Fleischmann, M., & Savelsbergh, M. (2011). Time slot management in attended home delivery. *Transportation Science*, *45*(3), 435–449.

Ahn, B. H., & Shin, J. Y. (1991). Vehicle-routing with time windows and time-varying congestion. *Journal of the Operational Research Society, 42*(5), 393–400. Retrieved May 06, 2011 from http://www.jstor.org/stable/2583752.

Allen, J., Thorne, G., & Browne, M. (2007). *BESTUFS good practice guide of urban freight transport*. Manual. Retrieved May 06, 2011 from http://www.bestufs.net/download/BESTUFS_II/good_practice/English_BESTUFS_Guide.pdf.

Ankerst, M., Breunig, M. M., Kriegel, H. P., & Sander, J. (1999). OPTICS: Ordering points to identify the clustering structure. In A. Delis, C. Faloutsos & S. Ghandeharizadeh (Eds.), *Proceedings ACM SIGMOD'99 international conference on management of data* (pp. 49–60). Philadelphia, PA: ACM Press

Applegate, D. L., Bixby, R. E., Chvátal, V., & Cook, W. J. (2007). *The traveling salesman problem: A computational study*. Princeton, NJ: Princeton University Press.

Arabie, P., Hubert, L. J., & De Soete, G. (1996). *Clustering and classification*. Singapore: World Scientific.

Augerat, P., Belenguer, J. M., Benavent, E., Corberan, A., & Naddef, D. (1998). Separating capacity constraints in the CVRP using tabu search. *European Journal of Operational Research, 106*(2–3), 546–557.

Banerjee, A., & Davé, R. N. (2004). Validating clusters using the Hopkins statistic. In IEEE Computational Intelligence Society (Ed.), *Proceedings of 2004 IEEE international conference on fuzzy systems* (pp. 149–153), Hungary: Budapest

Barr, R. S., Golden, B. L., Kelly, J. P., Resende, M. G. C., & Stewart, W. R. (1995). Designing and reporting on computational experiments with heuristic methods. *Journal of Heuristics, 1*(1), 9–32.

Beasley, J. E. (1981). Adapting the savings algorithm for varying inter-customer travel times. *Omega, 9*(6), 658–659.

Bellman, R., & Dreyfus, S. E. (1962). *Applied dynamic programming*. Princeton, NJ: Princeton University Press. Retrieved May 06, 2011 from http://www.rand.org/pubs/reports/R352.html.

J. F. Ehmke, *Integration of Information and Optimization Models for Routing in City Logistics*, International Series in Operations Research & Management Science 177,
DOI: 10.1007/978-1-4614-3628-7, © Springer Science+Business Media New York 2012

Bentner, J., Bauer, G., Obermair, G. M., Morgenstern, I., & Schneider, J. (2001). Optimization of the time-dependent traveling salesman problem with Monte Carlo methods. *Physical Review E*, *64*(036701).

Berthold, M. (2006). Datenbank-Mosaik. Data mining oder die Kunst, sich aus Millionen Datensätzen ein Bild zu machen. *(Database mosaic. Data mining or the art to derive information from millions of data sets)*, *c't* 2006, pp. 164–169.

Berthold, M., & Hand, D. J. (2009). *Intelligent data analysis—An introduction*. Berlin: Springer.

Berthold, M. R., Borgelt, C., Höppner, F., & Klawonn, F. (2010). *Guide to intelligent data analysis. How to intelligently make sense of real data*. London, UK: Springer.

Bierwirth, C., Mattfeld, D. C., & Kopfer, H. (1995). On permutation representations for scheduling problems. In *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature* (pp. 310–318). London, UK: Springer.

Bishop, R. (2005). *Intelligent vehicle technology and trends*. Boston: Artech House.

Bradley, P., Fayyad, U. M., & Reina, C. (1998). Scaling clustering algorithms to large databases. In R. Agrawal & P. Stolorz (Eds.), *Proceedings of 1998 international conference on knowledge discovery and data mining (KDD'98)* (pp. 9–15). New York

Bräysy, O., & Gendreau, M. (2005a). Vehicle routing problem with time windows, Part II: Metaheuristics. *Transportation Science, 39*(1), 119–139.

Bräysy, O., & Gendreau, M. (2005b). Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation Science, 39*(1), 104–118.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1983). *CART: Classification and regression trees*. Belmont, CA: Wadsworth.

Breitenberger, S., Grüber, B., & Neuherz, M. (2004). Extended floating car data—Potenziale für die Verkehrsinformation und notwendigen Durchdringungsraten. (Extended floating car data—Potentials for traffic information and penetration rates required). *Straßenverkehrstechnik, 10*, 522–531.

Brockfeld, E., Lorkowski, S., Mieth, P., & Wagner, P. (2007a). Benefits and limits of recent floating car data technology—An evaluation study. In *Proceedings of WCTR Conference. 11th World Conference on Vransport Research.* Berkeley, CA. Retrieved May 06, 2011 from: http://elib.dlr.de/49618/.

Brockfeld, E., Passfeld, B., &Wagner, P. (2007b). Validating travel times calculated on the basis of taxi floating car data with test drives. In *Proceedings of 14th ITS Conference, Beijing, China*. Retrieved May 06, 2011 from http://elib.dlr.de/50208.

Browne, M., Sweet, M., Woodburn, A., & Allen, J. (2005). *Urban freight consolidation centres: final report*. *Project Report,* University of Westminster. Retrieved May 13, 2011 from http://www.oresund.org/ecomobility/content/download/38831/239756/file/RR3%20Urban%20Freight%20Consolidation%20Centre%20Report.pdf.

Busch, F., Glas, F., & Bergmann, E. (2004). Dispositionssysteme als FCD-Quellen für eine verbesserte Verkehrslagerekonstruktion in Städten. (Disposition systems as FCD source for the improvement of traffic state estimation in urban areas). *Straßenverkehrstechnik, 9*, 437–444.

Campbell, A. M., & Savelsbergh, M. (2005). Decision support for consumer direct grocery initiatives. *Transportation Science, 39*(3), 313–327.

Campbell, A. M., & Savelsbergh, M. (2006). Incentive schemes for attended home delivery services. *Transportation Science, 40*(3), 327–341.

Carpenter, G. A., & Grossberg, S. (Eds.). (1991). *Pattern recognition by self-organizing neural networks*. Cambridge. MA: MIT Press.

Chambers, J. M., Cleveland, W. S., Kleiner, B., & Tukey, P. A. (1983). *Graphical methods for data analysis*. Pacific Grove, CA: Duxbury Press.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). *Cross industry standard process for data mining 1.0. Step-by-step data mining guide*. Retrieved May 13, 2011 from www.crisp-dm.org/CRISPWP-0800.pdf.

Chatterjee, A., & Cohen, H. (2004). *Accounting for commercial vehicles in urban transportation models: Task 4. methods, parameters and data sources*. Washington, DC: Federal Highway Administration.

Chaturvedi, A., Green, P., & Carroll, J. (1994). *K-means, k-medians and k-modes: Special cases of partitioning multiway data meeting presentation*. Houston, TX: The Classification Society of North America (CSNA).

Chaturvedi, A., Green, P. E., & Carroll, J. D. (2001). K-modes clustering. *Journal of Classification, 18*, 35–55.

Cheeseman, P., & Stutz, J. (1996). Bayesian classification (AutoClass): Theory and results. In U. M., Fayyad, G. Piatetsky-Shapiro, P. Smyth & R. Uthurusamy (Eds.), *Advances in knowledge discovery and data mining* (pp. 153–180). Menlo Park, CA: AAAI Press.

Chrobok, R., Kaumann, O., Wahle, J., & Schreckenberg, M. (2004). Different methods of traffic forecast based on real data. *European Journal of Operational Research, 155*(3), 558–568.

Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from central depot to a number of delivery points. *Operations Research, 12*(4), 568–581.

Cook, W. (2011). *World TSP*. Retrieved May 03, 2011 from http://www.tsp.gatech.edu/world/index.html.

Cooke, K. L., & Halsey, E. (1966). The shortest route through a network with time-dependent internodal transit times. *Journal of Mathematical Analysis and Applications, 14*(3), 493–498.

Cordeau, J. F., Gendreau, M., Laporte, G., Potvin, J. Y., & Semet, F. (2002). A guide to vehicle routing heuristics. *Journal of the Operational Research Society, 53*(5), 512–522.

Crainic, T. G. (2008). City logistics. In *Proceedings of INFORMS 2008,* pp. 181–212.

Crainic, T. G., Gendreau, M., & Potvin, J. Y. (2009a). Intelligent freight-transportation systems: Assessment and the contribution of operations research. *Transportation Research Part C, 17*(6), 541–557.

Crainic, T. G., Ricciardi, N., & Storchi, G. (2009b). Models for evaluating and planning city logistics systems. *Transportation Science, 43*(4), 432–454.

Daganzo, C. F. (1997). *Fundamentals of transportation and traffic operations*. Wagon Lane, UK: Emerald.

Dantzig, G. B. (1963). *Linear programming and extensions*. Princeton, NJ: Princeton University Press.

Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. *PAMI-1*(2), 224–227.

Day, W. H. E., & Edelsbrunner, H. (1984). Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of Classification, 1*(1), 7–24.

Dean, B. C. (2004a). Algorithms for minimum-cost paths in time-dependent networks with waiting policies. *Networks, 44*(1), 41–46.

Dean, B. C. (2004b). *Shortest paths in FIFO time-dependent networks: Theory and algorithms*. *Technical report,* MIT Department of Computer Science. Retrieved May 13, 2011 from http://www.cs.clemson.edu/~bcdean/paper23.html.

Dell'Amico, M., Iori, M., & Pretolani, D. (2008). Shortest paths in piecewise continuous time-dependent networks. *Operations Research Letters, 36*(6), 688–691.

Delling, D., & Wagner, D. (2009). Time-dependent route planning. In R. K. Ahuja, R. Möhring, & C. D. Zaroliagis (Eds.), *Robust and online large-scale optimization. Models and techniques for transportation systems* (pp. 207–230). Berlin: Springer.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society, 39*(1), 1–38.

Desrochers, M., Lenstra, J. K., Savelsbergh, M. W. P., & Soumis, F. (1988). Vehicle routing with time windows: Optimization and approximation. In B. L. Golden & A. A. Assad (Eds.), *Vehicle routing: Methods and studies* (pp. 65–84). Amsterdam: Elsevier Science

Desrosiers, J., Dumas, Y., Solomon, M. M., & Soumis, F. (1995). Time constrained routing and scheduling. In M. O. Ball, T. L. Magnanti, C. L. Monma & G. L. Nemhauser (Eds.), *Handbooks in operations research and management science. network routing* (pp. 35–139). Amsterdam: Elsevier Science.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik, 1*(1), 269–271.

Domschke, W. (1997). *Rundreisen und Touren*. *(Round trips and tours)*. München: Oldenbourg.

Donati, A. V., Montemanni, R., Casagrande, N., Rizzoli, A. E., & Gambardella, L. M. (2008). Time dependent vehicle routing problem with a multi ant colony system. *European Journal of Operational Research, 185*(3), 1174–1191.

Dreyfus, S. E. (1969). An appraisal of some shortest path algorithms. *Operations Research, 17*(3), 395–412.

Eglese, R., Maden, W., & Slater, A. (2006). A road timetable to aid vehicle routing and scheduling. *Computers & Operations Research, 33*(12), 3508–3519.

Ehmke, J. F., Meisel, S., & Mattfeld, D. (2010). Floating car data based analysis of urban travel times for the provision of traffic quality. In J. Barceló & M. Kuwahara (Eds.), *Traffic data collection and its standardization* (pp. 129–150). New York: Springer

Erman, J., Arlitt, M., & Mahanti, A. (2006). Traffic classification using clustering algorithms. In *Proceedings of the 2006 SIGCOMM workshop on mining network data* (pp. 281–286). Pisa, Italy: ACM

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases. In *Proceedings of 1996 international conference on knowledge discovery and data mining (KDD'96) 1996*(6) (pp. 226–231), Portland, OR

Ester, M., Kriegel, H. P., & Xu, X. (1995). Knowledge discovery in large spatial databases: Focussing techniques for efficient class identification. In *Proceedings of 1995 International Symposium on Large Spatial Databases (SSD'96)* (pp. 67–82), Portland, OR: Springer

Estivill-Castro, V. (2002). Why so many clustering algorithms: a position paper. In *ACM SIGKDD Explorations Newsletter 4,* (pp. 65–75). Retrieved May 13, 2011 from http://doi.acm.org/10.1145/568574.568575.

Everitt, B. S., Landau, S., & Leese, M. (2001). *Cluster analysis*. Chichester, UK: Wiley.

Fastenrath, U. (2008). Erfahrungen und Perspektiven mit FCD. (Experience with and perspectives of FCD). In Bundesministerium für Wirtschaft und Technologie (BMWi) (Eds.), *Verkehrsmanagement und Verkehrstechnologien. Mobile Zukunft mit intelligenten Verkehrssystemen* (pp. 33–34). Halle, Germany.

Fayyad, U. M., Piatetsky-Shapiro, G., & Smith, P. (1996). From data mining to knowledge discovery: An overview. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth & R. Uthurusamy (Eds.), *Advances in knowledge discovery and data mining* (pp. 1–34). Menlo Park, CA: AAAI Press

Figliozzi, M. A. (2007). Analysis of the efficiency of urban commercial vehicle tours: Data collection, methodology, and policy implications. *Transportation Research Part B, 41*(9), 1014–1032.

Figliozzi, M. A. (2009). A route improvement algorithm for the vehicle routing problem with time dependent travel times. In *Proceedings of the 88th Transportation Research Board Annual Meeting*, Washington, DC

Figliozzi, M. A. (2010). An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows. *Transportation Research Part C, 18*(5), 668–679.

Figliozzi, M. A., Kingdon, L., & Wilkitzki, A. (2007). Freight distribution tours in congested urban areas: Characteristics and implications for carriers' operations and data collection efforts. In *Proceedings of the 86th Transportation Research Board Meeting*, Washington, DC

Fisher, D. (1987). Improving inference through conceptual clustering. In *Proceedings of 1987 National Conference on Artificial Intelligence (AAAI'87)* (pp. 461–465). Seattle, WA.

Flatberg, T., Hasle, G., Kloster, O., Nilssen, E. J., & Riise, A. (2005). Dynamic and stochastic aspects in vehicle routing—A literature survey. In *SINTEF Applied Mathematics,* 16.

Fleischmann, B., & Gietz, M. (2002). Transport- und Tourenplanung. (Transportation Planning and Vehicle Routing). In D. Arnold, H. Isermann, A. Kuhn & H. Tempelmeier, (Eds.), *Handbuch Logistik* (pp. 3-45–3-61). Heidelberg: Springer.

Fleischmann, B., Gietz, M., & Gnutzmann, S. (2004). Time-varying travel times in vehicle routing. *Transportation Science, 38*(2), 160–173.

Fox, K. R., Gavish, B., & Graves, S. C. (1979). *The time dependent traveling salesman problem and extensions*. Working paper 7904, Rochester, NY.

Gambardella, L. M., Taillard, E. D., & Agazzi, G. (1999). MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows. In D. Corne, M. Dorigo & F. Glover (Eds.), *New ideas in optimization* (pp. 63–76). London, UK: McGraw-Hill

Garcia, B. L., Potvin, J. Y., & Rousseau, J. M. (1994). A parallel implementation of the tabu search heuristic for vehicle routing problems with time window constraints. *Computers & Operations Research, 21*(9), 1025–1033.

Geisberger, R., & Sanders, P. (2010). Engineering time-dependent many-to-many shortest paths. In T. Erlebach & M. Lübbecke (Eds.), *Proceedings of the 10th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems* (pp. 74–87). Germany: Schloss Dagstuhl—Leibniz-Zentrum für Informatik, Dagstuhl. Retrieved May 13, 2011 from http://drops.dagstuhl.de/opus/volltexte/2010/2751/.

Gendreau, M., Hertz, A., & Laporte, G. (1994). A tabu search heuristic for the vehicle routing problem. *Management Science, 40*(10), 1276–1290.

Gendreau, M., Laporte, G., & Séguin, R. (1996). A tabu search heuristic for the vehicle routing problem with stochastic demands and customers. *Operations Research, 44*(3), 469–477.

Gennari, J., Langley, P., & Fisher, D. (1989). Models of incremental concept formation. *Artificial Intelligence, 40*, 11–61.

George, B., Kim, S., & Shekhar, S. (2007). Spatio-temporal network databases and routing algorithms: A summary of results. In D Papadias, D Zhang & G Kollios (Eds.), *Advances in spatial and temporal databases. Proceedings of 10th international symposium, SSTD 2007* (pp. 460–477). Berlin: Springer.

Gevaers, R., van de Voorde, E., & Vanelslander, T. (2010). Characteristics and typology of last-mile logistics from an innovation perspective in an urban context. In *Proceedings of WCTR 2010*, Lisbon, Portugal

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, 151*(1), 1–11.

Gietz, M. (1994). *Computergestützte Tourenplanung mit zeitkritischen Restriktionen (Computer based vehicle routing with time restrictions)*. Heidelberg: Physica.

Gillett, B. E., & Miller, L. R. (1974). A heuristic algorithm for the vehicle routing dispatch problem. *Operations Research, 22*(2), 340–349.

Goel, A. (2008). *Fleet telematics*. *Real-time management and planning of commercial vehicle operations.* Boston: Springer.

Golden, B. L., Raghavan, S., & Wasil, E. A. (Eds.). (2008). *The Vehicle Routing Problem: Latest Advances and New Challenges*. Boston: Springer.

Golden, B. L., & Stewart, R. W. (1985). Empirical analysis of heuristics. In E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan & D. B. Shmoys (Eds.), *The traveling salesman problem. A guided tour of combinatorial optimization* (pp. 207–249). Chichester, UK: Wiley.

Gössel, F. (2005). Informationsentropische, spektrale und statistische Untersuchungen fahrzeuggenerierter Verkehrsdaten unter besonderer Berücksichtigung der Auswertung und Dimensionierung von FCD-Systemen. (Entropical, Spectral and Statistical Analysis of Vehicle Generated Traffic Data with Special Consideration of the Evaluation and Dimension of FCD-Systems). *Dissertation, Technische Universität Dresden*. Retrieved May 13, 2011 from http://www.qucosa.de/recherche/frontdoor/?tx_slubopus4frontend[id]=1116572388287-5050.

Gouveia, L., & Voß, S. (1995). A classification of formulations for the (time-dependent) traveling salesman problem. *European Journal of Operational Research, 83*(1), 69–82.
Guha, S., Rastogi, R., & Shim, K. (1998). Cure: An efficient clustering algorithm for large databases. In *Proceedings of 1998 International Conference on Management of Data (SIGMOD'98)* (pp. 73–84). Seattle, WA.
Guha, S., Rastogi, R., & Shim, K. (1999). ROCK: A robust clustering algorithm for categorical attributes. In *Proceedings of 1999 International Conference on Data Engineering (ICDE'99)* (pp. 512–521). Sydney, Australia.
Guha, S., Rastogi, R., & Shim, K. (2000). ROCK: A robust clustering algorithm for categorical attributes. *Journal of Information Systems, 25*(5), 345–366.
Gühnemann, A., Schäfer, R., Thiessenhusen, K., & Wagner, P. (2004). *Monitoring traffic and emissions by floating car data*. ITS Working Papers 2004, ITS-WP-04-07. Retrieved May 13, 2011 from http://elib.dlr.de/6675.
Haghani, A., & Jung, S. (2005). A dynamic vehicle routing problem with time-dependent travel times. *Computers & Operations Research, 32*(11), 2959–2986.
Halkidi, M., Batistakis, Y., & Vazirgannis, M. (2001). On clustering validation techniques. *International Journal of Intelligent Information Systems, 17*(2–3), 107–145.
Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *SIGKDD Explorations, 11*(1), 10–18.
Hall, R. (1985). What's so scientific about MS/OR. *Interfaces, 15*(2), 40–45.
Han, J., & Kamber, M. (2006). *Data mining concepts and techniques*. Amsterdam: Elsevier.
Hand, D. J., Mannila, H., & Smyth, P. (2001). *Principles of data mining*. Cambridge, MA: MIT Press.
Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics, 4*(2), 100–107.
Hartigan, J. A. (1975). *Clustering Algorithms*. Hoboken, NJ: Wiley.
Hashimoto, H., Yagiura, M., & Ibaraki, T. (2008). An iterated local search algorithm for the time-dependent vehicle routing problem with time windows. *Discrete optimization, 5*(2), 434–456.
He, L. L., Bai, H. T., Sun, J. G., & Jin, C. Z. (2006). A general incremental hierarchical clustering method. In G. R. Liu, V. B. C. Tan & X. Han (Eds.), *Computational methods* (pp. 1303–1307). The Netherlands: Springer.
Helsgaun, K. (2000). An effective implementation of the Lin-Kernighan traveling salesman heuristic. *European Journal of Operational Research, 126*, 106–130.
Hertz, A., Laporte, G., & Mittaz, M. (2000). A tabu search heuristic for the capacitated arc routing problem. *Operations Research, 48*(1), 129–135.
Hill, A. V., & Benton, W. C. (1992). Modeling intra-city time-dependent travel speeds for vehicle scheduling problems. *Journal of the Operational Research Society, 43*(4), 343–351.
Hill, A. V., Mabert, V. A., & Montgomery, D. W. (1988). A decision support system for the courier vehicle scheduling problem. *Omega, 16*(4), 333–345.
Hinneburg, A., & Keim, D. A. (1998). An efficient approach for clustering in large multimedia databases with noise. In *Proceedings of 1998 International Conference on Knowledge Discovery and Data Mining (KKD'98)* (pp. 58–65).
Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Journal on Data Mining and Knowledge Discovery, 2*(3), 283–304.
Hülsmann, M., & Windt, K. (2007). *Understanding autonomous cooperation and control in logistics. The impact of autonomy on management information, communication and material flow*. Berlin: Springer.
Ichoua, S., Gendreau, M., & Potvin, J. Y. (2003). Vehicle dispatching with time-dependent travel times. *European Journal of Operational Research, 144*(2), 379–396.
International Organization for Standardization. (2009). *Intelligent transport systems—Geographic data files (GDF)—Overall data specification*, 14825:2004.
Jacoby, W. G. (1997). *Statistical graphics for Univariate and bivariate data*. London, UK: Sage Publications.

Jain, A. K., & Dubes, R. C. (1988). *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice-Hall.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys, 31*(3), 264–323.

Janssen, B. J. P., & Oldenburger, A. H. (1991). Product channel logistics and city distribution centers; the case of The Netherlands. In *OECD Seminar on Future Road Transport Systems and Infrastructures in Urban Areas* (pp. 289–302). Chiba, Japan.

Karypis, G., Han, E. H., & Kumar, V. (1999). Chameleon: hierarchical clustering using dynamic modeling. *IEEE Computer, 32*(8), 68–75.

Kaski, S., Oja, E., & Oja, E. (1999). *Kohonen maps*. Amsterdam: Elsevier.

Kaufman, D. E., & Smith, R. L. (1993). Fastest paths in time-dependent networks for intelligent vehicle-highway systems application. *IVHS, 1*, 1–11.

Kaufman, L., & Rousseeuw, P. J. (1990). *Finding groups in data. An introduction to cluster analysis*. New York: Wiley.

Ketchen, D. J., & Shook, C. L. (1996). The application of cluster analysis in strategic management research: an analysis and critique. *Strategic Management Journal, 17*(6), 441–458.

Kim, S., Lewis, M. E., & White, C. C., I. I. I. (2005). Optimal vehicle routing with real-time traffic information. *IEEE Transactions on Intelligent Transportation Systems, 6*(2), 178–188.

Köhler, E., Langtau, K., & Skutella, M. (2002). Time-expanded graphs for flow-dependent transit times. In R. Möhring & R. Raman(Eds.), *Proceedings of 10th annual European symposium on algorithms – ESA 2002* (pp. 599–611). Berlin: Springer.

Köhler, U. (1997). An innovative concept for city-logistics. In *Proceedings of the 4th World Congress on Intelligent Transportation Systems.* Berlin, Germany.

Köhler, U. (2000). How to change urban traffic with city-logistics. In *Proceedings of the 9th IFAC Symposium Control in Transportation Systems 2000* (pp. 229–231), Braunschweig, Germany.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Journal of Biological Cybernetics, 43*(1), 59–69.

Kohonen, T. (1989). *Self-Organization and associative memory*. New York : Springer.

Kohonen, T., Kaski, S., Lagus, K., Salojarvi, J., Honkela, J., Paatero, V., et al. (2000). Self-organization of massive document collection. *IEEE Transactions on Neural Networks, 11*(3), 574–585.

Koskosidis, Y. A., Powell, W. B., & Solomon, M. M. (1992). An optimization-based heuristic for vehicle routing and scheduling with soft time window constraints. *Transportation Science, 26*(2), 69–85.

Kruskal, J. B. (1964). Multidimensional scaling by optimizing goodness of fit to a non metric hypothesis. *Psychometrica, 1*(29), 1–27.

Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research, 59*(3), 345–358.

Lauritzen, S. L. (1995). The EM algorithm for graphical association models with missing data. *Journal of Computational Statistics and Data Analysis, 19*(2), 191–201.

Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., & Shmoys, D. B. (Eds.). (1985). *The traveling salesman problem. A guided tour of combinatorial optimization*. Chichester, UK: Wiley.

Li, F., Golden, B., & Wasil, E. (2005). Solving the time dependent traveling salesman problem. In B. Golden, S. Raghavan & E. Wasil (Eds.), *The next wave in computing, optimization, and decision technologies* (pp. 163–182). New York: Springer

Lin, S., & Kernighan, B. (1973). An effective heuristic algorithm for the traveling salesman problem. *Operations Research, 21*(2), 498–516.

Lloyd, S. P. (1982). Least Squares Quantization in PCM. (original version: Technical Report, Bell Labs, 1957). *IEEE Transaction on Information Theory, 28*, 128–137.

Lorkowski, S., Mieth, P., & Schäfer, R. (2005). New ITS applications for metropolitan areas based on floating car data. In *ECTRI Young Researcher Seminar,* Den Haag, The Netherlands. Retrieved May 09, 2011 from http://elib.dlr.de/21000/.

Lorkowski, S., Mieth, P., Thiessenhusen, K., Chauhan, D., Passfeld, B., & Schäfer, R. (2004). Towards area-wide traffic monitoring-applications derived from probe vehicle data. In *Proceedings of AATT 2004* (pp. 389–394). Bejing, China. Retrieved May 09, 2011 from http://elib.dlr.de/6723/.

Lu, Q., George, B., & Shekhar, S. (2005). Capacity constrained routing algorithms for evacuation planning: A summary of results. In C. Bauzer Medeiros, M. J. Egenhofer & E. Bertino (Eds.), *SSTD 2005* (pp. 22–24). Heidelberg: Springer.

Lunce, S. E., Lunce, L. M., Kawai, Y., & Maniam, B. (2006). Success and failure of pure-play organizations: Webvan versus Peapod, a comparative analysis. *Industrial Management and Data Systems, 106*(9), 1344–1358.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In L. M. Le Cam & J. Neyman (Eds.), *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (pp. 281–287). Berkeley, CA: University of California Press.

Maden, W., Eglese, R., & Black, D. (2010). Vehicle routing and scheduling with time-varying data: A case study. *Journal of the Operational Research Society, 61*, 515–522.

Malandraki, C., & Daskin, M. S. (1992). Time dependent vehicle routing problems: formulations, properties and heuristic algorithms. *Transportation Science, 26*(3), 185–200.

Malandraki, C., & Dial, R. B. (1996). A restricted dynamic programming heuristic algorithm for the time dependent traveling salesman problem. *European Journal of Operational Research, 90*(1), 45–55.

Meisel, S., & Mattfeld, D. (2010). Synergies of operations research and data mining. *European Journal of Operational Research, 206*(1), 1–10.

Messelody, S., Modena, C. M., Zanin, M., de Natale, G. B. F., Granelli, F., Betterle, E. (2009). Intelligent extended floating car data collection. *Expert Systems with Applications, 36*(3), 4213–4227.

Michalski, R. S., Carbonell, J. G., & Mitchell, T. M. (Eds.). (1983). *Machine Learning: An artificial Intelligence approach*. Los Altos, CA: Morgan Kaufmann.

National Research Council. (2000). *Highway capacity manual*. Washington, DC: Transportation Research Board.

NAVTEQ's NAVSTREETS Street Data Reference Manual v2.0. *White Paper* 2006, Chicago. Retrieved April 26, 2011 from http://www.nn4d.com/getDocument.do?docId=1072.

Neveling, S. (2007). *Gebündelte Verkehre in den Altstadtgassen. (Bundled transportation in the historic district)*. LOGISTICS 2007, pp. 28–31

Ng, R., & Han, J. (1994). Efficient and effective clustering method for spatial data mining. In *Proceedings of 1994 International Conference on Very Large Data Bases (VLDB'94)* (pp. 144–155). Santiago, Chile.

OECD. (2003). *Delivering the goods*. *21st century challenges to urban goods transport,* Paris: Author. Retrieved May 13, 2011 from http://www.SourceOECD.org/9789264102828.

Or, I. (1976). *Traveling salesman-type combinatorial problems and their relation to the logistics of regional blood banking*. Dissertation, Northwestern University, Evanston, IL

Orda, A., & Rom, R. (1991). Minimum weight paths in time-dependent networks. *Networks, 21*(3), 295–319.

Paessens, H. (1987). *Tourenplanung mit TourMaster*. *(Vehicle Routing with TourMaster).* München: Oldenbourg.

Pallottino, S., & Scutella, M. (1997). *Shortest path algorithms in transportation models: Classical and innovative aspects*. Technical Report, Pisa, Italy. Retrieved May 09, 2011 from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.36.4398&rep=rep1&type=ps.

Pape, J. (2006). *Mobilität 2020. Perspektiven für den Verkehr von morgen. (Mobility 2020. Perspectives for future traffic)*. Konvent für Technikwissenschaften der Union der deutschen Akademien der Wissenschaften e.V., IRB, Stuttgart, Germany.

Parsons, L., Haque, E., & Liu, H. (2004). Subspace clustering for high dimensional data: A review. *SIGKDD Explorations, 6*(1), 90–105.

Partyka, J., & Hall, R. (2010). *On the road to connectivity*. *Vehicle routing software survey*. Retrieved May 13, 2011 from http://www.lionhrtpub.com/orms/orms-2-10/frsurvey.html.

Pinkofsky, L. (2006). *Typisierung von Ganglinien der Verkehrsstärke und ihre Eignung zur Modellierung der Verkehrsnachfrage*. *(Typisation of curves of traffic density and their suitability for the modeling of traffic demand)*, Aachen: Shaker.

Potvin, J. Y., & Bengio, S. (1996). The vehicle routing problem with time windows, Part II: Genetic search. *INFORMS Journal on Computing, 8*(2), 165–172.

Potvin, J. Y., & Robillard, C. (1995). Clustering for vehicle routing with a competitive neural network. *Neurocomputing, 8*(2), 125–139.

Potvin, J. Y., & Rousseau, J. M. (1993). A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *European Journal of Operational Research, 66*(3), 331–340.

Punakivi, M., & Saranen, J. (2001). Identifying the success factors in e-grocery home delivery. *International Journal of Retail and Distribution Management, 29*(4), 156–163.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*(1), 81–106.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. San Mateo, CA: Morgan Kaufmann.

Rehm, F., Klawonn, F., & Kruse, R. (2006). A novel approach to noise clustering for outlier detection. *Soft Computing, 11*(5), 489–494.

Richter, A. (2005). *Dynamische Tourenplanung*. *(Dynamic Vehicle Routing)*. Dissertation, Dresden, Germany. Retrieved May 18, 2011 from http://deposit.ddb.de/cgi-bin/dokserv?idn=981811361.

Ritter, H., Martinez, T., & Schulten, K. (1992). *Neural Computation and Self-Organizing Maps: An Introduction*. Reading, PA: Addison-Weasley.

Rodrigue, J., Comtois, C., & Slack, B. (2009). *The geography of transport systems*. London, UK: Routledge.

Rosenkrantz, D. J., Stearns, R. E., & Lewis, P. M. (1977). An analysis of several Heuristics of the Traveling Salesman Problem. *SIAM Journal of Computing, 6*(3), 563–581.

Roy, J. (2001). Recent trends in logistics and the need for real-time decision tools in the trucking industry. In R. H. Sprague (Eds.), *Proceedings of the 34th annual hawaii international conference on system sciences.* Los Alamitos, CA: IEEE Computer Society.

Ruesch, M., & Petz, C. (2008). *E-Commerce and urban freight distribution (home shopping)*. *Best Practice Update (2008) Updated Handbook from Year 2001*. Retrieved May 18, 2011 from http://www.bestufs.net/download/BESTUFS_II/key_issuesII/BESTUFS_II_D2_4_I_E_Commerce_2008.pdf.

Rumelhart, D. E., & Zipser, D. (1985). Feature discovery by competitive learning. *Cognitive Science, 9*(1), 75–112.

Ruske, W. (1994). City logistics—Solutions for urban commercial transport by cooperative operations management. In *OECD Seminar on Advanced Road Transport Technologies* (pp. 195–228), Omiya, Japan.

Sanders, P., & Schultes, D. (2007). Engineering fast planning algorithms. In C. Demetrescu (Ed.), *6th Workshop on Experimental Algorithms (WEA 2007)*(pp. 23–36). Berlin: Springer.

Santos-Pereira, C., & Pires, A. (2002). Detection of outliers in multivariate data: A method based on clustering and robust estimators. In *Proceedings of 5th Annual Conference of the Advanced School for Computing and Imaging* (pp. 291–296). Berlin: Physica.

Savelsbergh, M. (1992). The vehicle routing problem with time windows: Minimizing route duration. *Journal on Computing, 4*(2), 146–154.

Schneider, J. (2002). The time-dependent traveling salesman problem. *Physica A: Statistical Mechanics and its Applications, 314*(1–4), 151–155.

Schonfeld, E. (2010). *Forrester forecast: Online retail sales will grow to $250 billion by 2014*. Retrieved May 13, 2011 from http://techcrunch.com/2010/03/08/forrester-forecast-online-retail-sales-will-grow-to-250-billion-by-2014/.

Sheikholeslami, G., Chatterjee, S., & Zhang, A. (1998). WaveCluster: A multi-resolution clustering approach for very large spatial databases. In *Proceedings of 1998 International Conference on Very Large Data Bases (VLDB'98)* (pp. 428–439). New York.

Sohr, A., Wagner, P., & Brockfeld, E. (2009). Floating car data based travel time prediction with lomb periodogram. In *ITS in Daily Life. 16th ITS World Congress and Exhibition on Intelligent Transport Systems and Services,* Stockholm, Sweden.

Solomon, M. (1986). On the worst-case performance of some heuristics for the vehicle routing and scheduling problem with time window constraints. *Networks, 16*(2), 161–174.

Solomon, M. (1987). Algorithms for the vehicle routing and scheduling problem with time window constraints. *Operations Research, 35*, 254–265.

Solomon, M., & Desrosiers, J. (1988). Time window constrained routing and scheduling problems. *Transportation Science, 22*, 1–13.

Statistisches Bundesamt Deutschland. (2010). *Immer mehr Menschen in immer größeren Städten. (More and more people in increasing cities)*. Retrieved May 13, 2011 from http://www.destatis.de/jetspeed/portal/cms/Sites/destatis/Internet/DE/Presse/pm/2010/04/PD10__153__124,templateId=renderPrint.psml.

Stein, B., Eissen, S. M., & Wißbrock, F. (2003). On cluster validity and the information need of users. In M. H. Hamza(Eds.), *Proceedings of the 3rd IASTED international conference on artificial intelligence and applications.* Calgary, AB: ACTA Press

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

Tan, P. N., Steinbach, M., & Kumar, V. (2009). *Introduction to data mining*. Boston: Pearson Addison-Wesley.

Taniguchi, E., & Thompson, R. G. (Eds.). (2006). Recent advances in city logistics. In *Proceedings of the 4th International Conference on City Logistics* (pp. 15–30). Amsterdam: Elsevier.

Taniguchi, E, Thompson, R. G., & Yamada, T. (1999). Modelling city logistics. In E. Taniguchi & R. G. Thompson (Eds.), *City logistics I. Proceedings.* Kyoto, Japan: Institute of City Logistics.

Taniguchi, E., Thompson, R. G., Yamada, T., & van Duin, R. (2008). *City logistics. Network modelling and intelligent transport systems*. Bingley, UK: Emerald.

Teltarif.de. (2011). *o2-Handys liefern Kunden-Standortdaten an Stau-System. (o2 mobile phones provide location data to congestion system)*. Retrieved March 08, 2011 from http://www.teltarif.de/o2-stau-navigation-cebit-technologie/news/41862.html.

Thompson, R. G., & Taniguchi, E. (2001). City logistics and transportation. In A. M. Brewer, K. J. Button & D. A. Hensher (Eds.), *Handbook of logistics and supply-chain management* (pp. 393–405). Elsevier, Amsterdam.

Toledo, T., & Beinhaker, R. (2006). Evaluation of the potential benefits of advanced traveler information systems. *Journal of Intelligent Transportation Systems, 10*(4), 173–183.

Toth, P., & Vigo, D. (2002). *The vehicle routing problem*. Philadelphia, PA: SIAM.

Tufte, E. R. (1983). *The visual display of quantitative information*. Cheshire, CT: Graphics Press.

Tukey, J. W. (1977). *Exploratory Data Analysis*. Reading, PA: Addison-Wesley.

United Nations Population Fund. (2007). *Urbanization: A Majority in Cities*. *Linking Population, Poverty And Development*, United Nations Population Fund. Retrieved February 21, 2011 from http://www.unfpa.org/pds/urbanization.htm.

van Duin, J. H. R. (1997). *'Evaluation and evolution of the city distribution concept' in Urban Transport and the Environment for the 21st Century* (pp. 327–337). Southampton, UK: WIT Press.

van Woensel, T., Kerbache, L., Peremans, H., & Vandaele, N. (2008). Vehicle routing with dynamic travel times: A queueing approach. *European Journal of Operational Research, 186*(3), 990–1007.

Vanajakshi, L. D. (2004). *Estimation and prediction of travel time from loop detector data for intelligent transportation systems applications*. Dissertation, Texas. Retrieved May 13, 2011 from http://hdl.handle.net/1969.1/2667.

Verband Region Stuttgart (Ed.) (2002). Verkehrsplanung in der Region Stuttgart (Traffic planning in the conurbation of Stuttgart). Retrieved May 13, 2011 from http://www.region-stuttgart.org/vrs.

Wang, W., Yang, J., & Muntz, R. (1997). STING: A statistical information grid approach to spatial data mining. In *Proceedings of 1997 International Conference on Very Large Data Bases* (*VLDB'97*) (pp. 186–195), Athens, Greece.

Wilkinson, L. (1999). *The grammar of graphics*. New York: Springer.

Wu, C. H., Ho, J. M., & Lee, D. T. (2004). Travel-time prediction with support vector regression. *IEEE Transactions on Intelligent Transportation Systems, 5*(4), 276–281.

Zäpfel, G., & Wasner, M. (2002). Planning and optimization of hub-and-spoke transportation networks of cooperative third-party logistics providers. *International Journal of Production Economics, 78*(2), 207–220.

Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: An efficient data clustering method for very large databases. In *Proceedings of 1996 ACM-SIGMOD International Conference on Management of Data (SIGMOD'96)*(pp. 103–114). Montreal, Canada.

Index

J. F. Ehmke, *Integration of Information and Optimization Models for Routing in City Logistics*, International Series in Operations Research & Management Science 177,
DOI: 10.1007/978-1-4614-3628-7, © Springer Science+Business Media New York 2012

R

S

T

U

V

W