DISSERTATION | DOCTORAL THESIS

Titel | Title

Metaheuristics for security routing problems on mixed graphs

verfasst von | submitted by

Georg Erwin Adrian Fröhlich BSc MSc

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Doctor of Philosophy (PhD)

Wien | Vienna, 2025

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 794 370 403
Dissertationsgebiet lt. Studienblatt | Field of study as it appears on the student record sheet:	Business Analytics, Logistics and Operations Research
Betreut von | Supervisor:	Univ.-Prof. Mag. Dr. Karl Franz Dörner Privatdoz.

Acknowledgements

First-off, I would like to thank both of my supervisors Karl Doerner and Margaretha Gansterer, who guided me through the process by sharing their knowledge, their ideas, and also were my initial proof-readers for all of my papers. It surely was a pleasure working with them.

Then, I would also like to thank Guenter Kiechle, who was the supervisor of my Master thesis and possibly one of the contributing factors for me getting the chance to do a PhD at the University of Vienna.

My thanks also goes to the whole department, since the working environment surely was a welcoming one.

Furthermore, I want to thank my anonymous reviewers regardless of whether they liked my papers or not. For one, their inputs surely helped improve the publications. But additionally, reviewing a paper can not be taken for granted, and still is a voluntary work in most cases.

A major thanks also goes to all my friends along the way. Even though you might have had nothing to do with my work or PhD, the energy and happiness gained through you, surely helped me.

Especially my partner Franziska, you were my biggest mental support through some though times, you never judged me unnecessary harshly, and you always tried to motivate me.

Finally, I would like to thank my family, which gave me all the chances in life and surely allowed me to study without any concern about what the future will bring. When seeing the cards others have been dealt in life, I can only thank you even more. Without you it wouldn't have been possible (or at least I guess I would not have chosen to do it).

Abstrakt (Deutsch)

Das Thema dieser Doktorarbeit sind Transportprobleme mit Sicherheitsaspekten. Sie setzt sich primär aus drei Publikationen zusammen, welche in folgender Reihenfolge präsentiert werden: (i) Eine Literaturanalyse von Publikationen, die sich mit der Risikominimierung bei Transportproblemen befassen. (ii) Ein neuartiges Mehrziel-Transportproblem, bei dem Kosten minimiert werden sollen und Routeninkonsistenz maximiert werden soll. Für jenes Problem ist Routeninkonsistenz äußerst granulär definiert, sodass sie bereits durch die Wiederholung von kleinsten Straßensegmenten reduziert wird. Das Problem wird mit einer fachüblichen Methode (ALNS) und Mehrziel-Frameworks (Epsilon-Box-Splitting Heuristik, Epsilon-Constraint Heuristik, Multi-Directional Local Search und ein hybrider Ansatz) gelöst. Die Ergebnisse der Mehrziel-Frameworks werden für eine Vielzahl an Mehrziel-Indikatoren verglichen. Ebenso wird getestet, ob die Verwendung eines Multigraphens die Ergebnisse verbessern kann. (iii) Ein neuartiges dynamische Transportproblem, bei dem Straßen, die durch einen Schneesturm immer wieder unsicher werden, durch den Einsatz von Räumfahrzeugen so sicher wie möglich gehalten werden sollen. Jenes wird mit einer fachüblichen Methode (ALNS) und einem rollenden Horizont - sodass auf etwaige Sturmänderungen reagiert werden kann - gelöst. Es werden verschiedene Berechnungsintervalle und Betrachtungszeiträume getestet. Ebenso wird der Effekt von verschiedenen Schneesturmbewegungen und Prognosegenauigkeiten getestet.

Abstract (English)

This doctoral thesis focuses on routing problems with security aspects. It is mainly composed out of three publications, which are presented in following order: (i) A literature review of vehicle routing publications with aspects of risk minimization. (ii) A novel multi-objective routing problem aiming to minimize costs and to maximize route inconsistency. For this problem, route inconsistency is defined very granularly, so that even the slightest repeat of a street segment causes a decrease. The problem is solved with a state-of-the-art method (ALNS) und multi-objective frameworks (epsilon-box-splitting heuristic, epsilon-constraint heuristic, multi-directional local search, and a hybrid approach). The frameworks' results are compared for several multi-objective indicators. Furthermore, the value of using multi-graphs instead of single-graphs is examined. (iii) A novel dynamic routing problem, where streets have to be kept safe and traversable during a snowstorm, while snow steadily but unpredictably accumulates. The problem is solved with a state-of-the-art method (ALNS), which is integrated into a rolling horizon framework. The effects of different updating intervals and different look-aheads are examined. Furthermore, the influence of snowstorm movements and prediction quality on the solution quality are examined.

Preface

This thesis consists of research conducted during my time as a PhD student at the University of Vienna, and is mainly comprised out of vehicle routing problems focussed on safety and security aspects. Each chapter, except for the introduction and conclusion, corresponds to a published paper. However, they are not ordered according to their publication dates, as beginning with a detailed survey and thereafter focusing on singular problems and solution methods seemed like a more natural structure.

Chapter 2 with the title “Safe and secure vehicle routing: a survey on minimization of risk exposure” has been published in *International Transactions in Operational Research* (Fröhlich et al. (2023)). While Chapter 3 and 4 with the titles “Secure and efficient routing on nodes, edges, and arcs of simple-graphs and of multi-graphs” and “A rolling horizon framework for the time-dependent multi-visit dynamic safe street snow plowing problem” have both been published in *Networks* (Fröhlich et al. (2020, 2024)).

Preliminary results of these works have been presented at several conferences: the EURO 2018, European Conference on Operation Reserach; the WARP 3 in 2019 and the WARP 4 in 2022, Workshop on Arc Routing Problems; OR 2021, International Conference on Operations Research.

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

ACO Ant Colony Optimization.

ALNS Adaptive Large Neighborhood Search.

ARP Arc Routing Problem.

ATM Automated Teller Machines.

B&C Branch-and-Cut.

B&P Branch-and-Price.

CIT Cash-In-Transit.

CVRP Capacitated Vehicle Routing Problem.

DisARP Dissimilar Arc Routing Problem.

EBSH ϵ-Box Splitting Heuristic.

EBSH* ϵ-Box Splitting Heuristic combined with MDLS.

ECH ϵ-Constraint Heuristic.

GA Genetic Algorithm.

GRASP Greed Randomized Adaptive Search Procedure.

ILS Iterated Local Search.

kd-VRP k-dissimlar Vehicle Routing Problem.

LNS Large Neighborhood Search.

MCGRP Mixed Capacitated General Routing Problem.

MDLS Multi-Directional Local Search.

MILP Mixed Integer Linear Program.

MIP Mixed Integer Program.

MO-P-MCGRP Multi-Objective Periodic Mixed Capacitated General Routing Problem.

m-PVRP m-peripatetic Vehicle Routing Problem.

m-PTSP m-peripatetic Traveling Salesperson Problem.

NEARP Node, Edge, Arc Routing Problem.

POI Point Of Interest.

PVRP-TW Periodic Vehicle Routing Problem with Time Windows.

RCTVRP Risk-constrained Cash-in-Transit Vehicle Routing Problem.

RWI Real-World Instance.

SA Simulated Annealing.

TDMVD-TOARP Time-Dependent Multi-Visit Dynamic Team Orienteering Arc Routing Problem.

TDVRP Time-Dependent Vehicle Routing Problem.

TOARP Team Orienteering Arc Routing Problem.

TS Tabu Search.

TSP Traveling Salesperson Problem.

UAV Unmanned Aerial Vehicle.

VND Variable Neighborhood Descent.

VNS Variable Neighborhood Search.

VRP Vehicle Routing Problem.

VRPTW Vehicle Routing Problem with Time Windows.

CHAPTER 1

Introduction

This thesis focuses on vehicle routing problems (VRPs) with safety and security aspects, which due to their broadness and real-world-applications alone can be seen as an important topic. It ranges from topics such as hazardous material transportation to patrol routing, and cash-in-transit. My research is partially a wide-ranging literature review of such problems, and partially the exploration of two novel problems using state-of-the-art methods. Within the latter problems, many of the common methods and characteristics of VRPs can be found: neighborhood-based searches including diversification and intensification strategies, mixed integer programming, solution approaches for multi-objective problems, node- and/or arc-based formulations, multi-graphs, time-dependencies, static and dynamic problems, and computationally non-trivial solution functions.

This thesis first aims to build an understanding of existing research on safe and secure vehicle routing problems, while also explaining several basic concepts shared by many of these problems.

Then it tries to answer several questions: How do different multi-objective frameworks compare to one another? Are simple-graphs or multi-graphs better representations, when solving very large problems? How much look-ahead should methods have, when computation time is a seriously limiting factor? Is it better to solve an ongoing dynamic problem in frequent small intervals (e.g., during the next 30 seconds, calculate the solution for 30 seconds thereafter) or in less frequent but larger intervals (e.g., during the next 300 seconds, calculate the solution for 300 seconds thereafter)?

Besides these questions, some of this thesis's complexity and quality lie in the methods, the analytical methodology used, and the decompositions and transformations of the problems: As underlying method for the frameworks, I used a well-known and proven method - adaptive large neighborhood search (ALNS). I adapted it to the specific problems by not only using standard-operators within it, but also custom ones. I further strengthened it with intensification and diversification strategies. The methods' efficacy was proven by comparing it to benchmark instances, and only thereafter were tests on the new problem types with real-world-based instances performed. For transformations, the conversion of a mixed capacitated general routing problem (MCGRP) into a multi-graph VRP can be highlighted, and for decompositions, the splitting logic for the rolling horizon can be highlighted.

The remainder of the thesis is structured as follows:

Chapter 2 begins with a thorough survey on vehicle routing papers that consider risk minimization. Within the chapter the publications are further grouped into the (i) transportation of hazardous materials, (ii) patrol routing, (iii) cash-in-transit, and (iv) dissimilar routing problem. Furthermore, the concept of how to model multi-graphs and why they might be relevant for risk minimization are detailed in this chapter. The chapter was published in *International Transactions in Operational Research* (Fröhlich et al. (2023)) with my co-authors M. Gansterer, and K. Doerner - both being supervisors of mine, but also aiding me with a collection of papers to include and base my further search on, and a first draft for some of the sections.

Chapter 3 is inspired by patrol routing, where patrols are supposed to be difficult to predict, but also cost efficient. Since patrolling may be required for both arcs and vertices, we formulate the problem as a MCGRP (or as some might call it a node edge arc routing problem). To solve it, we then (i) transform it into a capacitated VRP, (ii) generate a multi-graph to consider a diverse portfolio of routes between vertices, (iii) implement a node-based ALNS, and (iv) integrate the ALNS into different multi-objective frameworks - namely an ϵ-constraint heuristic, an ϵ-box-splitting-heuristic, and a multi-directional local search. After validating efficacy of the ALNS, the multi-objective frameworks are compared for real-world-based instances, and the trade-off between cost and inconsistency examined. The paper was published in *Networks* Fröhlich et al. (2020) - again with my co-authors being M. Gansterer and K. Doerner.

Chapter 4 is inspired by a severe snow plowing problem, where the snow storm is ongoing and plowing must take place without the precise knowledge of how much and where snow will fall. We model our problem as a time-dependent multi-visit dynamic team orienteering arc routing problem, with the goal of keeping the streets as safe as possible. To solve it, we use a rolling-horizon-based approach, where problems are solved in rather quick succession (between 30 and 300 seconds). The problem is thereby effectively reduced to a time-dependent team orienteering arc routing problem. We begin with explaining the base problem, its non-trivial conversion, the information handling for the rolling horizon, and how we later manage to very efficiently calculate solution value changes. Next, our solution method is presented - an arc-based ALNS embedded in a rolling-horizon framework. After validating efficacy of the ALNS on benchmark-instances, we perform a computational study on real-world-based instances. We test a wide range of different information-updating intervals and look-aheads for the rolling horizon method, and examine how the solution quality is affected (i) by different snowstorm movements and (ii) by having prefect versus imperfect information. The paper was published in *Networks* Fröhlich et al. (2024) - M. Gansterer and K. Doerner being my co-authors, once more.

For chapters corresponding to published work, the only changes made compared to the published versions are formatting, structural changes (renumbering of sections), updates to cited references, and correcting the rare typo that fell through the peer-review-process.

CHAPTER 2

Safe and secure vehicle routing: a survey on minimization of risk exposure

Published in: *International Transactions in Operational Research*
Safe and secure vehicle routing: a survey on minimization of risk exposure.
Georg E. A. Fröhlich, Margaretha Gansterer, Karl F. Doerner
https://doi.org/10.1111/itor.13130

Abstract Safe and secure vehicle routing problems refer to the transportation of dangerous (e.g., flammable liquids) or valuable goods (e.g., cash), the surveillance of streets (e.g., police patrols) or other areas (e.g., those within a factory or building), and the response to sudden incidents (e.g., robberies or street disruptions). It thus covers a multitude of models and methods with each having its own objective and constraints, such as unpredictability or risk. We review and classify literature in this field and thereby identify a starting point for researchers in this evolving and practically-relevant field. Our study reveals that there are 82 articles that cover aspects related to safe and secure routing, a majority of which were published in the last five years. We classify the articles into five main categories: (i) transportation of hazardous materials, (ii) patrol routing, (iii) cash-in-transit, (iv) dissimilar routing problems, and (v) modeling of multi-graphs. Categories (i)-(iv) elaborate on the problem studied, while (v) provides a general concept based on road network characteristics most commonly found in safe and secure routing problems. Relevant methods and instances, along with their similarities and dissimilarities, have also been discussed in the paper. Furthermore, specific problem characteristics and future research directions are identified.

2.1 Introduction

Vehicle routing is a very well-studied topic with a lot of ongoing research. However, despite its many classes of problems and importance in real life, safety and security problems (e.g., transportation of cash, police patrols, transportation of hazardous material) appear to be under-represented in the literature. An issue possibly faced by the research community is that safety and security problems can take very different forms and might seem unrelated. This results in an even more split community, even though there are knowledge and several overarching concepts between problems to be gained. A reason for this might be the non-existence of a literature survey, as it could show similarities, synergies, and enable new researchers to have an easier entry into this field of research. We want to address this by giving an exhaustive classification for problems related to safe and secure routing with a special focus on problem characteristics, problem classes, and solution techniques. We only focus on studies in which different variants of vehicle routing problems (VRPs) are modeled and solved using heuristics or exact solution techniques. Related fields, especially the shortest paths problem, do not fit within the scope of our survey and are not considered.

As there is no clear definition of *risk* or *danger* within the studied literature, from hereon we refer to *risk* when a mathematical term is used to measure risk, and the term *danger* when the model is impacted by threats but does not explicitly formulate this (e.g., the danger of being robbed is correlated to the amount of money being transported).

Almost all VRP-based problems focusing on safe and secure routing are in the field of cash-in-transit (CIT), patrol routing, or hazardous material transportation. These three topics are not only very distinct in real life but are also vastly different in terms of risk, objectives, constraints and concerns, and solution approaches. For CIT problems, the danger can be seen as the probability of getting robbed, while specific points must be visited to either pick-up or drop-off valuables. Therefore, the routes must be highly unpredictable. Patrolling problems might be similar to CIT problems as they are also focused on visiting specific check points. Once again unpredictable visiting times and routes are essential. However, when patrolling problems focus on covering specific areas over time (e.g., a police patrol being able to arrive within a few minutes after the reporting of an incident) or on preventing incidents by showcasing presence (e.g., military patrols), the danger lies in insufficient coverage of these areas. Hazardous material transportation typically involves visits of specific points. However, in this case, the danger lies in the occurrence of accidents that are generally not caused by malicious outside forces. These problems aim at minimizing expected harms to the population. Predictability of routes or arrivals is not of interest in these settings. Nevertheless, route diversity might be desired to spread the risk among the population.

Additionally, as some common concepts have been identified in several publications on CIT, patrolling, and hazardous material transportation, we want to take a closer look at them even if they are not specifically tailored for the context of safe and secure routing. One of them are dissimilar routing problems, which are some of the strictest models related to dissimilarity. We observe that some extensions have to be made in order to make them fully applicable for safe and secure routing. Similarly, multi-graph modeling can be found in CIT operations and, as such, could also be applied to some types of patrolling problems. In our study, we give an extensive overview on used data sets in the field. Real-world instances (RWI) and artificial instances are both commonly used. Depending on the problem they are for arc-based formulations, simple node-based formulations, or node-based formulations using a multi-graph. Arc-based formulations allow to consider alternative paths between points of interest (e.g., one being more expensive and another one being cheaper but slower). As these formulations are typically computationally more complex than simple node-based formulations, a trade-off between complexity and the advantages of including alternative paths can arise in multi-graphs.

For our literature study, we searched many library databases that contain major journals on operations research, operations management, management science, etc. This was done using the publicly accessible University of Vienna's platform *u:search* and includes 84 different databases for business and economics (University of Vienna (2021)). The search terms used were pairs of the following two groups' members with the *AND*-operator (e.g.,"security routing", "security vehicles", but not "security safety"):

1. "security", "safety", "hazardous material", "risk", "patrols", "guards", "cash-in-transit" and

2. "routing", "vehicles", "logistics", "transportation", "multigraph", "dissimilar", "inconsistent", "peripatetic".

However, we decided to only consider journal publications in order to limit our survey to a reasonable number of articles. For each of the initially identified papers, we screened the reference list and selected the articles that were not found in the first step. Finally, we applied a descendancy approach, i.e., moving from old information to new by screening all articles that cite one of the papers that we had found to be relevant via *Scopus*. We proceeded until we converged to a final set of publications. This final set consists of 82 articles, the first of which was published in 2007, and more than 50% of them appeared within the last six years (2016-2021).

The remainder of our survey has been organized as follows: Related classifications and definitions are provided in Section 2.2. Hazardous material routing is surveyed in Section 2.3. Sections 2.4 and 2.5 are dedicated to routing of patrols and CIT, respectively. In Sections 2.6 and 2.7, we elaborate on general concepts of dissimilar routes and multi-graph modeling. A summary is presented in Section 2.8.

2.2 Classifications and definitions

Security services have a broad field of applications:

- Personal protection
- Transportation of cash, valuables, hazardous materials, or individuals
- Mobile guarding (inspection of different objects in irregular time intervals)
- Plant security services (inspection of objects in irregular time intervals)
- Event security services
- Monitoring (e.g., alarm monitoring, response to elevator calls)
- Custody of private and public buildings or construction sites
- Additional services (e.g., maintenance of security systems, access of control to public buildings, porter services)

Studies in the field of safe and secure routing can be classified according to the movement of guards and persons or objects, which can be either be in motion or fixed. Table 2.1 provides examples of this classification. Note that some tasks can be assigned to more than one category (e.g., for personal protection, the goal might be to transport persons to a specific place or to take care of a them at a specific place). Only the case with fixed guards and fixed persons or objects does not fall under safe and secure routing due to the absence of routing aspects. Table 2.2 classifies hazmat, patrol routing, and CIT according to these criteria.

When considering the surveyed literature, some similarities can be observed for hazmat and patrolling problems. In the former, the guards/special vehicles are in motion, with the loaded material being secure and moving as well. However, as risks for hazmat operations are often measured via the population possibly exposed to an incident, parts of the secured objects are immovable. Similar to hazmat, patrol routing involves moving guards. Existing models consider either specific sites or hotspots that are in fixed positions. For literature on CIT operations, a split can be observed. Guards are moving in a vehicle, and valuables are also moving with that vehicle. However, sometimes the assumption is that routes are too unpredictable and, therefore, no attacks will happen, whereas sometimes arrival times are assumed to be unpredictable and no attacks during (un)loading duties at fixed sites are to be expected.

Table 2.1: Classification of security services.

		Aspect to be secured	
		Fixed	**Moving**
Guards	**Moving**	e.g., plant security services, mobile guarding	e.g., transportation of valuables, hazardous material, or cash
	Fixed	e.g., construction site safety services, custody of a private house	e.g., a guard in a central location monitoring the movement of trucks throughout the city, access control, geo-locators for children

Table 2.2: Classification of problem classes.

	Problem		
	Hazmat (Section 2.3)	**Patrol routing** (Section 2.4)	**CIT** (Section 2.5)
Guards	Moving	Moving	Moving
Aspects to be secured	Material (moving) & population (fixed)	Hotspots, facilities, streets (fixed)	Incident during travel (moving) or during service (fixed)

Regarding modeling aspects, the first classification can be made based on the points that must be serviced. These could include street segments that must be patrolled or specific locations such as bank branches. Therefore, they can either be VRPs, arc routing problems (ARPs), or node, edge, arc routing problems (NEARPs). In some settings, VRPs or NEARPs can be simplified by aggregating several locations on the same arc to just one arc, resulting in ARPs or NEARPs. If, for example, the task is to patrol a point by just passing it, aggregating several nodes that are on the same street segment is reasonable. However, this approach might lead to loss in solution quality, if individual tasks are more time consuming, e.g., service times, or require specific capacities such as for filling money into automated teller machines (ATMs). In this case, this conversion might be inappropriate as vehicles might be able to service a subset of nodes, but not the complete arc.

An important aspect in safe and secure routing applications are the definition of danger or its inverse, i.e., the definition of safety, from which either can be used as a risk measure. In the studied literature, danger or safety are measured as follows:

- Path inconsistency (safety, large value desired): Nodes, edges, or arcs have to be traversed in inconsistent sequences. This can be achieved through the minimization of the number of traversing times for arcs or edges by varying the routes to be taken between points of service or by varying the sequence in which the services take place (e.g., Duchenne et al. (2012); Talarico et al. (2015a)). This measure is commonly applicable in multi-period models, but typically requires either an arc-based formulation or a multi-graph.

- Time inconsistency (safety, large value desired): Periodical services for the same node, edges, or arc are done at different points in time. This makes it harder to predict the time of service (e.g., Hoogeboom and Dullaert (2019); Soriano et al. (2020)), which is a typical requirement in multi-period problems.

- Population exposure (danger, small value desired): It is commonly found in hazmat publications and combines the likelihood of the occurrence of an incident and its severity for the population (e.g., Bula et al. (2017); Pradhananga et al. (2014)), typically via a sumproduct.

- Coverage (safety, large value desired): Patrol routing problems often aim at covering a large area. Whether a point is covered is usually decided by if it can be reached within a certain time span or if it is within a certain distance (e.g., Capar et al. (2015); Sun et al. (2018)).

- Detectability (danger, small value desired): Surveillance operations such as unmanned aerial vehicles (UAVs) typically do not want to risk being detected under radar and shot down. This can be found in military patrol routing (e.g., Cao et al. (2019); Margolis et al. (2022)).

- Response time (safety, small value desired): Another common goal for patrol routing is to be able to react quickly to any incident. Sometimes average response times are used and, other times, the maximum response time (e.g., Sun et al. (2018); Sun and Wang (2018)).

- Expected monetary loss (danger, small value desired): For CIT operations, the severity of an incident can be vastly different depending on whether it took place at the beginning or at the end of a route. When picking up valuables, the possible severity will increase during the trip, while the inverse holds when delivering valuables. Therefore, identical itineraries might have very different risks depending on the nodes serviced and their order. In addition, symmetry will not hold. One possibility is forcing the expected damage of a route to be below a certain threshold (cf. Talarico et al. (2015b)) or minimizing it.

It should be noted that path and time inconsistency, which are most common in safe and secure routing literature, are only relevant in multi-period settings. Hence, models are typically based on a multi-period formulation where nodes have to be visited more than once within a given time horizon. Note that both *multi-period* and *periodic* problems cover multiple planning periods. On the other hand, *periodic* implies that the periods are similar, whereas *multi-period* refers to the general case. Therefore, from here on, we use the term *multi-period model or problem*.

In addition to the classification presented above, problems can be classified based on the number and characteristics of objectives. We might, for instance, differentiate between (i) minimizing the maximum risk among all routes or (ii) ensuring that risk on each route is below a certain threshold. We identify four types of objectives, namely, (i) single objective, (ii) weighted multi-objective, (iii) lexicographic multi-objective, where objectives can be ranked on the basis of priority, and (iv) Pareto-based multi-objective, where no priority ranking can be made, but several (non-dominated) solutions are identified (cf. Fröhlich et al. (2020)). [1]

Note that most of these classifications are not limited to safe and secure routing and are suitable for most routing problems.

2.3 Hazardous material transportation

In this section, we survey literature on hazardous material transportation and identify future research directions in this field.

2.3.1 Literature review

The majority of studies on hazardous material routing solve the adapted versions of VRPs or capacitated VRPs, and typically try to minimize the danger of population exposure. Generally in these models, edges between two nodes are associated not only with travel cost but also with a predetermined danger level. Solution algorithms are used to generate routes that minimize total travel time and risk exposure. In Figure 2.1, we have provided an example of a VRP of hazardous materials, showing the change in the solution if risk in form of population exposure is considered.

Table 2.3 gives an overview of the studies contributing to the field of hazardous material vehicle routing. We classify them according to (i) the underlying planning problem, (ii) risk measurement method, (iii) the solution method, and (iv) additional characteristics. Furthermore, the instances used are listed. However, as not many are shared, more detailed descriptions are presented in Appendix A

Tarantilis and Kiranoudis (2001) build routes not in the real space where only travel distances among geographical points are considered, but in a so-called *risk space*. A risk space is defined by the sum of the products of populations of a geographic object (towns, cities, etc.) and distance-length between these objects and the point in the graph. The lengths of the projected roads in

[1] Non-dominated solutions are those for which no other solution exists that is equally good or better regarding all objectives and better in at least one.

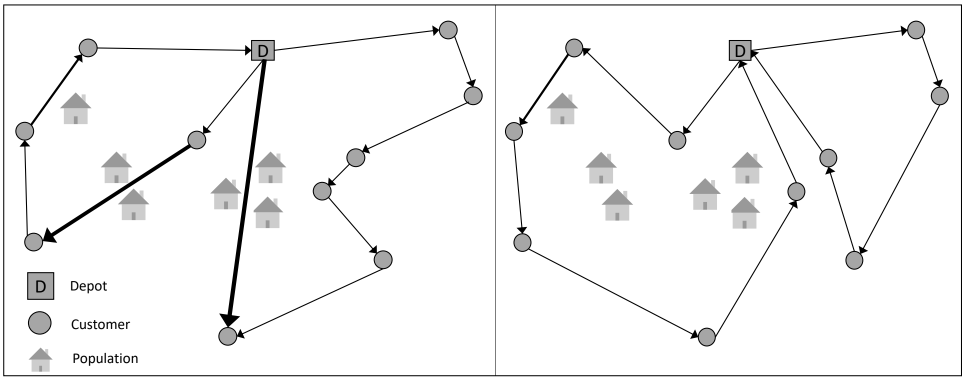


Figure 2.1: Illustration of a hazardous material VRP. On the left part, only travel distance is considered while routes on the right side are such that travel distance and population risk exposure are also taken into account. The thickness of the arcs indicates the level of population risk exposure.

Table 2.3: References and basic characteristics for hazardous material routing (see Appendix A for an explanation of the instances)

Reference	Problem	Risk measure	Method	Additional	Instances
Androutsopoulos and Zografos (2010)	2-SPPTW	Population exposure	Labeling algorithm	Time-dependent	Art. inst.
Androutsopoulos and Zografos (2012)	2-CVRPTW	Population exposure	Insertion algorithm	Time-dependent	Art. inst.
Bula et al. (2017)	HF-CVRP	Population exposure	VNS	Heterogenous fleet	Art. inst.
Ma et al. (2012)	VRPTW	Via link capacity constraints	TS	Link capacity	sol (modified)
Mohri et al. (2020)	Stackelberg	Population exposure	ALNS		RWI (Singapore)
Pradhananga et al. (2010)	2-VRPTW	Population exposure	ACO		sol
Pradhananga et al. (2014)	2-VRPTW	Population exposure	ACO		RWI (Osaka)
Tarantilis and Kiranoudis (2001)	CVRP	Population exposure	Threshold accepting		RWI (Athens)
Taslimi et al. (2020)	PLCVRP	Transportation & storage	Heuristic decomposition	No partial pick-ups	Art. inst., RWI (Dolj)
Zhao and Zhu (2016)	2-MDCVRP	Population exposure	Tchebycheff method		RWI (Nanchuan)
Zografros and Samara (1989)	3-LRP	Via link capacity constraints	Exact solver	Link capacity	Art. inst.
Zografos and Androutsopoulos (2008)	2-CVRPTW	Population exposure	Insertion algorithm	Compartments	RWI (Thriasian Plain)

SPP: Shortest Path Problem, VRP: Vehicle Routing Problem, C: Capacitated, HF: Heterogeneous Fleet, TW: Time Windows, MD: Multi-Depot
VNS: Variable Neighborhood Search, TS: Tabu Search, ACO: Ant Colony Optimization, LRP: Location Routing Problem, 2-: bi-objective, 3-: three objectives
PLCVRP: Periodic Load-dependent CVRP, ALNS: Adaptive Large Neighborhood Search
Art. inst.: Artificial instances, RWI: Real World Instances
sol: see Appendix A

the risk space are curves computed by an integral. They apply a threshold-accepting algorithm to generate the time and load-restricted routes in the risk space graph. Thus, in this work, the risk exposure of the population is explicitly minimized. For testing, they use RWIs based on the city of Athens (Greece).

A bi-objective model is presented by Zografos and Androutsopoulos (2008). While the first objective minimizes transportation cost, the second one deals with risk exposure of the surrounding population. The risk for using an arc is the accident probability multiplied by population size. The accident probability is determined using a discrete choice model, where several parameters, such as geometrical features, traffic characteristics, emergency preparedness, and weather conditions are taken into account. The affected population for an arc is estimated via different scenarios, such as jet fire, vapor cloud explosions, and other likely incidents. Therefore, population close to the arc, even when not directly located on it, might be relevant. It is assumed that customers have to be serviced within specific time windows. Hazardous materials are transported in compartments, and each vehicle has several compartments with a given loading capacity. Solutions for RWIs are generated using an insertion algorithm. A very similar setting is investigated by Zhao and Zhu (2016). Again, the authors propose two objective functions, where one minimizes total travel time and the second one minimizes risk exposure. A parameter indicates the population size per edge. Thus, for each edge, the number of persons affected by a hazardous material released along this edge are given. It is assumed that an edge is homogeneous in terms of the size of the affected population along the arc (e.g., not split into high- and low-density areas within one specific the arc). The authors propose a modified lexicographic weighted Tchebycheff method to generate routes for load constrained vehicles and multiple depots with limited space and test it for artificial and RWIs. Androutsopoulos and Zografos (2010, 2012) study an extension of the problem. Like most other authors, they aim at minimizing the total travel time and risk exposure in a bi-objective manner with time-dependent travel times and risks as their problem extensions. Androutsopoulos and Zografos (2012), however, also include a fleet of vehicles. In both papers, they aim at creating a set of non-dominated solutions. To address the computational complexity, Androutsopoulos and Zografos (2010) use a labeling algorithm that helps generate the k-shortest non-equivalent paths. With it they manage to generate 20 paths in less than one minute, even for most larger instances. In contrast, Androutsopoulos and Zografos (2012) use a weighted-sum approach and perform an insertion algorithm that is able to deliver a reasonable number of good solutions within a short computation time.

Pradhananga et al. (2010, 2014) propose an ant colony optimization (ACO) algorithm for a similar two-objective problem. However, major differences between the two are a limited number of customers serviced per tour in Pradhananga et al. (2010) and the use of real-world data based on the city of Osaka (Japan) in Pradhananga et al. (2014).

Risk minimization for a routing problem with capacitated heterogeneous vehicles is investigated by Bula et al. (2017). The authors propose a single objective wherein the lengths of arcs influence the level of risk. Thus, the lengths of the routes are not explicitly optimized. For each vehicle, an accident rate is given. The authors assume that risk depends on three factors, (i) the probability of an incident, during which the hazardous material is released, (ii) the impact of such incidents, and (iii) the population exposure inside the predefined impact area of the incident. The probability of an incident is determined based on the truck accident rate, the truck load, the conditional release probability of an accident, and the arc length. Solutions are generated using a variable neighborhood search (VNS) algorithm. Artificial instances are used for testing.

A three-objective combined location-routing model is proposed by Zografros and Samara (1989). The objectives are (i) the minimization of risk at disposal sites, (ii) the minimization of routing risk, and (iii) the minimization of travel time. The risk level of disposal sites as well as the risk factors on arcs are assumed to be exogenous. The amount of goods transported on an arc is restricted, and solutions are generated on artificial instances by a commercial solver.

Ma et al. (2012) do not minimize a risk, but they restrict the tonnage transported on an arc. These constraints are motivated by a business project of a Hong Kong hazardous materials transportation company. The authors incorporate these link capacities into a model for the VRP with time windows. A tabu search (TS) algorithm is applied to generate results of artificial instances.

Taslimi et al. (2020) study a novel hazmat problem, where the collection of medical waste is examined. In their case, risk is composed of danger due to transportation, similar to Bula et al. (2017), and danger due to storage. The problem is modeled as a periodic load-dependent capacitated VRP (CVRP), where a multi-day schedule for collecting medical waste without any partial pick-ups has to be created. The authors apply a decomposition-based heuristic by splitting the problem into single-period cases and applying column generation. They test this approach on randomly-generated instances against a mixed-integer programming formulation of their problem solved via the commercial solver CPLEX. For instances that CPLEX could solve within 24 hours, they manage to achieve reasonable percentage gaps (between 0% and 6.06%) with their heuristic, with computation times between sub one second and 20 minutes. On larger instances, they deliver better results than CPLEX. The approach is applied to a case study based on the county of Dolj (Romania).

A hazmat problem has also been modeled as a non-cooperative Stackelberg game by Mohri et al. (2020). The leader is the dispatcher who aims to minimize risk and travel time, and the follower can be seen as the worst-case scenario, which aims to maximize the risk with a given amount of road links where accidents may occur. The authors implement an adaptive large neighborhood search (ALNS), and validate it by comparing it for a case study based on the city of Singapore (Singapore), to the non-dominated solutions obtained by the commercial solver GAMS. With a time limit of one hour, the heuristic performs only marginally worse on the largest instances that GAMS can still solve. They also examine the trade-off between the objectives and the impact of several other factors, such as the Stackelberg game characteristics, safe waiting arcs, more diverse routes, and different vehicle fleet sizes.

2.3.2 Future research directions

Studies on hazardous material vehicle routing are very consistent with respect to the modeling of risk. All risk-based studies listed in Table 2.3 incorporate risk by including a measure for population exposure into the objective function, and can be considered risk-neutral; meaning indifference between two distributions with the same expected values. However, most decision makers are risk-averse (cf. Erkut et al. (2007)) and favor high-probability low-consequence events to low-probability high-consequence events. Therefore, it is desirable to model risk closer to decision makers' preferences. This can be achieved by using perceived risk (cf. Erkut et al. (2007)) or expected disutility (cf. Erkut and Ingolfsson (2000)). Alternatively, a promising approach could be to consider risk via additional constraints, rather than in the objective function. These constraints could limit the highest consequence events within the decision maker's preferences. For this purpose, risk-constrained VRPs, as investigated by Talarico et al. (2015a), Talarico et al. (2017a), and Talarico et al. (2017b) for CIT operations, should be enhanced.

Hazardous material routing is closely related to green VRPs (cf. Eglese and Bektaş (2014); Lin et al. (2014); Bektaş et al. (2019)). Many aspects, such as expected disutility or danger, are relevant in both of these vehicle routing applications. However, these occur specifically on the traversed paths in hazardous material routing, whereas carbon emissions are not just restricted to the routes but affect the whole system. Furthermore, hazardous material routing, contrary to green vehicle routing, is usually not concerned with time-specific travel times, and therefore, little research has been done in this area. This stems from the fact that travel during congestion is seen as too risky due to higher chances of accidents and their severe consequences, should they occur. Several exact solution approaches have been proposed for green VRPs (e.g., Andelmin and Bartolini (2017)). To the best of our knowledge, hazardous material vehicle routing has not been tackled using an exact solution method or a matheuristic (cf. Archetti and Speranza (2014)), which is a hybrid of heuristics and mathematical programming methods. This constitutes a clear research gap. Moreover, multi-period versions of hazardous material routing problems are not explicitly studied in the the literature, as the the only paper discussing the problem (see Taslimi et al. (2020)) proposes a solution approach consisting in the solution of single-period problems. Besides the reduction of danger and high consequence events, a fair distribution of danger within the population might be of interest. This would most likely result in multi-period versions concerning finding different paths or increasing the usage of multi-graphs, in which several predetermined diverse paths are considered.

If hazardous material needs special handling (e.g., cooling, security container with very limited lifespan), stochastic models accounting for possible disruptions - mainly traffic jams and accidents - are recommended. As an incident involving such goods is likely to be of high impact, *unreachability*, as discussed in Nolz et al. (2011), could be a valid measure. This measure expresses the likelihood of a solution becoming impossible, independent of the routing. This could happen if time limitations are applied.

Finally, hazardous material routing problems can be seen in the context of the *Sharing Economy*. Carriers might be willing to cooperate with competitors in order to share transportation dangers. While there are several studies on sharing capacities in carrier collaborations (see Gansterer and Hartl (2018); Pan et al. (2019)), fair and efficient methods to share transportation risks are yet to be discovered.

2.4 Patrol routing

In this section, we deal with the routing of security personnel, e.g., guards and police officers being in charge of surveillance operations or highway patrolling. These problems can be modeled as an ARP if segments of a street network have to be patrolled. An example of a road patrolling problem with hotspots (e.g., streets with high probability of criminal incidents) is depicted in Figure 2.2.

If objects must be inspected, the patrolling problem is modeled based on nodes. If the objects or street segments are visited once or regularly within a given time horizon, *multi-period* routing problems have to be tackled. In Figure 2.3, we depict a node-based patrolling problem in which different kinds of objects must be observed.

Patrol routing has several important aspects. Patrols should be visible in order to have a deterrent effect by increasing the perceived danger/chance of detection (e.g., Willemse and Joubert (2012); Oliveira et al. (2015)) or to increase public confidence in policing (e.g., Chen et al. (2018)). The routes of the patrols should, however, be unpredictable in order to prevent criminals from being able to side-step them (e.g., Willemse and Joubert (2012)). Routes are expected to have minimum coverage, which suggests that roads, paths, or regions have to be patrolled evenly and regularly (e.g., Willemse and Joubert (2012); Keskin et al. (2012); Capar

Figure 2.2: Illustration of a street network with crime hotspots. Patrol routes are expected to traverse as many hotspots as possible; based on Chen et al. (2018).

et al. (2015)). Patrolling duties should be evenly distributed in order to avoid discontentment and work overload among guards (e.g., Willemse and Joubert (2012); Oliveira et al. (2015)). Finally, routes need to be cost-efficient as well (e.g., Sun and Wang (2018)).

2.4.1 Literature review

In Table 2.4, we have listed and classified contributions in the field of patrol routing. We categorized them according to (i) the underlying planning problem, (ii) inconsistency measurement method, (iii) the solution method, and (iv) additional characteristics. Instances used are listed. However, not many are shared, and several real-world and randomly-generated instances are used. Therefore, more detailed descriptions are presented in Appendix A.

Most of the studied problems have either a minimization of costs or a maximization of coverage as their objective. Minimizing a fleet of vehicles could be seen as cost minimization, and reducing total response times is perceived as coverage maximization. Cost minimization is usually the focus when a specific degree of security is desired. However, when a fleet of vehicles or other resources are sunk costs, for instance, due to already fixed wages being paid, coverage can be maximized.

As highway (and freeway) patrolling can be seen as having a few different characteristics than non-highway (non-freeway) patrolling, literature shall be split accordingly. Highway and freeway both tend to have longer street segments with less crossings, easily leading to far more expensive and time intensive alternative routes. One of the earliest non-highway patrolling problems was that of the overnight security service, which was introduced and tackled by Wolfler Calvo and Cordone (2003). The authors present a real-world case based in Italy where guards not only perform their routine inspections but are also in charge of responding to alert signals, which strongly influence the organization of their duty. The authors propose a solution method based

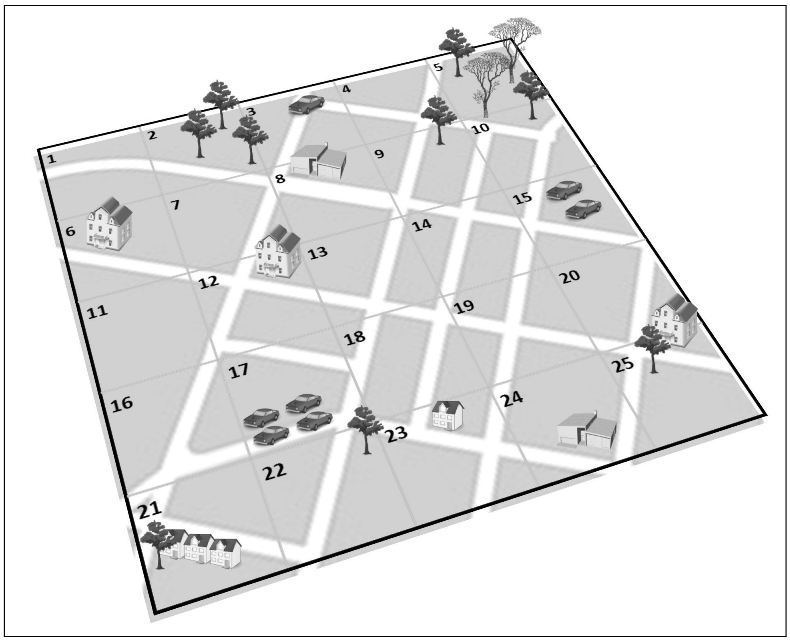


Figure 2.3: Node-based patrolling problem where different kinds of objects must be observed; based on Chen (2012).

on clustering and routing with the aim of minimizing the number of routes. A related study on fleet minimization as its objective is proposed by Prischink et al. (2016). The authors formally introduce the districting and routing problem for security control. While the former splits all objects into a minimum number of disjoint districts, the latter must be solved for each combination of district and day. Inconsistency is considered for the visiting times. It is assumed that successive visits of an object may overlap, but must be separated by a minimum duration. As for the solution approach, a combination of iterative destroy and recreate operations for the districting and a variable neighborhood descent (VND) are proposed and tested using artificial instances. Willemse and Joubert (2012) investigate the routing of security guards in gated communities. The authors formulate the problem as an ARP, more specifically, as a min-max k-rural postperson problem or a min-max k-Chinese postperson problem. The solution, therefore, can be seen as cost minimization. Solutions are generated by a tabu search (TS) that generates sets of routes that are then randomly applied in order to account for unpredictability. RWIs from an estate in South Africa are analyzed. The problem is modeled as multi-vehicle covering tour problem by Oliveira et al. (2015) with the objective of cost minimization. However, additional balance constraints between individual routes are also included. Several heuristics are used to generate solutions. Experiments based on real-world data as well as artificial data are presented. Although inconsistency is mentioned as an important aspect, it is not explicitly included in the model. Similar to Willemse and Joubert (2012), it can be assumed that the heuristics generate sets of solutions that are applied for subsequent periods in a random order. A problem expansion of Willemse and Joubert (2012) can be found in Chen et al. (2018). The authors of the latter study formulate the police patrolling problem as min-max multiple-depot rural postperson prob-

Table 2.4: References and basic characteristics for patrol routing (see Appendix A for an explanation of the instances)

Reference	Problem	Inconsistency	Method	Additional	Instances
Cao et al. (2019)	M-MUCR		GA	Risk via radar	Art. inst.
Capar et al. (2015)	TOPTW		Exact solver		RGI, RWI (Alabama)
Chen (2012)	OP	Time	Cross entropy	Dynamic arrivals	RGI
Chen et al. (2018)	ARP	Paths	TS		RWI (Camden), egl
Hajibabai and Saha (2019)	PPRP		CG&GA		RWI (Sioux Falls)
Hsieh et al. (2015)	BGPRP		EA		Art. inst.
Keskin et al. (2012)	TOPTW		LS&TS		RGI, multiple RWI
Li and Keskin (2014)	FSPP		Heuristics	Multiple locations	RWI (Alabama)
Margolis et al. (2022)	MCTPTW		Exact solver & custom additions	Risk via radar	sol (modified)
Oliveira et al. (2015)	MVCTP		Heuristics		TSPLIB (kro, lin, rd), RWI (Vinhedo)
Prischink et al. (2016)	DRPSC	Time	IDR&VND		TSPLIB (burma, st, rd, tsp, gr, berlin, ft, ch)
Saint-Guillain et al. (2021)	TD-SS-VRP-R		LS		RWI (Brussels)
Sugiura et al. (2015)	VRP		STEN		Art. inst.
Sun et al. (2018)	FSPP		GA	Response time	RWI (Sioux Falls)
Sun and Wang (2018)	FSPP		GA	Response time	RWI (Sioux Falls)
Wallar et al. (2015)	MVCTP		Heuristic decomposition	Risk via radar	Art. inst.
Willemse and Joubert (2012)	ARP	Paths	TS		RWI (Midfield Estate)
Wolfler Calvo and Cordone (2003)	DRPSC	Time	Heuristic decomposition	Alerts	RWI (Milan)
Yassen et al. (2017)	PPRP		Heuristics	Hotspot priorities	sol

VRP: Vehicle Routing Problem, ARP: Arc Routing Problem, MVCTP: Multi-Vehicle Covering Tour Problem
OP: Orienteering Problem, BGPRP: Bi-objective Grid Patrol Routing Problem
TOPTW: Team Orienteering Problem with Time Windows, FSPP: Freeway Service Patrol Problem
DRPSC: Districting and Routing Problem for Security Control
IDR: Iterated Destroy and Repair, VND: Variable Neighborhood Descent
PPRP: Police Patrol Routing Problem, LS: Local Search, TS: Tabu Search, EA: Evolutionary Algorithm,
STEN: Space-Time Extended Network, GA: Genetic Algorithms, CG: Column Generation
M-MUCR: Multi-base multi-UAV cooperative reconnaissance problem
MCTPTW: Multi-vehicle covering tour problem with time windows
TD-SS-VRP-R: Time-Dependent Static and Stochastic VRP with Random requests
Art. inst.: Artificial instances, RGI: Randomly Generated Instances, RWI: Real World Instances
egl, sol, TSPLIB: see Appendix A

lem. Again, a TS is proposed, and real-world cases where crime hotspots in the city of London (England) must be patrolled are presented. The first coverage maximization found is introduced by Chen (2012). Here, patrol route planning problems are investigated in dynamic environments, in particular, heterogeneous dynamic crime arrivals at individual locations. The model is based on an orienteering problem, and solutions are generated using a cross entropy method on artificial instances. Yassen et al. (2017) has dealt with maximum coverage with hotspot prioritization. Their model is based on a VRP with time windows (VRPTW), and solutions for artificial instances are generated using tailored greedy construction heuristics. A multi-objective problem, in the form of a bi-objective grid patrol routing problem, is introduced by Hsieh et al. (2015). Their model includes both the objectives of cost minimization and maximum route coverage. Inconsistency of routes is not explicitly considered. The authors generate solutions using an immune-based evolutionary approach and test them on newly generated instances. A recent study of a time-dependent patrolling problem, where vehicles have to be relocated over the day to optimize the responsiveness for any dynamically appearing requests, is handled by Saint-Guillain et al. (2021). Their objective is to minimize the sum of expected response times for all incidents. They study the problem with and without time-dependent travel times by comparing a simple wait-and-serve method and a local search (LS), where historical data is taken into account on instances based on the city of Brussels (Belgium). Their results indicate that relocating vehicles can yield large improvements.

In the context of highway (and freeway) patrol routing, the early work of Keskin et al. (2012) should be highlighted. The authors work on the state trooper highway patrol problem as a team-orienteering problem. Solutions are generated by local search and TS-based heuristics. Using real-world data from the state of Alabama (USA), the authors give recommendations for (i) critical levels of coverage, (ii) factors influencing the service measures, and (iii) dynamic changes in routes. Improved formulations for this problem and real-world analysis are proposed in Capar et al. (2015). Due to the improved formulations, larger instances could be solved and a better coverage could be provided. Sun et al. (2018) and Sun and Wang (2018) also maximize coverage by minimizing the overall average incident response time. They study a freeway service patrol problem, which involves patrol routing design and fleet allocation. The authors compare and assess geographically overlapping and non-overlapping strategies using genetic algorithms (GA) for real-life instances based on the city of Sioux Falls (USA). Hajibabai and Saha (2019) tackle the patrol routing problem with the goal to minimize the total expected incident response times. A column generation-based solution technique is used to solve the problem for different station designs. To assess the impact of dispatching scenarios on routing costs, the authors integrate GAs with continuous approximation. They also test this on instances of the city of Sioux Falls (USA) used by Sun et al. (2018) and Sun and Wang (2018). However, Sugiura et al. (2015) deviates slightly, as road patrols - at least in Japan - are in charge of not only preventing accidents by driver control but also of removing objects that have fallen on the routes and repairing potholes. They propose to use smart forecasting methods and statistical analyses for road patrolling services, since, for example, pothole occurrences seem to be relatively easy to predict. Therefore, the authors formulate the road patrolling problem as a VRP and solve it for artificial instances using a space-time extended network with the aim to minimize the travel times. A bi-criteria location-routing problem for patrol coverage of highways is studied by Li and Keskin (2014). They aim to minimize costs, including salaries and costs of routing and facilities, and maximize coverage. In addition, it allows state troopers on patrol routes to have multiple starting and stopping spots and considers that hotspots and temporary station change dynamically. They solve it using real-world data from the state of Alabama (USA) via a heuristic and compare the results to the commercial mixed-integer programming solver CPLEX.

Another special case of patrol routing can be found in military applications. Wallar et al. (2015) study a multi-vehicle covering tour problem, where a fleet of UAVs has to scan a specific area. Risk occurs by UAVs being within the range of detection by sensors, which can be mitigated by flying at a greater altitude, which, however, will lower the data quality of the feed. As such, risk and data quality can be seen as a direct trade-off. As a solution approach, first, a two-dimensional routing can be created, ignoring the altitude. After the route is fixed, they adjust the altitude to generate desired trade-off solutions. Cao et al. (2019) also consider a UAV-routing problem, a multi-base multi-UAV cooperative reconnaissance problem to be more precise. Several UAVs, which may be located in different bases, have to scout a set of targets, and to accomplish this, sometimes multiple UAVs must be used within a very specific time frame. Similar to Wallar et al. (2015), they establish the time within detection range of radars as a risk. While they aim to minimize risk, they engage in a conversion into costs. This can be done since their UAVs fly at constant speeds. As a solution approach, they propose a GA. Margolis et al. (2022) consider a multi-objective multi-vehicle-covering tour problem within a specific time window by taking risk minimization and coverage maximization as objectives. Here, risk is defined as in the cases above. They determine efficient sets via the ϵ-constraint method with an underlying exact solver that is strengthened with labeling algorithms and the use of Lagrangian dual variables.

2.4.2 Future research directions

We observe that route or time inconsistency has been hardly considered in relation to patrol routing, with Chen (2012) being the only exception. This is surprising as unpredictability is regularly listed among the most prevalent issues in surveillance operations. Most studies seem to propose generating several single-period problems and applying them in a random order for subsequent periods (e.g., Willemse and Joubert (2012)), while others modify time windows to achieve diversity in solutions (Wolfler Calvo and Cordone (2003)). At this point, researchers tackling patrol routing problems could benefit from contributions in CIT, where both time and path inconsistencies are intensively investigated (see Section 2.5).

In the field of patrol routing, a majority of studies is based on static settings with all necessary information being available. However, it can be assumed for highway and freeway patrolling that incidents as well as sudden changes in traffic conditions (e.g., traffic jams, accidents) occur dynamically, i.e., inconsistently. Non-highway-non-freeway patrolling might face similar events, but to a lesser degree and/or with more alternative routing possibilities. Although a few studies include the dynamic aspect of incidents either via dynamic arrivals, alerts, or maximum response times (Chen (2012); Sun and Wang (2018); Sun et al. (2018); Wolfler Calvo and Cordone (2003)), they do not consider changes in traffic. Adapting optimization problems by including real-time traffic data and creating online algorithms, therefore, seem to be a promising extension that could be used in real-world-applications (Borodin and El-Yaniv (2005)). Online algorithms are characterized by not having all information available and creating a plan during execution with new incoming information (e.g., a driver executes a part of the route, while the next part of the route is determined). Offline algorithms, in contrast, create a plan ahead of time with simple guidelines how to handle interruptions/alarms (e.g., let the closest vehicle handle the alarm, insert it within the current solution at the best position) and do not perform major recalculations (e.g., use an LP-model with the new data and execute a metaheuristic). These characteristics allow online algorithms to be more responsive but the limited computation time granted for updates might result in poor solution quality. Therefore, a valid alternative might be predetermined alternative routes (e.g., a secondary route, if an unexpected traffic jam occurs), which could be used in a stochastic model that is solved offline.

Solution schemes for handling incidents (e.g., reacting to an alarm) and whether these are robust seem to be promising research. A simple way of handling incidents would be to reroute a single vehicle. However, this might cause a previously well-covered area to be less secure and would be exploitable by faking an incident and then striking somewhere else. By rerouting multiple vehicles, this problem could be addressed, but would increase its complexity. A robust solution could handle the majority of incidents with marginal changes to costs or security.

Solution schemes for handling incidents (e.g., reacting to an alarm) and investigation whether these are robust seems to be a promising research question. Robust schemes are able to handle the majority of incidents with marginal changes to costs and security. Simple methods of handling incidents like rerouting just a single vehicle, might cause previously well-covered areas to be less secure. Also, such an approach would be exploitable by faking an incident and then striking somewhere else. By rerouting multiple vehicles, the loss of security could be addressed, but at the costs of an increased complexity which demands more efficient methods.

2.5 Cash-in-transit

In this section, we focus on CIT operations, where money has to be either picked up from or delivered to banks, ATMs, etc.

2.5.1 Literature review

The literature review on CIT is split into three parts: (1) time inconsistency, (2) path inconsistency, and (3) other measures. In Table 2.5, we list and classify relevant contributions to this field. Categorization is according to (i) the underlying problem, (ii) the way inconsistency is measured, (iii) the solution method, and (iv) additional characteristics. Subcategories are made according to the type of inconsistency or other measures used. While instances are also listed, explanation for the more common ones are found in Appendix A.

Time inconsistency

One possible concern of CIT operations is the diversification of service times for periodically visiting customers. Current studies model this as a constraint by restricting customer service times within a certain interval (e.g., Hoogeboom and Dullaert (2019); Michallet et al. (2014); Soriano et al. (2020)). Figure 2.4 depicts an example for time inconsistent routing. The time window for a customer i is given by e_i and l_i for the earliest and the last start of service, respectively. The service times of previous periods are denoted by a_1 to a_3 with the surrounding ϵ_i time units currently not being sufficiently diverse.

Michallet et al. (2014) model such a CIT problem as a periodic VRP with time spread constraints on services (PVRP-TW), in which a set of customers must be serviced in multiple periods at a minimal cost. They assume a limited fleet with limited vehicle capacity. Waiting time is not allowed, and it is assumed that information on required customer services is available a priori. A mathematical model is provided and solved by multi-start iterated local search (ILS). Due to the large amount of constraints, the authors allow for infeasible solutions, which are penalized. The structure of the penalty for the time spread constraints is important. The penalty increases with the closeness to the forbidden point in time (see Figure 2.5). A modification of the problem is presented by Hoogeboom and Dullaert (2019). The authors assume that only the customers of the current day are known. This changes the problem to a sequential solution of VRPs with multiple time windows. They also allow their heuristic - an iterated granular TS - to create solutions violating the constraints. However, when comparing to Michallet et al. (2014), minor changes for the penalty evaluation due to time spreads are imposed. While both study a version in which waiting is not allowed and the departure time is set in a manner that the penalty is minimized, Hoogeboom and Dullaert (2019) also consider a departure as early as possible and a version in which waiting is allowed. In addition, Hoogeboom and Dullaert (2019) provide a computationally more efficient implementation. A related work is done by Soriano et al. (2020). The authors use a problem formulation similar to Michallet et al. (2014) and solve it by applying an ALNS with further intensification of promising solutions through a VND. Additionally, they examine the effect of introducing multiple alternative paths to the problem. As waiting itself is not allowed, it might help to artificially wait by simply taking a longer route, which is similar to a penalty in other models. Their results show that a multi-graph approach can be beneficial as they have improved previous results.

In Yan et al. (2012), a PVRP-TW is studied for instances based on Taiwan's district of Chungli. The authors, however, assume that demands are known only one day in advance and allow for the penalization of waiting times. The problem is modeled as a multiple commodity network flow problem that includes risk with two sets of parameters $\beta_1, ..., \beta_n$ and $\delta_1, ..., \delta_n$. Parameter β_i indicates the required difference in arrival time at the nodes compared to the arrival times i periods earlier, while parameter δ_i is the maximum similarity between the current routes allowed as compared to the routes i periods earlier. The difference to recent solutions is considered more important (i.e., they have a higher weight) than earlier solutions. The authors solve the integer linear programming formulation (ILP) by splitting it into sub-periods. These are sequentially solved from the earliest to the latest, while decisions are fixed regarding earlier service arcs. This model is extended in Yan et al. (2014) by taking stochastic travel times into consideration as well as introducing penalty costs for unanticipated late arrivals. The non-stochastic and stochastic models are tested against each other for the stochastic case by generating multiple scenarios. The authors show that the stochastic model yields superior solutions.

A dissimilar ARP (DisARP) is studied in Constantino et al. (2017) where dissimilarity is also defined on a time-basis - but later handled differently. The DisARP consists of a mixed capacitated ARP, which is extended to multiple days. Each day is split into multiple periods (or time windows), while identical periods of neighboring days may not share the service of any arc. Each period is required to have a minimum number of services. Furthermore, periods as well as services do not have actual times attributed to them. Since the time inconsistency is remodeled as route inconsistency, it is not clear whether it is to be considered time-based or not. To solve this problem, Constantino et al. (2017) use a matheuristic, which identifies tours via a linear model and later composes a solution out of multiple tours. First, they create a set of initial solutions for single days, from which they then choose a set of solutions for the set of days. Thus, the total routing cost is minimized, while similarity constraints are respected. The exact model is compared against the matheuristics on newly generated instances. For larger instances where the exact model reaches a time limit, the matheuristic delivers better results.

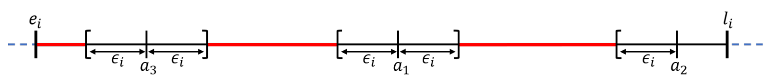


Figure 2.4: Time inconsistency expressed as time spread. a_1, a_2, a_3 are service times in previous periods; the red lines depict the times that would meet the time inconsistency; based on Hoogeboom and Dullaert (2019).

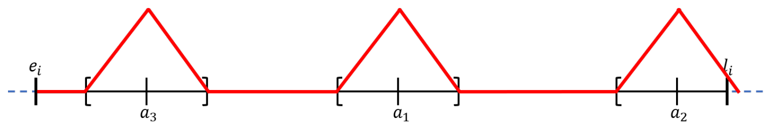


Figure 2.5: Penalty for violating time inconsistency as expressed by time spread. a_1, a_2, a_3 are service times in previous periods; the red line depicts the amount of penalty when servicing at a specific point of time; based on Michallet et al. (2014).

Path inconsistency

Several very different approaches for path inconsistency in CIT can be found in the literature. The strictest of them might be m-peripatetic models, such as the m-peripatetic traveling salesperson problem (m-PTSP) in Duchenne et al. (2005, 2007, 2012), and the m-peripatetic VRP (m-PVRP) in Ngueveu et al. (2010a,b, 2013). These consist of a base problem such as the traveling salesperson problem, which has to be solved for a number of m periods without reusing edges or arcs. All these studies use artificial instances.

In Duchenne et al. (2005), undirected m-PTSPs are mathematically modeled in 2-index and 3-index formulations. Since the 2-index formulation is a relaxation of the 3-index formulation, repair methods help guarantee that feasible solutions are used. Duchenne et al. (2007) expands on this by introducing more inequalities for stronger cuts. Furthermore, the decomposition technique for the 2-index formulation is improved. Both papers show that their 2-index approach delivers better results. The undirected m-PTSP is relaxed to an m-capacitated peripatetic TSP in Duchenne et al. (2012). It allows edges and arcs to be traversed multiple times during the m periods as this might be more applicable to many real-world security problems. The authors use three different mathematical models in conjunction with appropriate solution methods: a 3-index formulation with a branch-and-cut (B&C) algorithm, an edge-edge formulation with a B&C algorithm, and a set-partitioning formulation with a branch-and-price (B&P) algorithm. The latter two models address the problem of symmetry for m-peripatetic problems, meaning that two solutions in the search space that are similar are exchanged. They achieve the best results with their B&P using the set-partitioning formulation.

Ngueveu et al. (2010b) can be seen as the starting point for the follow-up work presented in Ngueveu et al. (2010a) and Ngueveu et al. (2013). Two lower bounds for the m-PVRP are given by solving the relaxations of the m-PVRP. Additionally, upper bounds are heuristically determined via an ILS and a guided/granular TS with diversification. In Ngueveu et al. (2010a), the guided/granular TS with diversification is hybridized with linear programming in the form of a b-matching problem, which is a relaxation of the m-PVRP. This approach delivers good results not only for the m-PVRP but also for two special cases, namely VRP and m-PTSP. In Ngueveu et al. (2013), better lower bounds are created using a column generation-based bounding procedure, in which multiple heuristics are used for generating more efficient cuts. Furthermore, they develop a B&C on edge-based formulations as an exact solution approach.

A less restrictive definition for path inconsistency and security can be found in Bozkaya et al. (2017). The authors study a capacitated VRP with pickups, deliveries, and time windows. The aim is to minimize a weighted objective that consists of cost and risk. Risk is classified into two different types, namely socio-economic risk and usage-based risk. Both types are defined on the paths taken. Socio-economic risk is seen as a fixed risk, depending on the traversed neighborhood. Usage-based risk, on the other hand, is determined by how often a certain arc is used. The authors update usage-based risk similar to an exponential smoothing approach with a weighted sum of the previous usage-based risk and the number of times that an arc was used on the previous day. For solving their problem, they first generate a set of dissimilar and shortest paths between all points that need to be serviced. The days are solved sequentially by first selecting one of the alternative paths between each pair of points for each direction and then applying an ALNS. Alternative paths are selected randomly with probabilities being inverse to the current risk. After having a solution, a time adjustment algorithm is applied in order to diversify arrival times at demand points. During this step, the sequence of the routes cannot change, resulting in the total distance and the risk of the route not changing, thereby ensuring that the total objective value remain the same. However, it should be noted that, even though the risk is path-based, time-based risk is considered within post-processing. Experiments are conducted on RWIs.

The k-dissimilar VRP (kd-VRP), which is based on path dissimilarity, can also be found in the security context. It aims at finding a set of k solutions and solving a VRP that minimizes the costs of the most expensive route, while all solutions must be sufficiently dissimilar. Talarico et al. (2015a) compare all pairs of routes from two solutions and set the similarity to the highest level between the pair of routes. Similarity of routes is determined by the proportion of the costs of shared edges. This measure, as well as the use of artificial instances, can be found in Zajac (2018) who additionally use a spatial measure by taking the proportion of shared grid units for the similarity between routes. This requires the imposition of a grid onto the map of the investigated area. A grid unit is part of a path if the path somehow touches any border of the grid unit. Figure 2.6 depicts such a grid with two paths.

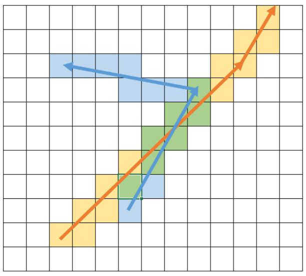


Figure 2.6: Spatial similarity of two routes according to grid units. Yellow and blue grid units are part of just the orange and blue path, respectively. Green grid units are part of both.

Other aspects

An alternative approach for inconsistency based on paths can be found in the risk-constrained CIT VRPs (RCTVRP), e.g., Talarico et al. (2015b, 2017a,b); Radojičić et al. (2018). The RCTVRP is a remodeled CVRP where routes are limited not by a vehicle's capacity but a maximum risk that is allowed per route. Conceptually, the risk of a route where the consequences

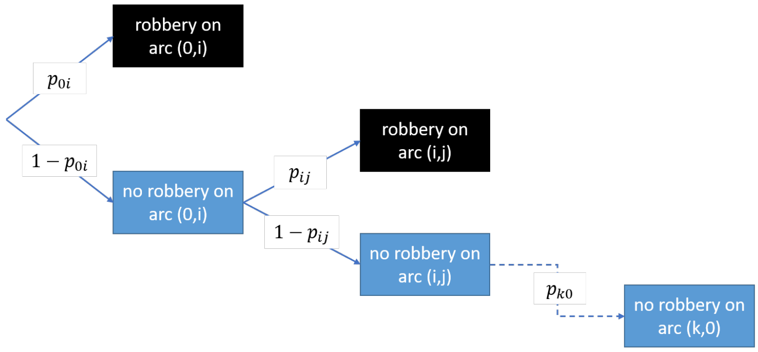


Figure 2.7: Probability tree for being robbed. p_{xy} indicates the probability of being robbed while traversing arc xy; based on Talarico et al. (2015b).

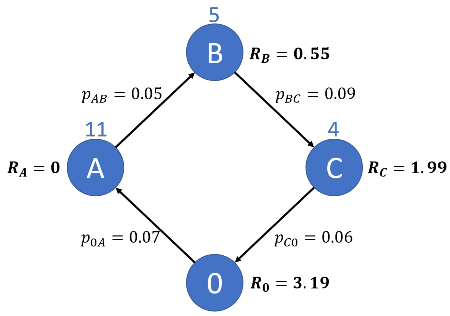


Figure 2.8: Risk in RCTVRP. R_i depicts the cumulative risk until arriving at node i in the case of a CIT pickup problem. p_{ab} depicts the probability of being robbed from node a to b.

of being robbed and the likelihood of being robbed are taken into account. Since a route would end after a robbery, the probability tree can be depicted as shown in Figure 2.7. However, since the probability of being robbed multiple times during a route can be neglected, a non-conditional probability is used. This reduces the probability for an arc b with a set of prior arcs A from $p_b \prod_{a \in A}(1 - p_a)$ to p_b. Figure 2.8 illustrates the risk for a given route from the depot 0 to nodes A, B, and C, and back. The numbers above the nodes and arcs correspond to the money being picked up and the probabilities of a robbery, respectively. The terms R_i indicate the risk at node i.

In Talarico et al. (2015b, 2017b), heuristics for the RCTVRP are developed. In Talarico et al. (2015b), combinations of four construction heuristics with two local search frameworks, one being a greedy randomized adaptive search procedure and the other being an iterative local search, are examined. In Talarico et al. (2017b), the authors develop a combination of ACO and large neighborhood search (LNS). An ACO generates the initial solution via a TSP, where the risk constraint is relaxed, which is then split into a solution respecting the risk constraint. The LNS tries to improve the solution. Additional diversification is generated via perturbation, starting from a new initial solution. Thus, the results presented in Talarico et al. (2015b) are improved. Further improvements are presented in Radojičić et al. (2018). The authors use a fuzzy GRASP in combination with path relinking and present their approach with and without *fuzzification.* Their fuzzy algorithm yields better results regarding average solutions, average iterations, average computation time, and a number of best-found solutions. A combination of time-dependent travel times and their uncertainty is studied by Tikani et al. (2021), with risk being defined similarly to Talarico et al. (2015b). They assume that arcs might get congested, increasing travel time and, consequently, danger. Different arcs might have different probabilities of congestion, which does not always result in the same path from one node to another being the optimal choice. Therefore, a multi-graph is applied. They test an exact two-phase dynamic programming-based algorithm, a heuristic dynamic programming method, and a hybrid method based on GAs and dynamic programming. Using a case study based on the city of Isfahan (Iran), they compare their methods on a deterministic and a stochastic case. They show that their heuristic yields promising results as it manages to solve many of the instances which could be solved by the exact version to optimality. An improvement by hybridizing the GA is also shown. Based on their case study, they also deliver the value of perfect information, which they deem as important but difficult to obtain.

Talarico et al. (2017a) study a multi-objective RCTVRP using RWIs. For the minimization of the maximum risk, the second objective, a local search similar to Talarico et al. (2015b) is applied. The authors propose a weighted sum of the objectives. The weights are set during the search according to the decision makers' preferences and the current set of solutions. A weighted multi-objective secure pickup and delivery problem with time windows is studied by Allahyari et al. (2021). They try to minimize the sum of total costs and risk stemming from the routes with risk being composed from arcs that have been used multiple times. A similar methodology is true for arrival times and total travel time. Similarly to Talarico et al. (2015b), the valuables within the vehicles are also taken into account. After providing a mathematical model and noting its complexity and impracticability for larger instances, they approach it via an ALNS. They show that their ALNS outperforms their CPLEX-formulation under a time limit of three hours and that a further intensification with improvement neighborhoods and a departure time optimization yield great benefits. Furthermore, they validate their method by comparing with other heuristics for the Solomon instances (Solomon (1987)) and also supply a case study based on the city of Tehran (Iran).

2.5.2 Future research directions

A lot of different models and assumptions have already been proposed for CIT problems. However, it seems that in most cases, either a type of time inconsistency or path inconsistency is examined with both advantages and disadvantages. Investigating whether there are strong positive correlations between some diversity criteria (e.g., optimizing criterion 1 results in a good value for criterion 2), which might not be symmetrical, might be of great interest. It could allow future models to focus on a smaller number of diversity criteria and, thus, could result in overall better comparability of methods.

Current models focus mostly on a single criterion. Decision makers, however, often consider multiple criteria or would prefer criteria, besides their primary one, to also be optimized. Lexicographic approaches, as in Bozkaya et al. (2017), could sometimes be included at a very low

Table 2.5: References and basic characteristics for CIT-related work (see Appendix A for an explanation of the instances)

Reference	Problem	Inconsistency	Method	Additional	Instances
Bozkaya et al. (2017)	CVRP-PD-TW	path	ARBIPSA		RWI (Izmet)
Duchenne et al. (2005)	m-PSP	path	B&C		RGI, TSPLIB (burma, gr, fri, bays)
Duchenne et al. (2007)	m-PSP	path	B&C		RGI, TSPLIB
Duchenne et al. (2012)	m-CPSP	path	B&C, B&P		RGI, TSPLIB
Ngueveu et al. (2010b)	m-PVRP	path	ILS, TS	Lower bounds	TSPLIB (bays, fri, gr), A, B, P, vrpnc
Ngueveu et al. (2010a)	m-PVRP	path	Hybrid-TS		A, B, P, vrpnc
Ngueveu et al. (2013)	m-PVRP	path	CG		A, B, P
Talarico et al. (2015a)	kd-VRP	path			R, V
Zajac (2018)	kd-VRP	path			A, B, F, X, tai, vrpnc
Constantino et al. (2017)	DisARP	time	MILP, MH		Art. inst.
Hoogeboom and Dullaert (2019)	PVRP-TW	time	IG-TS	TSC	sol for CIT, RWI (Netherlands)
Michallet et al. (2014)	PVRP-TW	time	MS-ILS	TSC	sol for CIT, RWI (France)
Soriano et al. (2020)	PVRP-TW	time	ALNS	TSC, MG	sol for CIT, RWI (Vienna)
Yan et al. (2012)	MCNFP	time	MH		RWI (Chungli)
Yan et al. (2014)	MCNFP-STT	time	MH		RWI (Chungli)
Allahyari et al. (2021)	S-PDPTW	other	ALNS		sol, Art. inst., RWI (Tehran)
Talarico et al. (2015b)	RCTVRP	other	GRASP, ILS		A, B, F, tai, kelly, vrpnc
Talarico et al. (2017a)	MO-RCTVRP	other	ILS & Promethee II		L, R
Talarico et al. (2017b)	RCTVRP	other	ACO-LNS		R, V, O, S
Tikani et al. (2021)	TD-RCTVRP	other	DP, heuristic DP, GA	Stochastic, MG	RWI (Isfahan)
Radojičić et al. (2018)	RCTVRP	other	F-GRASP & PR		R, V, O, S

PVRP-TW: Periodic Vehicle Routing Problem with Time Windows
m-PSP: m-Peripatetic Salesperson Problem, m-CPSP: m-Peripatetic Capacitated Salesperson Problem
m-PVRP: m-Peripatetic Vehicle Routing Problem, MCNFP: Multi-Commodity Network Flow Problem
CVRP-PD-TW: Capacitated VRP with Pickup and Delivery and Time Windows
kd-VRP: k-dissimilar VRP, RCTVRP: Risk-constrained Cash-in-Transit Vehicle Routing Problem
DisARP: Dissimilar Arc Routing Problem, MO-RCTVRP: Multi-Objective RCTVRP
TS: Tabu Search, IG-TS: Iterated-Granular TS, CG: Column Generation, GA: Genetic Algorithm, DP: Dynamic Programming
ARBIPSA: Adaptive Randomized BI-objective Path Selection Algorithm
MH: Matheuristic, ALNS: Adaptive Large Neighborhood Search, ILS: Iterated Local Search
TD: Time-Dependent, MS-ILS: Multi-Start ILS, GRASP: Greed Randomized Adaptive Search Procedure
F-GRASP & PR: Fuzzy GRASP & Path Relinking, IG-TS: Iterated Granular Tabu Search
MILP: Mixed Integer Linear Programming, B&C: Branch and Cut, B&P: Branch and Price
ACO-LNS: Ant Colony Optimization with Large Neighborhood Search
S-PDPTW: Secure Pickup and Delivery Problem with Time Windows, TSC: Time Spread Constraints, MG: Multi-Graph
Art. inst.: Artificial instances, RGI: Randomly Generated Instances, RWI: Real World Instances
A, B, F, L, O, P, S, R, V, X, kelly, sol, tai, vrpnc, TSPLIB: see Appendix A

computational cost. Bozkaya et al. (2017) aim at minimizing costs and maximizing route inconsistency. They fix the routes and only diversify the arrival times. This computationally simple adaptation improves the time inconsistency, while not altering route inconsistency or costs. As an alternative to lexicographic approaches, methods used for CIT-problems could be extended to generate Pareto sets (sets of non-dominated solutions, where no solution exists that is better regarding at least one objective and equally good regarding all others). This would not only be of interest for models with multiple types of inconsistency but also for models considering just one type, as it is also not trivial to combine costs and security as a weighted sum.

CIT operations also face the problem that it is difficult to measure whether any incidents have actually been prevented due to unpredictable routing. This is further exacerbated by the fact that robberies are rare types of events, which makes it challenging to use measured real-life data for creating and validating models (cf. Prieto Curiel and Bishop (2016)). Therefore, it might be interesting to develop models, where the objective is to perform robberies and that can dynamically adapt to the CIT routes. These findings can then be used in the original models. While such research might seem ethically questionable, it is similar to military operations and might be applied to some of these problems (e.g., drone surveillance and drone strikes). Secondly, the use of such a model in a real-life-application would depend on the amount and quality of information and input data, and thus should not be usable for robbers.

2.6 Dissimilar routing problems

A very restrictive definition of inconsistency might not be desirable, as it allows for a high predictability as soon as parts of the solution are observed. Some of the strictest forms of route-inconsistency are found in inconsistent routing, and, while not all might be of practical interest, the area in itself provides valuable ideas.

In the literature, inconsistent routing is covered by three methodological concepts: k-dissimilar VRP (kd-VRP), m-peripatetic TSP (m-PTSP), and m-peripatetic VRP (m-PVRP). Both the kd-VRP and the m-PVRP aim for a set of routes that include the TSP and VRP constraints: all customers are visited only once; all vehicles start and end at the depot, and the capacity of the vehicle is not exceeded (Talarico et al. (2015a)). However, the differences between these concepts are the following: (i) in the m-PTSP or m-PVRP, the edges can only be used in one specific period, while in the kd-VRP, the multiple usage of an edge is limited by similarity constraints; (ii) the m-PVRP minimizes total cost over all periods, while the kd-VRP aims for solutions, where the worst case cost (over all periods) is minimal (Talarico et al. (2015a)). Another interesting aspect of peripatetic models is that, for safe and secure routing problems, the assumption that edges are not allowed to be traversed more than once might be too restrictive.

An alternate modeling approach is routing problems on multi-graphs, also denoted as road networks. In such an approach, not only the shortest paths but also sub-optimal paths and more detailed network information are taken into account. An overview of positive effects of this modeling approach and a recent state-of-the art survey have been provided by Ticha et al. (2017). Obviously, these multi-graphs can be useful in the field of inconsistent routing. Related literature on multi-graphs is discussed in Section 2.7.

While m-PVRP and kd-VRP problems in the field of CIT, hazardous material, and patrol routing are discussed in Sections 2.3-2.5, Table 2.6 lists the methodological contributions to this problem class. Additionally, we point the reader to studies that we deem to be most relevant for inconsistent routing on multi-graphs. The second column indicates whether the m-PTSP, the m-PVRP, the kd-VRP, or a multi-graph is considered. The third and fourth columns indicate the solution method as well as additional characteristics.

Table 2.6 reveals that the m-PTSP and the m-PVRP have been intensively studied in the literature. However, most of the research regarding the m-PTSP focuses on special cases such as the 2-PTSP, and several of them aim at minimizing costs. Commonly found methods are approximation algorithms, which are algorithms that deliver a solution within a given bound of the best solution with only polynomial run times. More details about the non-m-PTSP are described in Section 2.5, e.g., Duchenne et al. (2005, 2007, 2012), where results for using B&C and B&P on the m-PTSPs have been provided.

2.7 Modeling of multi-graphs

Many routing problems consider only the shortest (or best) paths between nodes. In the case of arcs with different characteristics (e.g., costs and safety), there might not exist a single best path between two nodes. This is the case for most security problems. Two possibilities to address

Table 2.6: Methodological contributions for the m-PTSP, the m-PVRP, the kd-VRP; references for routing problems on multi-graphs.

Reference	Problem class	Method	Additional
Ageev and Pyatkin (2009)	m-PTSP	AA	2-PTSP, metric weights
Ageev et al. (2007)	m-PTSP	AA	2-PTSP-max
Baburin et al. (2009)	m-PTSP	AA	2-PTSP, special weights
De Kort (1992)	m-PTSP	H, LB	2-PTSP
De Kort (1993)	m-PTSP	LB, B&B	2-PTSP
De Kort and Volgenant (1994)	m-PTSP	/	2-GTSP
Duchenne et al. (2005)	m-PTSP	B&C	
Duchenne et al. (2007)	m-PTSP	B&C	
Duchenne et al. (2012)	m-PTSP	B&C, B&P	
De Brey and Volgenant (1997)	m-PTSP	TE	2-PTSP, diverse special weights
Gimadi et al. (2009)	m-PTSP	AA	2-PTSP, special weights
Gimadi and Ivonina (2012)	m-PTSP	AA	2-PTSP-max, special weights
Gimadi et al. (2014)	m-PTSP	AA	2-CTSP, 2-CTSP-max, special weights
Gimadi et al. (2015)	m-PTSP	AA	Asymmetric 2-PTSP & m-PTSP
Gimadi (2015)	m-PTSP	AA	m-PTSP
Gimadi et al. (2016)	m-PTSP	AO	m-PTSP
Gimadi and Tsidulko (2017)	m-PTSP	AA	m-PTSP
Gimadi and Tsidulko (2018)	m-PTSP	AO	m-PTSP-max, Euclidian
Glebov and Zambalaeva (2012b)	m-PTSP	AA	2-PTSP-max
Glebov and Zambalaeva (2012a)	m-PTSP	AA	2-PTSP, special weights
Glebov et al. (2015)	m-PTSP	AA	Asymmetric 2-PTSP-max
Krarup (1995)	m-PTSP	/	/
Ngueveu et al. (2010a)	m-PVRP	Hybrid-TS	
Ngueveu et al. (2013)	m-PVRP	CG	
Shenmaier (2014)	m-PTSP	AA	m-PTSP-max
Talarico et al. (2015a)	kd-VRP		
Zajac (2016)	kd-VRP		
Zajac (2018)	kd-VRP		
Alinaghian and Naderipour (2016)	multi-graph		
Behnke and Kirschstein (2017)	multi-graph		
Fleischmann (1985)	multi-graph		
Garaix et al. (2010)	multi-graph		
Huang et al. (2017)	multi-graph		
Lai et al. (2016)	multi-graph		
Liu et al. (2014)	multi-graph		
Letchford et al. (2014)	multi-graph		
Ticha et al. (2017)	multi-graph		
Ticha et al. (2019)	multi-graph		
Setak et al. (2015)	multi-graph		
Setak et al. (2017b)	multi-graph		
Setak et al. (2017a)	multi-graph		
Tikani and Setak (2019)	multi-graph		
Fröhlich et al. (2020)	multi-graph		

m-PTSP: m-Peripatetic Traveling Salesperson Problem
m-CTPSP: m-Peripatetic Capacitated Traveling Salesperson Problem
m-PVPR: m-Peripatetic Vehicle Routing Problem
CVRP-PD-TW: Capacitated VRP with Pickup and Delivery and Time Windows
kd-VRP: k-dissimilar VRP, max: Maximize objective
TS: Tabu Search, CG: Column Generation, H: Heuristic method, TE: Tailor-made Exact method
B&C: Branch and Cut, B&P: Branch and Price, AA: Approximation Algorithm
AO: Asymptotically Optimal algorithm, LB: Lower Bound

this fact are the use of either an arc-based formulation or a multi-graph. Therefore, an excerpt of relevant publications for the field of security routing is given in the following paragraphs. Problems on road networks are often transformed into VRPs by converting every node, edge, and arc to be serviced into a vertex. The shortest path is calculated between each pair of vertices in both directions, where edges can be traversed in both directions. Thus, a simple-graph is created. Figure 2.9 depicts such a transformation of an NEARP into a VRP. The full nodes, edges, and arcs on the left side require service and are all converted to vertices. Due to the orientation flexibility of the edge, it is converted into a vertex with two sub-vertices. The dashed, double-sided arrows on the right do not represent edges and, instead, represents two arcs going in opposite directions. However, since most risk definitions are either directly (e.g., arc usage) or indirectly (e.g., arrival times) influenced by the chosen path between vertices, a valid approach for security routing is to model a multi-graph, which implies that arcs with the same orientation can exist between two vertices. A multi-graph is depicted in Figure 2.10 with two arcs going from vertex D to B.

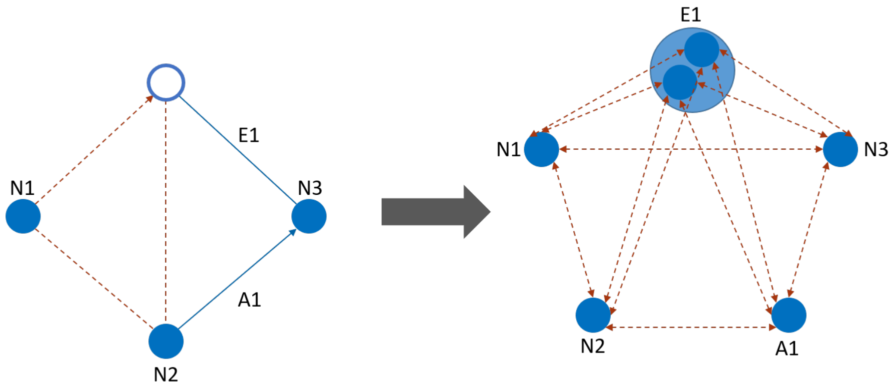


Figure 2.9: Conversion of a NEARP into a VRP.

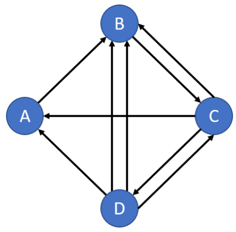


Figure 2.10: An example of a multi-graph with two arcs going from vertex D to vertex B.

Garaix et al. (2010) point out that conversion into a simple-graph VRP can lead to a loss of optimality when multiple attributes such as cost and travel time are defined on the arcs. Hence, they model a dial-a-ride problem and solve it with a heuristic and an exact B&P. In both the cases, they solve underlying fixed sequence arc selection problems that determine the best path for the given solution. Furthermore, they compare their results related to the fixed-sequence arc selection problems with the results on simple graphs for both artificial and real-world instances. The B&P from Garaix et al. (2010) is further improved by Letchford et al. (2014). The authors develop a pricing for elementary and non-elementary routes for a VRPTW, which takes advantage of the sparse graphs.

Similar results as the ones from Garaix et al. (2010) regarding the benefit of using a multi-graph can be found in Lai et al. (2016), where artificial instances are used, as well as in Ticha et al. (2017, 2019) that is based on RWIs. In Lai et al. (2016), a time-constrained heterogeneous VRP is studied. The results from a TS with a heuristic for the fixed sequence arc selection problem and with a mixed integer program are compared, and managerial insights regarding the transportation cost savings due to the multi-graph are provided. In Ticha et al. (2017), an empirical analysis of multi-graphs for a real-world setting is performed. The authors examine a VRPTW, which is solved via B&P on a multi-graph and two simple graphs for the cheapest paths as well as for the least time-consuming paths. Several studies show that RWIs with multi-graphs are tractable. However, Ticha et al. (2017) highlight some limitations and concerns. In particular, computational times increase considerably and correlation between costs and travel times exist, reducing the effect of the achieved benefits. In Ticha et al. (2019), they furthermore develop an ALNS for the problem. In this paper, the approach by Garaix et al. (2010) is improved by a double-sided labeling approach. The authors compare the results of their heuristic on the multi-graph to their optimal results provided in Ticha et al. (2017). The ALNS outperforms the B&P on the simple graphs, which emphasizes the benefit of multi-graphs.

The usage of multi-graphs can also be found in literature on emission-based routing. This likely stems from the fact that emission-efficient paths depend on the load of vehicles and travel times. The latter, however, might change due to congestions during the day. Generally, these studies support the benefits of multi-graphs. However, since they are not within the scope of our survey, we do not provide details on these studies. Interested readers can refer to Setak et al. (2015) and Setak et al. (2017b) where time-dependent VRPs (TDVRP) are studied. Both of them are modeled on multi-graphs with given sets of alternative paths between the vertices. Liu et al. (2014) propose a TDVRP with a heterogeneous fleet and apply a GA to generate results. Another TDVRP is studied in Alinaghian and Naderipour (2016) with a focus on modeling fuel consumption. In Huang et al. (2017), a mixed integer linear program (MILP) is applied to a deterministic and two-stage stochastic model of a TDVRP. Behnke and Kirschstein (2017) focus on an emission-minimizing VRP with heterogeneous vehicles. The authors show how to generate all emission-minimal paths and apply an MILP to solve the problem.

Soriano et al. (2020) study the effect of using a multi-graph for a periodic VRP with time spread constraints on services in the context of safe and secure routing. Fröhlich et al. (2020) propose a NEARP in which costs and route consistency are to be minimized. The authors compare their results for artificial as well as RWIs on a simple and a multi-graph. Improvements can be observed up to a certain instance size. For larger instances, the inherent complexity of the multi-graph does not allow to benefit from working on the multi-graph. In Tikani and Setak (2019), a TDVRP in the context of disaster relief operations is studied. The authors aim at minimizing the delays in delivering prioritized items while requiring routes to stay sufficiently reliable. They approach their non-linear model with three different methods and conclude that using a multi-graph yields significant improvements.

2.8 Conclusion

The focus of our review is on VRPs with safety and security aspects. The three largest, distinct subgroups were CIT operations, patrol routing, and the transportation of hazardous materials. Usually, connectivity between them is not well highlighted in individual publications, and a review did not exist prior to our work. In our opinion, however, major benefits and further improvements in these fields can be achieved when sharing knowledge gained in the individual studies. As such, we aimed to (i) give an extensive overview of the individual fields and analyze what risk measures, methods, objectives, and constraints are commonly found, (ii) provide suggestions for promising adaptations/expansions for the problems, (iii) enable sharing of knowledge from areas with researchers in this field, and (iv) and give information on the instances used within safe and secure routing.

We did this by, first, giving a brief overview of the most common classifications and definitions that can be found, which may prove to be especially helpful for researchers who might be just starting in this area. Afterwards, details pertaining to hazardous material transportation, patrol routing, and CIT operations are presented. Our focus is mainly on the models and their more distinct characteristics, along with the methods used and the most valuable findings. For each of these areas, after summarizing the literature, we have also proposed possible extensions.

One such extension would be a unification of the used instances. The instances in different publications are very diverse; many have considered a case study. A problem associated with such case studies is the underlying data: (i) crimes prevented due to routes being unpredictable or police presence being high cannot be directly measured; (ii) such incidents are rare, making data measurement less reliable; (iii) data of security companies is often sensitive and not willingly shared without certain non-disclosure agreements. We suggest to tackle this unification by generating new benchmark instances of complex and realistic problems (e.g., dynamic travel times, time windows, random congestion, alarms). Solutions for less restricted problems (e.g., ignoring dynamic travel times) could then be evaluated for the master problem with all characteristics. This might provide insight into whether certain characteristics need to be considered for good solutions. Special care should be taken to examine what factors truly affect the likelihood of specific crimes and to what degree. A different approach would be the development of methods aiming to, instead of preventing crimes, perform them. Such methods should be dynamic in order to be able to adapt to the routes of security companies and used as a simulator for testing methods that aim at preventing crimes.

A second consideration for future research paths might be the adaptability of methods and how to handle disruptions or other changes to the system, with one of the most commonly observed one being traffic jams. However, disruptions might also be caused by reactions to sudden incidents and leave formerly well-covered areas less protected. If something like that could be easily observed, it would allow for undesired exploitation.

Since dissimilar routing problems, while currently not being too present in safe and secure routing, also deliver possibly important insights and methods, we have also included this problem class by presenting a brief summary of the relevant literature. Finally, we also looked at multi-graph modeling in more details as they can be a compromise between arc-based formulations, which are high in complexity but allow for more route inconsistencies, and node-based formulations, with the inverse attributes.

We hope to enable and encourage other researchers to follow these suggestions or gain new ideas and investigate this academically challenging, yet practically relevant field.

CHAPTER 3

Secure and efficient routing on nodes, edges, and arcs of simple-graphs and of multi-graphs

Published in: *Networks*
Secure and efficient routing on nodes, edges, and arcs of simple-graphs and of multi-graphs.
Georg E. A. Fröhlich, Karl F. Doerner, Margaretha Gansterer
https://doi.org/10.1002/net.21993

Abstract Many security companies offer patrolling services, such that guards inspect facilities or streets on a regular basis. Patrolling routes should be cost efficient, but the inspection patterns should not be predictable for offenders. Related cash-in-transit services also result in routing problems that require trade-offs between cost minimization and route inconsistency. We introduce this setting as a multi-objective periodic mixed capacitated general routing problem with the objectives being minimization of costs and maximization of route inconsistency. The problem is transformed into an asymmetric capacitated vehicle routing problem, on both a simple-graph and a multi-graph. To solve the problem, we propose embedding an adaptive large neighborhood search procedure into three multi-objective solution frameworks: multi-directional local search (MDLS), the ϵ-constraint heuristic, and the ϵ-box splitting heuristic (EBSH). Tests with both artificial and real-world instances, in an extensive computational study show that for most indicators the ϵ-based methods are dominant, with EBSH being the better of the two. Since MDLS is still better regarding one indicator, a hybrid search procedure, combining EBSH with MDLS, is developed and benchmarked against the individual solution methods. Generally, results indicate that considering more than one shortest path between nodes can significantly increase solution quality for smaller instances, but is quickly becoming a detriment when focusing on larger instances.

3.1 Introduction

Security companies offer a range of services to protect objects (e.g., buildings, cash, other valuables) and people (e.g. personal protection, patrolling services). To plan for and assign their personnel efficiently, these companies must resolve challenging routing problems. Attacks theo-

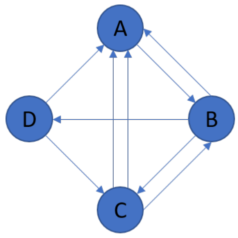


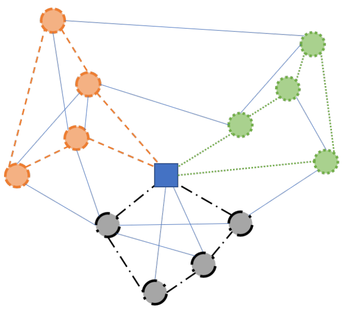
Figure 3.1: Example of a multi-graph with two arcs going from node C to node A

retically could take place anywhere during the route; either at a point being serviced, or while traversing between these point. Sometimes service points might make good targets to intercept transports, but not always; military facilities, central banks, or locations with a high proximity to police stations are well-guarded and, therefore, not attractive targets. In those cases, attacking vehicles while they are moving between them might be the more rational choice. To complicate this for robbers, times of traversing the street segments or the street segments themselves can be varied. We would argue that varying times alone is not sufficient, since this can cause street segments that are always traversed, but only at different times, allowing for easy ambushes. Instead more emphasis should be put on how often street segments are used in combination with how often the sequence of services show similarities.

Both visited nodes and traversed edges are relevant for solving this problem, which accordingly reflects a type of the general routing problem (cf. Orloff (1974)) and the node, edge, and arc routing problem introduced in Prins and Bouchenoua (2005).

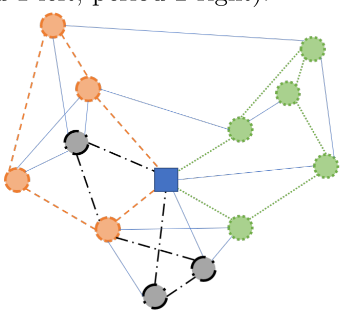
We propose to model this problem as a multi-objective periodic mixed capacitated general routing problem (MO-P-MCGRP), in which nodes, edges, and arcs must be visited regularly but with inconsistent routes to ensure the unpredictability of visiting or service patterns. This problem is NP-hard, since it is a generalization of the capacitated vehicle routing problem (CVRP). Its two objectives are to minimize the cost and to maximize the route inconsistency. Because maximizing route inconsistency and minimizing route consistency are interchangeable, we model the problem with consistency; a multi-objective problem becomes easier to represent if all objectives must be either maximized or minimized. But we will refer to maximizing inconsistency, while measuring route consistency by how often an edge/arc gets traversed and how often pairs of required nodes, edges, and arcs are serviced directly after one another. Since for most street segments ambushes could be set on either side with nearly the same effect, we deemed the direction in which edges are traversed irrelevant for the route consistency. For solving the problem, we also use a conversion into a multi-graph, where several arcs with the same orientation can exist between two nodes. Although MCGRPs often get converted into simple-graphs, this conversion can lead to a loss in solution quality, whereas this issue can be mitigated with the use of a multi-graph. Figure 3.1 depicts a simple multi-graph with two arcs from node C to A.

Figure 3.2 and 3.3 both depict solutions for two periods, in which only the nodes need to be serviced. Figure 3.2 shows a very poor solution for inconsistency in that the sequence of nodes and used edges is the same in both periods. The solution in Figure 3.3 is 7.25% worse for costs



Costs 69.26, consistency 30

Figure 3.2: Example of a node-based routing problem (3 tours) with low inconsistency over 2 periods (period 1 left; period 2 right).



Costs 74.28, consistency 15

Figure 3.3: Example of a node-based routing problem (3 tours) with increased inconsistency (compared to Figure 3.2) over 2 periods (period 1 left; period 2 right).

but 50% better in terms of inconsistency, because fewer edges are repeated, and the sequence of nodes gets changed.

Although any multi-objective problem can be converted into a single-objective problem by weighting the individual objectives and adding them, objectives that differ in their nature might be more difficult to combine, such as a monetary objective (costs) and a non-monetary objective (inconsistency). Therefore, it might be more practical to solve the multi-objective problem by finding the Pareto set, or a set of non-dominated solutions, and then suggesting rules for choosing from this set. Furthermore, decision makers might not want to share their precise preferences regarding objectives a priori (cf. Doerner et al. (2006)), in which case the appropriate conversion of a multi-objective problem into a single-objective is more difficult.

Dominance in the context of multi-objective problems arises in the following way: Assume two solutions $S_1 = \{s_{11}, ...s_{1n}\}$ and $S_2 = \{s_{21}, ...s_{2n}\}$, where s_{ij} refers to the individual objective values that must be minimized. Solution S_1 dominates S_2 if the following holds: $s_{1j} \leq s_{2j}$ for all objectives and $s_{1j} < s_{2j}$ for at least one objective. A solution not being dominated by any solution, is part of the Pareto set. Solutions of the Pareto set can furthermore be classified into supported and non-supported solutions. For this, the relevant criterion is whether there exists a combination of weights for a weighted sum of objectives such that the solution is optimal. For example, assume $\{1, 0\}$, $\{0.51, 0.51\}$, and $\{0, 1\}$ are non-dominated solutions for a problem; $\{1, 0\}$ and $\{0, 1\}$ are supported, but $\{0.51, 0.51\}$ is not. The conversion of a multi-objective problem into a single-objective problem via weighting the objectives would therefore only allow supported solutions of the Pareto set to emerge and thus might exclude or ignore some promising trade-off solutions.

In our effort to contribute to this field of research, we:

- Introduce our proposed MO-P-MCGRP as a means to minimize cost and consistency (maximize route inconsistency) and mathematically formulate its conversion to a node-based formulation on a multi-graph.
- Develop an adaptive large neighborhood search (ALNS)-based algorithm for the single-objective problem and embed it into three multi-objective frameworks.
- Analyze performance using artificial and real-world instances for simple- and multi-graphs. All instances are made publicly available on the BDA homepage (bda.univie.ac.at).
- Deliver insights on the trade-off between cost and inconsistency according to the real-world instances.
- Show that the solutions benefit from a higher degree of freedom provided by the multi-graphs and also specify how the gain may differ with target performance indicators and problem sizes.

In Section 3.2 we discuss related literature. Then in Section 3.3, we provide a problem description for the MO-P-MCGRP and the conversion of the MO-P-MCGRP, followed by the ALNS procedure and its incorporation into multi-objective frameworks in Section 3.4. After we present the computational study in Section 3.5, we summarize the findings and their implications in Section 3.6. Additionally, the detailed mathematical model of the conversion of the MO-P-MCGRP can be found in the appendix.

3.2 Literature Review

Reflecting our focus on a MO-P-MCGRP and security aspects, we discuss research contributions pertaining to (i) routing with security aspects, (ii) MCGRP, (iii) multi-graphs, and (iv) multi-objective problems.

A safe and secure routing problem arises when cash or patrol guards must be transported or scheduled. In cash-in-transit operations and patrolling, unpredictability is a crucial issue that usually is addressed by paths that are inconsistent in their routes or times. Route inconsistency requires the adoption of different edges and arcs; time inconsistency implies performing the service at different points in time. We focus on route inconsistency. The m-peripatetic vehicle routing problem (m-PVRP), which aims at route inconsistency, is analyzed in Ngueveu et al. (2010a), where a hybrid tabu search is used. In Wolfler Calvo and Cordone (2003) the authors consider a security problem with three service types and assume that some nodes must be visited only once, while others are served multiple times. A third set of services refers to alarms, which are not known a priori. The authors use a two-stage approach, such that they first cluster the alarms according to a capacitated p-median algorithm and then solve the routing and scheduling problems with an adapted version of alternate k-exchange reduction, which builds a set of initial solutions and the improves them via local search (Cordone and Wolfler Calvo (2001)). Cost-effective routing including security aspects is considered in Yan et al. (2012), where the problem is split into subproblems, solved with CPLEX. A solution is considered secure if not too many routes within it are similar, defined by the number of demand points visited at similar points in time. In studies of a periodic VRP with time windows and time spread constraints, authors use an iterated granular tabu search (Hoogeboom and Dullaert (2019)) or a multi-start iterated local search (Michallet et al. (2014)). Visiting a node at a point in time that is too close to a previous visit of the node is deemed insecure. Multiple heuristics, like iterated local search and an ant colony optimization for the risk constrained cash-in-transit VRP (Talarico et al. (2017b, 2015b)), might involve risk defined as the amount of money being transported at a given point in time. The proposed methods aim to find cost effective solutions, at a risk level below a given threshold. The methods' authors provide several new benchmark instances for their problem. In Talarico et al. (2017a) the authors introduce the multi-objective risk-constrained problem with two objectives, namely cost and risk minimization. Risk in their case is the sum product of money and the distance it gets transported. Their multi-objective optimization method is based on iterated local search. However, they do not focus on generating a Pareto set approximation but instead attempt to integrate the decision maker's preference into the optimization process by providing a subset of appropriate solutions.

The single-objective MCGRP has been analyzed from several research groups. In Bosco et al. (2013) the authors use an integer programming model and branch-and-cut algorithm to introduce the problem and provide benchmark instances (MGGDB). In Bosco et al. (2014) a matheuristic is applied to the newly introduced problem. This method consists of a destroy-and-repair algorithm for diversification, a variable neighborhood descent for local search, and the aforementioned branch-and-cut algorithm. A memetic algorithm for the MCGRP as well as a new set of instances (CBMix) are introduced in Prins and Bouchenoua (2005). Further instances (BHW and DI-NEARP) as well as a lower bound procedure are provided in Bach et al. (2013). Most of the currently best known results for these instances have been firstly published in Dell'Amico et al. (2016), where an adaptive iterated local search is used, and Vidal (2017), where the multi-start iterated local search from Prins (2009) and the unified hybrid genetic search from Vidal et al. (2012) and Vidal et al. (2014) are applied. Multi-objective versions of the MCGRP have also been studied in Halvorsen-Weare and Savelsbergh (2016). The authors focus on cost and route balance, with multiple different measurements. A set-partitioning model and the box splitting method from Hamacher et al. (2007) are used to solve some of the smaller MGGDB instances.

Several heuristics convert the MCGRP into a VRP with the shortest paths connecting the nodes of the VRP, but for some routing problems conversion schemes into a simple-graph are not sufficient. This has been pointed out in Garaix et al. (2010). The authors noted that once multiple attributes are defined on arcs or edges, a transformation of a problem into a standard VRP can lead to the risk of losing optimality and good solutions. They conducted experiments with simple- and multi-graphs and showed that improvements could be gained with the latter. In Lai et al. (2016) the authors performed some similar experiments for a heterogeneous VRP with limited duration with two arcs between any pair of vertices. They made similar observations as

Garaix et al. (2010). A multi-graph for a time dependent alternative VRP, where time windows and travel times depended on the time of the day, was studied in Wang et al. (2014). From the two edges between any pair of nodes, the first had a time-dependent travel speed distribution, while the second had an, except for peak hours, far longer, but constant travel time. A multi-graph is also used in Soriano et al. (2020), where a VRP with arrival time diversification is investigated. The authors assume that customers must be serviced multiple times. The aim is to optimize costs, while having sufficiently different service times at each customer. It is shown that the increased flexibility due to the multi-graph enabling better solutions.

For multi-objective problems several solution approaches have been developed. In Matl et al. (2017) the authors compared the ϵ-constraint heuristic (ECH) and ϵ-box splitting heuristic (EBSH) while using the hybrid genetic search from Vidal et al. (2012). They then compared their method to the greedy randomized adaptive search procedure with advanced starting point introduced in Oyola and Løkketangen (2014) and the multi-start split-based path relinking in Lacomme et al. (2015). However, the results of Matl et al. (2017) show that ECH and EBSH are significantly better than the other two. Multi-directional local search (MDLS) was proposed in Tricoire (2012). The authors show that they perform comparable to other state-of-the-art methods for multiple problems like the non-dominated sorting genetic algorithm (Deb et al. (2002)), the strength Pareto evolutionary algorithm 2 (Ziztler et al. (2002)), the multi-objective genetic local search (Alves and Almeida (2007)), and the multi-objective evolutionary algorithm based on decomposition (Zhang and Li (2007)). MDLS, ECH, and EBSH need an underlying heuristic. For this, we use ALNS, which is a renowned metaheuristic that has been successfully applied to several planning problems (e.g. Ghilas et al. (2016); Pisinger and Ropke (2007); Soriano et al. (2018)) including a multi-graph in Ticha et al. (2019). Furthermore, an ALNS was presented in Dell'Amico et al. (2016), where it was used to generate benchmark results for several of the MCGRP instances. Hence, it is considered a valid approach for our extension.

Compared to the found literature on MCGRP we are, to the best of our knowledge, the first to tackle the MCGRP with the two objectives cost minimization and route inconsistency in a multi-objective setting. Route inconsistency is measured by the number of times an arc or edge is traversed, excluding service, and by how often required elements are visited in the same order. The former aspect of route inconsistency has similarities to the m-PVRP (e.g. Ngueveu et al. (2010a)), since it is based on edges and arcs. Conversions of the MCGRP into a simple-graph have already been performed in Dell'Amico et al. (2016) and Vidal (2017), but not into a multi-graph, since their problems were not facing the risk of loss of optimality as pointed out in Garaix et al. (2010) when converting them. Additionally, MDLS, ECH, and EBSH have not been used on a multi-graph before.

3.3 Problem formulation

3.3.1 Problem description

The MO-P-MCGRP can be represented on a graph $G_{MO-P-MCGRP} = (N \cup E \cup A)$, where N, E, and A are the sets of nodes, edges, and arcs, respectively. Every period, subsets, composed of the *required* nodes $N_R \subseteq N$, edges $E_R \subseteq E$, and arcs $A_R \subseteq A$, must be serviced by a fleet of vehicles that must start and end their routes at the depot, each with a limited capacity. Traveling on edges and arcs causes costs. For the required nodes, edges, and arcs, we consider additional costs for service and demands.

The MO-P-MCGRP has two objectives: (i) cost minimization and (ii) maximization of route inconsistency (expressed as minimization of route consistency) between different periods.

For route inconsistency, we assume an adapted version of the R-type distance (Martí et al. (2005)). For two sequences $p = (p_1, ..., p_n)$ and $q = (q_1, ..., q_m)$, the R-type distance $d(p, q)$ is the number of times p_i does not immediately follow p_{i-1} in q for $i = 1, ..., n-1$. We adapt this measure so that we can consider inconsistency in regard to nodes, edges, and arcs. We assume that inconsistency diminishes if:

1. An edge or arc is used more than once ignoring the direction, in which edges are traversed. This point does not apply to required edges or arcs (E_R, A_R), if they are serviced.

2. Two required nodes, edges, or arcs directly follow each other. This point applies even if non-required nodes, edges, or arcs are traversed in between them.

For this, we determine how often (x) every edge and arc is traversed (excluding service) and set the generated consistency v_l of the individual segment l to $x - 1$. Similarly, we determine how often (y) a specific required node, edge, or arc i is serviced directly before servicing a second required node, edge, or arc j, and set the generated consistency v_{ij} to $y - 1$.

3.3.2 Transformation of the Mixed Capacitated General Routing Problem to the Capacitated Vehicle Routing Problem

The MCGRP can be transformed to an asymmetric CVRP on a complete graph $G_{CVRP} = (V, A)$ (Vidal (2017)). The union set of required nodes, edges, and arcs (N_R, E_R, and A_R) and the depot N_0 composes a set of nodes $V := N_R \cup E_R \cup A_R \cup N_0$. We use the term point of interest (POI) to refer to any node resulting from this transformation, to differentiate them from the nodes on the original graph. The paths between all POIs in V are calculated to create a complete set of arcs A. However, POIs that were edges before need special consideration. They can be represented as two sub-POIs, from which only one must be visited, or there might be multiple shortest paths between them and other POIs, which would result in a multi-graph.

Figure 3.4 shows such a transformation. The three full nodes, the full edge, and the full arc require service. The arc is converted into a POI, and the edge is converted into one POI with two sub-POIs, of which only one must be visited. The double-sided arrows simplify the figure and do not represent edges but instead reflect two arcs going in opposite directions. It is important to note that paths of the conversion often consist of several edges and arcs from the original problem. This information has to be stored, to calculate route inconsistency. Figure 3.5 shows the original edges and arcs used for the path from node 3 (N3) to node 1 (N1) and for the path from node 1 (N1) to node 2 (N2).

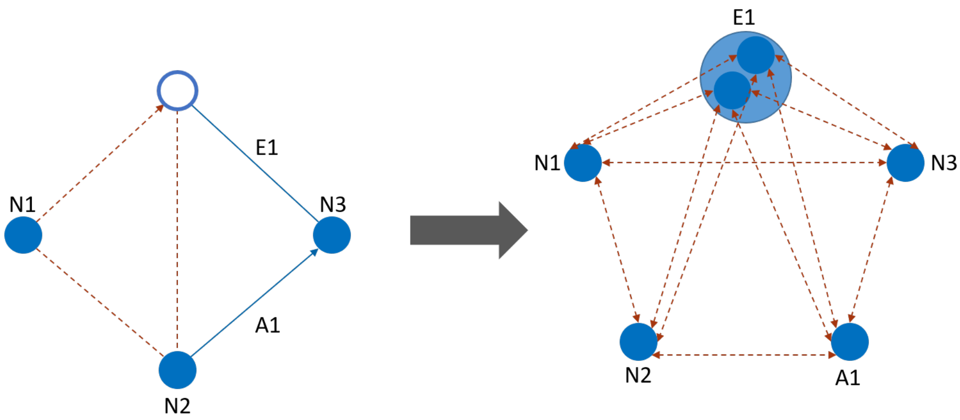


Figure 3.4: Transformation of a MCGRP to a CVRP. Full circles and lines represent required nodes/edges/arcs. Double-sided arrows represent two arcs going in opposite directions.

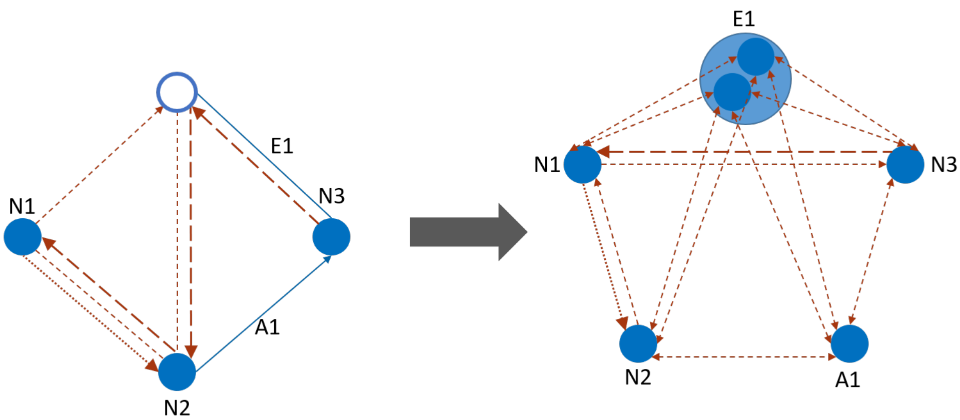


Figure 3.5: Paths of created CVRP on the original MCGRP. Bold dotted and dashed line in the MCGRP illustrate the paths used for the bold dotted and dashed line in the CVRP.

3.3.3 Transformation of the Multi-Objective Periodic Mixed Capacitated General Routing Problem

The transformation in Section 3.3.2 has some limitations when being applied in the same manner to the MO-P-MCGRP. In the case of the standard MCGRP, the objective is to minimize costs and calculating the shortest paths between the new POIs is sufficient to guarantee that an optimal solution on G_{CVRP} is an optimal solution on G_{MCGRP}. This outcome does not hold for the MO-P-MCGRP. Instead, it requires a multi-graph, in which multiple arcs can exhibit the same orientation between two nodes.

To guarantee that the non-dominated solutions on the multi-graph are also non-dominated on the original graph, all non-dominated paths between two POIs for a given orientation must be included in the multi-graph, which is easy to show:

1. When having all possible paths a non-dominated solution on the multi-graph is non-dominated on the original graph.

2. Exchanging a dominated path with a path that dominates it only improves a solution.

In this case, a path $p_{ij} = \{v_1, ..., v_m\}$ between POIs i and j, consisting of segments v_1 to v_m, dominates a path $q_{ij} = \{w_1, ..., w_m\}$, if there exists at least one segment of q_{ij} that is not in p_{ij}, and every segment of p_{ij} is in q_{ij}.

3.4 Methods

The preprocessing of the multi-graph and the ALNS and the three multi-objective frameworks (MDLS, ECH, EBSH) and a combination of them (EBSH*), which share the former two as basic components, are described in the following subsections. We decided to use an ALNS with similar mechanism as presented in Dell'Amico et al. (2016). The dominance of this method is shown on the SINTEF homepage (Sintef.no (2012)), where benchmarks for several instances along with the respective publications are presented.

Note that in Vidal (2017) slightly better results with an alternative solution representation are presented. This representation is used in the genetic algorithm by Vidal et al. (2012) and the iterated local search by Prins (2009). These results are, however, not provided on the SINTEF homepage. According to Vidal (2017), the approach relies on the solution representation and very efficient move evaluation tailored for the single-objective problem. Since an adaption to the multi-objective problem is not straightforward, and ALNS comes very close in terms of solution quality, we decided to stay with ALNS.

3.4.1 Basic component: Multi-graph generation

For creating multi-graphs, we reimplemented an algorithm described in Liu et al. (2017) to determine the k-shortest paths with a specified diversity. Note that other recent research on multi-graphs, where non-dominated solutions are created, does exist (e.g. Ben Ticha et al. (2019), Letchford et al. (2014)). However, they focus on a VRP with time windows, where cost and traversal time are taken into account. This allows for far easier and stronger cuts based on domination during the paths generation. Furthermore, it should be noted that the road-network of our real-world instances contains more than 20,000 nodes and 50,000 edges and arcs. This is by far larger than the largest networks in Letchford et al. (2014), where 500 nodes and 744 edges and arcs are included. Besides the problem of computational complexity in producing large networks, additional paths cause a higher computational effort during the ALNS. Therefore, we decided to rather generate a low number of high quality paths; results in Section 3.5.3 indicate that this was already sufficient for even slightly larger instances, as they deteriorate for some indicators when using three paths.

For the individual multi-graph segments between two nodes s and t with k-diverse shortest paths, we start with generating the first shortest path via the Dijkstra algorithm (Dijkstra (1959)). From this path $\{s, x_1, ..., x_n, t\}$ we generate all partial, deviating parts without loops and enter them into a priority queue (see Algorithm 1, lines 1-4). For this, first all partial parts $p_i = \{s, x_1, ..., x_i\}$ are generated. Then every partial path is extended from the end node x_i to every possible neighbor $n_j(x_i)$, except to the former successor. The resulting sets of deviating partial paths could be described as $P(p_i) = \{\{s, x_1, ..., x_i, n_j(x_i)\} | n_j(x_i) \neq x_{i+1}\}$. Thereafter, we enter a loop for extending the most promising partial path in the priority queue until we have either generated the remaining $k-1$ paths or there are no paths to extend. The partial paths are extended in the same manner as above, unless they do not have a successor. Obviously, paths that create loops ($\{s, ..., x_i, ..., x_i, ...\}$) or that are already dominated by other paths are removed (see Algorithm 1, lines 5-13).

To determine the most promising path, a lower bound for the minimum length required to satisfy the diversity is calculated. To measure similarity between two paths S_i and S_j, we use the intersection length ($\frac{L(S_i \cap S_j)}{L(S_i \cup S_j)}$), where we assume a maximum of 80 percent. This measure of similarity deviates from the one in the multi-objective frameworks later used, but also strengthens the bounds. Thus, the path generation algorithm is accelerated significantly. Therefore, we would receive the lower bound (LB) for a partial path p due to an already accepted path S_i of

$$\frac{L(S_i \cap p)}{L(S_i \cup p)} \leq 0.8 \Rightarrow LB(p) \geq 2.25 \times L(S_i \cap p) - L(S_i). \tag{3.1}$$

In addition to the lower bounds from already accepted paths, there is also the lower bound for the shortest distance from the current end node x of a partial path p to t, resulting in

$$LB(p) \geq L(p) + L(x, t). \tag{3.2}$$

Algorithm 1 Multi-graph generation for k paths between s and t

```
1:  S_1 ←performDijkstraAlgorithm(s, t)
2:  S ← S_1
3:  PQ ←generateAllPartialPaths(S_1)
4:  PQ ←filterLoopingPathsAndSetLowerBounds(PQ, S)
5:  while !PQ.empty ∧ S.size < k do
6:      NP ←extendMostPromisingPath(PQ)
7:      NP ←filterLoopingPathsAndSetLowerBounds(NP, S)
8:      PQ ← PQ ∪ NP
9:      if S_{S.size+1} ←includesNewFullPathSatisfyingDiversity(NP) then
10:         S ← S ∪ S_{S.size+1}
11:         PQ ←setLowerBounds(PQ, S)
12:     end if
13: end while
14: return S
```

3.4.2 Basic component: Adaptive Large Neighborhood Search

Using destroy and repair operators (d and r), ALNS iteratively selects pairs to apply to the incumbent solution s. The selection is based on scores updated in regular intervals of size ν, assigned to the used operators on the basis of their performance. Finding a new best solution s^*, a new incumbent solution s, or a previously unidentified solution are events that prompt the awarding of a score to the pair of operators. Following any updates, the old scores are replaced by new scores. The new solution s' then undergoes an acceptance test via simulated annealing (SA) to determine, whether it is accepted as new incumbent solution or not. SA has a *temperature* and a *cooling* parameter. As long as the temperature is high, the likelihood of solutions being worse than the current one to be accepted is also high - promoting diversification. Cooling indicates how much the temperature is lowered in each iteration. This process repeats until it reaches a given stopping criterion (see Algorithm 2, lines 1-4, 8, 14-20).

For the operators, we mostly follow Pisinger and Ropke (2007):

1. **Destroy Random**: Random POIs are removed.

2. **Destroy Worst**: The most beneficial removals are executed.

3. **Destroy Related**: A random POI r is selected and removed together with a number of POIs that are closest in distance to r.

4. **Destroy Small**: POIs with the smallest demands are removed.

5. **Destroy Route**: A route is randomly selected. All POIs included in this route are removed.

6. **Destroy Historical Pair**: A pair of POIs ij with the largest historical values is removed. The historical value of ij is the best objective value observed for a route that includes ij.

7. **Repair Random**: POIs are inserted into random positions.

8. **Repair Greedy**: POIs are sequentially inserted into the position that leads to the smallest increase in cost.

9. **Repair Regret**: POIs are sequentially inserted into the position with the highest opportunity cost.

For all operators except for 1, 5, and 7, we include noise factors u, which we randomly draw from an interval $U = [1 - \psi, 1 + \psi]$. The values used to select POIs are multiplied by $1 + u$, which increases diversification, because operators 1 and 7 are only used if no new best solution has been found for several iterations.

For further diversification, we design the operators such that promising but infeasible solutions are generated as well. For this effort, capacity and vehicle violations are included but penalized. We adjust penalties during the search procedure, depending on the number of observed violations. The objective values used for determining whether a new solution should be accepted or not are as follows:

1. $s.objValue = s.cost + \omega_{cap} \times s.vCap + \omega_{route} \times s.vRoute$

2. $s.objValue = s.cons + \omega_{cap} \times s.vCap + \omega_{route} \times s.vRoute$

3. $s.objValue = s.cost + \omega_{cap} \times s.vCap + \omega_{route} \times s.vRoute + \omega_{cons} \times s.vCons$

The first and second values apply to the MDLS, and the third applies to the EBSH and ECH. Here, ω_{cap}, ω_{route}, and ω_{cons} are the penalties for each unit of capacity, route, and consistency violation, respectively.

The ALNS is strengthened by a variable neighborhood decent (VND), which initiates if an incumbent solution is within a given percentage of the best solution or better. Hence, the VND intensifies good solutions (see Algorithm 2, lines 5-7). The following operators then follow the first fit procedure:

1. **Swap**: Two POIs are swapped.
2. **Or-Opt Route**: Three paths of a route are removed and reconnected such that the POI sequences remain unchanged.
3. **2-Opt Route**: Two paths of a route are removed and reconnected.
4. **2-Opt Different Routes**: Two paths of different routes are removed and reconnected.
5. **Flip**: A path is removed and reinserted after the sequence being flipped.

If no new best solution results after a given number of iterations ϕ, operators 1 and 7 as well as VND are applied to the incumbent solution (see Algorithm 2, lines 9-13).

Considering route inconsistency increases the amount of stored data for the ALNS significantly, because we store for every path a vector of components, whose average size typically increases with larger instances. Furthermore, the move evaluation becomes far more costly, since for every path to be changed, the changes in inconsistency due to its components (i.e., edges and arcs of the original MO-P-MCGRP) must be calculated.

Consequently, by using a multi-graph, the average computation time of almost all operators is increased as well. In particular because of the increased number of operations that must be evaluated. For destroy operators (besides operator 2), the increase is negligible, since only disruptions in the routes after removing POIs must be fixed. This is done by finding the path, where the smallest increase in objective value is observed. For destroy operator 2, the best path to reconnect the multi-graph has to be determined, every time removal benefits are observed. Complexity increases of repair operators 8 and 9, where calculation of insertion costs requires to determine the pair of paths yielding the smallest increase in objective value. These time consuming operations, quadruple on a multi-graph with two paths, and increase by a factor of 9 if three paths are considered. We tried to reduce this increase by cutting-off calculations once we observe that insertion costs will not decrease. However, computational complexity is still huge. The same applies to VND's neighborhoods, where, however, the impact is reduced due to VND's share of computation time being smaller.

3.4.3 Solution approach: Adaptive Large Neighborhood Search-based Multi-Directional Local Search

For MDLS, introduced in Tricoire (2012), first a set S of non-dominated solutions is created. This is followed by a random selection of a solution out of this set. A single-objective heuristic for every

Algorithm 2 Adaptive-Large-Neighborhood-Search

$\{s', s, s^*\} \leftarrow$generateInitialSolution()
while $stoppingCriterionNotReached$ **do**
  $\{d, r\} \leftarrow$selectDestroyAndRepairOperators($D, R, scores$)
  $s' \leftarrow r(d(s))$
  if $s'.objective < \mu \times s^*.objective$ **then**
    $s' \leftarrow$VND(s')
  end if
  $\{s, s^*, scores', f\} \leftarrow$acceptanceTest($s', s, s^*, scores', d, r, f$)
  if $f = \phi$ **then**
    $s \leftarrow$randomDestroyAndRepair(s)
    $s \leftarrow$VND(s)
    $f \leftarrow 0$
  end if
  $n \leftarrow n + 1$
  if $n = \nu$ **then**
    $\{scores, scores'\} \leftarrow$updateScores($scores, scores'$)
    $n \leftarrow 0$
  end if
end while
return s

objective is applied, such that a new solution per objective is created. Non-dominated solutions are added to S. The procedure is repeated until a stopping criterion is met (see Algorithm 3 and Figure 3.6).

The ALNS described in Section 3.4.2 is used to optimize costs and inconsistency. To create the initial set of non-dominated solutions, we use the proposed ALNS, such that we generate:

- 3 solutions using the single-objective ALNS for costs.
- 1 solution by iteratively including previously non-inserted POIs in each period, to fulfill capacity and inconsistency restrictions. If there is no such position available, then we insert the POI at a random position.
- 30 solutions using a random construction heuristic, such that the POIs are selected randomly and inserted into a random route that can meet capacity restrictions.

We then filter the non-dominated solutions from this set of 34 solutions.

Algorithm 3 Multi-Directional-Local-Search

$S \leftarrow$generateInitialSolutions()
while $stoppingCriterionNotReached$ **do**
  $s \leftarrow$selectRandom(S)
  for $objective \in Objectives$ **do**
    $s' \leftarrow$ALNS$_{objective}$(s)
    $S \leftarrow S \cup s'$
  end for
  $S \leftarrow$filterNonDominatedSolutions(S)
end while
return S

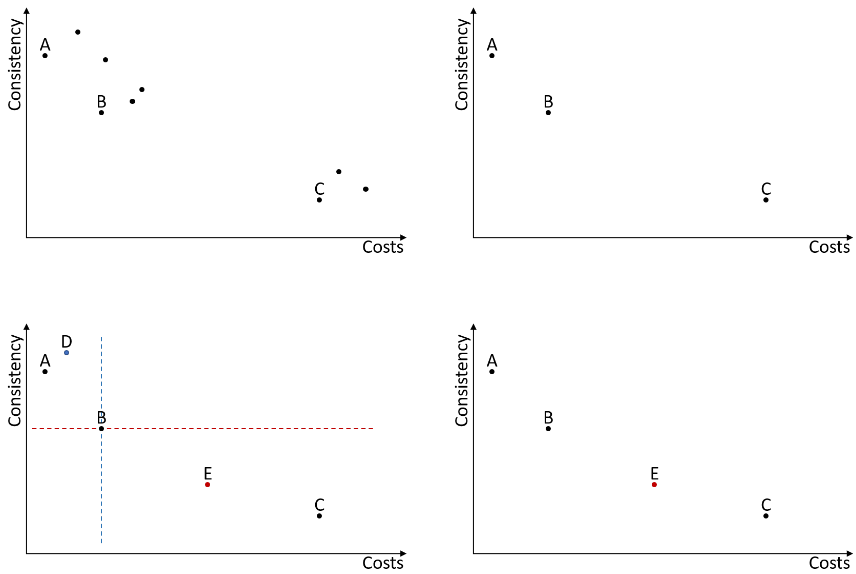


Figure 3.6: MDLS - initialization and first iteration. Top left: Initially generated solutions; Top right: Non-dominated solutions from the initial set; Bottom left: Old solutions with two newly created solutions D & E from the randomly selected solution B; Bottom right: Non-dominated solutions from the old and newly created solutions.

3.4.4 Solution approach: Adaptive Large Neighborhood Search-based ϵ-Constraint Heuristic

The second multi-objective solution approach for the MO-P-MCGRP is ECH, which is based on the ϵ-constraint method (Haimes (1971)). First, it generates an initial solution by solving the single-objective problem with the aim to minimize costs using ALNS (see Section 3.4.2). This solution's consistency (*secondObjective*) provides a constraint (ϵ-constraint), when solving the single-objective problem for the cost objective via ALNS, which has a time limit. This step repeats with the least consistent solution found so far (*S.last*) until no new solution fulfilling the ϵ-constraint can be found or a time limit for the ECH has been reached (see Algorithm 4) .

Algorithm 4 ϵ-Constraint-Heuristic

$S \leftarrow$generateInitialSolution()
$\epsilon \leftarrow S.last.secondObjective$
do
  $newSolutionFound \leftarrow false$
  $s' \leftarrow \text{ALNS}_{cost}(S.last, \epsilon)$
  if $s'.secondObjective < \epsilon$ **then**
    $S \leftarrow$ filterNonDominatedSolutions($S \cup s'$)
    $newSolutionFound \leftarrow true$
    $\epsilon \leftarrow S.last.secondObjective$
  end if
while $newSolutionFound$
return S

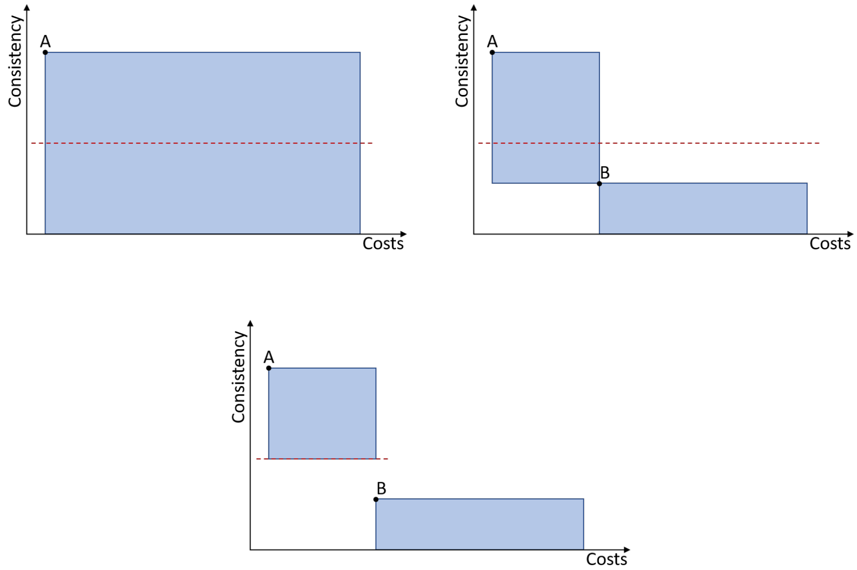


Figure 3.7: EBSH - Updating boxes. Top left: Box of solution A before splitting with red line representing ϵ-constraint; Top right: Boxes of solutions A and B after splitting; Bottom: Boxes of solutions A and B after splitting and removing examined solution space of A's box.

3.4.5 Solution approach: Adaptive Large Neighborhood Search-based ϵ-Box-Splitting-Heuristic

Similar to ECH, EBSH starts by solving the single-objective problem for the first objective heuristically (see Section 3.4.2). After determining a box of the non-dominated area, an iterative procedure starts, in which: (i) parameter ϵ is calculated on the largest box by splitting the box in two equal halves, (ii) the single-objective problem including the ϵ-constraint gets solved via ALNS, and (iii) the boxes and solutions are updated (Matl et al. (2017)). The algorithm stops if no boxes are left or a time limit is reached (see Algorithm 5).

To set the box sizes, this procedure assumes that both objectives are to be minimized. The first box uses the initial solution as one corner point and the lower/upper bound for the second/first objective. If a new solution emerges, all boxes are updated, and the dominated area is removed. If a box is split and no new solution emerges, the examined area can be removed as well, because there is thus a high probability that no Pareto solution is included (see Figure 3.7).

3.4.6 Solution approach: Adaptive Large Neighborhood Search-based two-stage approach of ϵ-Box-Splitting-Heuristic and consistency-ALNS

Our results for the real-world instances in Section 3.5.3 indicated that EBSH was delivering on average the best results for all except one indicator, for which MDLS delivered the best instead. This was mostly due to MDLS finding more inconsistent solutions, with just very few instances

Algorithm 5 ϵ-Box-Splitting-Heuristic

$S \leftarrow$generateInitialSolution()
$B \leftarrow$initializeBoxes()
while $stoppingCriterionNotReached$ **do**
  $\{s, \epsilon\} \leftarrow$findLargestBox$(S, B)$
  $s' \leftarrow$ALNS$_{cost}(s, \epsilon)$
  if $s'.secondObjective < \epsilon$ **then**
    $\{S, B\} \leftarrow$filterNonDominatedSolutionsAndUpdateBoxes(S, s', B)
  end if
end while
return S

where MDLS found more cost-efficient solutions. Therefore, we decided to combine EBSH with just the consistency-ALNS of the MDLS into a two-stage approach, which is later called EBSH*. First, the EBSH is performed as before (see Algorithm 6, lines 1-9). Then the least consistent solution and solutions being within a certain percentage of it are used as initial solutions S' for the second stage (see Algorithm 6, line 10). For the second stage similarly to MDLS, solutions are picked at random and improved via the consistency-ALNS until a stopping criterion is reached (see Algorithm 6, line 11-15), after which the non-dominated solutions from both stages are determined and returned (see Algorithm 6, line 16-17).

Algorithm 6 ϵ-Box-Splitting-Heuristic*

1: $S \leftarrow$generateInitialSolution()
2: $B \leftarrow$initializeBoxes()
3: **while** $stoppingCriterionNotReached$ **do**
4:   $\{s, \epsilon\} \leftarrow$findLargestBox$(S, B)$
5:   $s' \leftarrow$ALNS$_{cost}(s, \epsilon)$
6:   **if** $s'.secondObjective < \epsilon$ **then**
7:     $\{S, B\} \leftarrow$filterNonDominatedSolutionsAndUpdateBoxes(S, s', B)
8:   **end if**
9: **end while**
10: $S' \leftarrow$filterLeastConsistentSolutions(S)
11: **while** $stoppingCriterionNotReached$ **do**
12:   $s \leftarrow$selectRandom(S')
13:   $s' \leftarrow$ALNS$_{consistency}(s)$
14:   $S' \leftarrow$filterNonDominatedSolutions$(S' \cup s')$
15: **end while**
16: $S \leftarrow$filterNonDominatedSolutions$(S \cup S')$
17: **return** S

3.5 Computational study

In the first part of the computational study, we apply the proposed ALNS to publicly available, standard MCGRP instances (Bach et al. (2013); Bosco et al. (2013); Prins and Bouchenoua (2005)). We show that the algorithms yield good results for single-objective problems. In the second part, we focus on multi-objective problems and apply MDLS, EBHS, and ECH to both artificial and real-world instances. Tables 3.1 and 3.3 list the parameter settings for the single-

Table 3.1: Parameters for single-objective case without any tuning

Destroy & Repair Operators	2-6 & 8-9
Noise ψ	0.05
Starting violations for capacity ω_{cap}, consistency ω_{cons}, and routes ω_{route}	$\{1000, 200, 1000\}$
Adaption for violations	Increase/decrease by 5%
Scores for ALNS	$\{10, 3, 1\}$
Cooling factor SA	0.999
Threshold of VND μ	1.05

and multi-objective results; for the single-objective case, they were set similar to ones found in the literature, whereas we tuned them via Irace (López-Ibáñez et al. (2016)) for the multi-objective case.

3.5.1 Single-objective results

We use several benchmarks to assess the performance of the proposed ALNS for the single-objective problem: MGGDB Bosco et al. (2013), BHW Bach et al. (2013), CBMix Prins and Bouchenoua (2005), and DI-NEARP Bach et al. (2013). These sets comprise a total of 205 instances, and except for the MGGDB, none of the instance types limits the number of vehicles.

The instances of MGGDB are rather small, ranging between 8 and 48 POIs, created from the 23 GDB instances. Six variations were created for each of those instances, by shifting the demand from required edges and arcs to adjacent nodes. The percentage of shifted required edges and arcs is indicated in the name by the middle part called β (e.g. mggdb_0.50_1 has 50% shifted). Therefore, instances with a higher β tend to have fewer POIs than those with a lower β. Furthermore, required nodes have higher demand than required edges and arcs, on average. Both BHW and CBMix represent medium size instance sets, such that CBMix instances range from 20 to 212 POIs, and BHW instances range from 20 to 410 POIs, though most are within the span of 110 to 240. The 24 DI-NEARP instances are the largest, with 240, 422, 442, 447, 699, and 833 POIs, and 4 capacities for each. However, they do not include required arcs.

The ALNS was applied 15 times to every instance with a time limit after preprocessing of 1 hour, which is similar to Dell'Amico et al. (2016) and Vidal (2017). Table 3.2 lists the best, average, and worst percentage gap, relative to the best known results of the instances $(\frac{res_{alns}-res_{bench}}{res_{bench}})$.

The ALNS, sharing all the core components presented in Dell'Amico et al. (2016), delivers results very close to the benchmarks. Thus, we deem our implementation well-suited for the multi-objective frameworks.

3.5.2 Performance indicators for multi-objective results

In accordance to Schilde et al. (2009), we select four indicators to evaluate the multi-objective results, using a wide spectrum of criteria. The solutions were normalized to the range $[1.0, 2.0]$, where 1.0 and 2.0 are the best and worst results of the non-dominated solutions, respectively.

Table 3.2: ALNS results for single-objective standard instances. We report average gaps.

Instance type	Best Gap	Avg. Gap	Worst Gap
MGGDB	0.00%	0.00%	0.00%
BHW	0.00%	1.57%	4.74%
CBMix	0.00%	2.23%	6.35%
DI-NEARP	1.84%	3.05%	4.60%

Hence, a normalized result of an individual instance can be larger than 2.0, if the result is worse than the worst solution regarding this objective of the non-dominated solutions. Furthermore, since the set of all Pareto optimal solutions could not be generated for most instances - even when using a mixed integer program (MIP) - we created reference sets R for the instances by taking all solutions from all runs and methods together and determining the non-dominated solutions. For details on the selected performance indicators, see Schilde et al. (2009).

Range covering This indicator gives the closeness of an approximation set A to the minimum and maximum value for individual objectives. The closer the value to 1.0, the more the solutions reach the extreme points for the objectives.

Hypervolume The hypervolume indicator measures the weakly dominated area by a set A. We generally allow for normalized values above 2.0, so any larger values are set to 2.0, considering that $(2.0, 2.0)$ represents the nadir point with the worst objective values for the non-dominated solutions. Accordingly, the results of the indicator must be within $[0.0, 1.0]$, and a higher values indicate better solutions than lower ones.

Multiplicative unary ϵ This indicator gives the multiplicator ϵ being applied to the objectives of solutions of reference set R such that approximation set A weakly dominates R. The smaller the value of this indicator, the better, with the best value being 1.0.

R3 The R3 indicator calculates, for an approximation set A and a reference set R, the utility values given a set of weight vectors Λ. If the indicator is close to 0.0, set A yields similar utility to R.

$$I_{R3}(A,R) = \frac{\sum_{\lambda \in \Lambda} \frac{u^*(\lambda,R)-u^*(\lambda,A)}{u^*(\lambda,R)}}{|\Lambda|} \tag{3.3}$$

$$u^*(\lambda, A) = \max_{z \in A} u_\lambda(z) \tag{3.4}$$

To calculate utility, we use an augmented Tchebycheff function (3.5) with ρ and z^* fixed to 0.01 and 1.0, respectively.

$$u_\lambda(z) = -(max_{j \in 1,\dots,n} \lambda_j |z_j^* - z_j| + \rho \sum_{j \in 1,\dots,n} |z_j^* - z_j|) \tag{3.5}$$

We assess the proposed multi-objective frameworks on the basis of all performance indicators.

3.5.3 Multi-objective results

We test the proposed multi-objective frameworks with both the MGGDB instances and real-world instances. We decided to only elaborate on the tests for the smaller artificial instances due to the heavy increase in problem complexity from the MCGRP to the MO-P-MCGRP. The real-world instances cover 748 banks and automated teller machines (ATMs) located in the city of Vienna. To determine paths between these locations, we gathered street network data from an open source (Data.gv.at (2019)). All instances can be downloaded from the BDA homepage (bda.univie.ac.at).

Using these data, we created three paths between all banks and ATMs using an algorithm found in Liu et al. (2017) and explained briefly in Section 3.4.2. Note that for the MO-P-MCGRP, including several shortest paths between POIs should increase solution quality, reflecting the second objective of route inconsistency. Since they, however, also cause complexity increase and larger computation times for the heuristics, it is important to limit the number of paths to a reasonable number.

MGGDB and real-world instances do not considerably differ regarding the number of POIs. Paths of the real-world instances, however, include far more edges and arcs. This results in more data handling for the ALNS, but also increases the amount of possible trade-offs between costs and consistency, resulting in larger Pareto sets and more complexity for instances with a similar number of POIs.

Because no Pareto sets were available for the MGGDB and the proposed objectives, we used an approximation. This was created by taking all results from all runs and methods and determining a set of non-dominated solutions. Before doing so, we tried to determine Pareto sets via the MIP-model described in Appendix B. However, the complexity did not allow to find these within time limits of 72 hours for any but two instances.

For parameter tuning, we used Irace (López-Ibáñez et al. (2016)). However, due to rather long runtimes of the multi-objective frameworks, we decided to tune the frameworks' ALNSs by tuning the single-objective ALNS for costs with a time limit of 1 minute per run and applying the results to them.

As test and training sets, real-world instances of different sizes were taken. Considering the amount of parameters, we decided to perform two runs with a budget of 5,000 each. First, a subset of parameters was tuned, then, while keeping the values for the first subset, the second subset was tuned. Table 3.3 lists the tuned parameters, the classification to subsets, their ranges, and the final setting after the tuning.

For EBSH and ECH, the time limit per iteration was set to 5 and 10 minutes, respectively. ECH's higher limit was caused by deeming it more important for ECH than EBSH to find a high-quality solution within each iteration. In case of EBSH, a box and the solution space around it might be explored again later during the run, resulting in the chance to find formerly overlooked solutions. ECH, in contrast, cannot return to a search space above the current ϵ anymore. For MDLS, we applied time limits of 1, 5, 10, 30, 60, 120, and 300 seconds for the ALNS.

Table 3.3: Settings for parameter tuning in the multi-objective case

Parameter	Subset	Range	Final setting
Noise ψ	1	[0; 0.5]	0.2
Capacity penalty ω_{cap}	1	[1; 2000]	1674
Route penalty ω_{route}	1	[1; 5000]	3445
Penalty increase	1	[1; 1.5]	1.49
Penalty decrease	1	[1; 1.5]	1.34
Cooling factor SA	1	[0.5; 0.9999]	0.9755
Threshold for VND μ	1	[1; 1.5]	1.09
Iterations for random destroy & repair ϕ	1	[50; 2500]	933
Score ALNS new best	2	[3; 100]	10
Score ALNS new incumbent	2	[2; 100]	3
Score ALNS new solution	2	[1; 100]	1
Update interval ALNS ν	2	[10; 200]	40

Across all four indicators, the best average was achieved with a time limit of 1 second for the ALNS. The loss in solution quality for larger time limits arises because the algorithm overlooks good solutions, while focusing on a single objective.

However, even with differing time limits for the ALNSs of the individual frameworks, we have equal global runtimes for all three methods.

Results for artificial instances

In Table 3.4, we summarize results for the three basic multi-objective frameworks across the MGGDB instances, with a time limit of 20 hours excluding preprocessing. The table also indicates whether the results are significant according to the Wilcoxon signed-rank test (Wilcoxon (1945)). A 99%-significance is shown by dot ($\bullet$) and prime ($\prime$), whereas a 99.9%-significance is shown by diamond ($\Diamond$) and star ($\star$). A dot ($\bullet$) and diamond ($\Diamond$) indicate a significant improvement compared to the first of the other two entries in the subcolumn for the given indicator. A prime ($\prime$) and star ($\star$) indicate a significant improvement compared to the second of the other two entries in the subcolumn for the given indicator. For example, in Table 3.4 the R3 indicator of -0.109 for the EBSH, marked by a diamond ($\Diamond$) and star ($\prime$), is with a significance of 99.9% different (better) than the first of the other two entries (MDLS) and with a significance of 99% different (better) than the second of the other two entries (ECH).

For three of the four indicators, MDLS is outperformed by ECH and EBSH. However, MDLS shows very good solution quality if we apply the range covering indicator, because ECH and EBSH have problems finding inconsistent solutions. When just comparing ECH and EBSH, EBSH delivers better results for three of the four indicators, though the results in all cases are rather close. Furthermore, the Wilcoxon signed-rank test does not show a significant difference in the range covering and multiplicative unary ϵ indicators. These results for the artificial instances indicate that the ALNS-based EBSH is the method of choice for the newly introduced problem. Except for cases where range covering is of a far greater importance.

Table 3.4: Average results for the basic multi-objective frameworks applied to MGGDB instances. Average indicator values are reported. Column *Objective* indicates whether the indicator aims for high values (↑), low values (↓), or closeness to 0. Best and second best values per indicator are displayed in bold and italic numbers, respectively. $\bullet$, $\Diamond$, $\prime$, $\star$ indicate significance according to the Wilcoxon signed-rank test.

Indicator	Objective	Methods MDLS	ECH	EBSH
Range covering	↑	**0.932**$^{\Diamond\star}$	*0.836*	0.835
Hypervolume	↑	0.217	*0.648*$^{\Diamond}$	**0.650**$^{\Diamond\star}$
Multiplicative unary ϵ	↓	1.528	*1.134*$^{\Diamond}$	**1.132**$^{\Diamond}$
R3	0	-1.034	*-0.113*$^{\Diamond}$	**-0.109**$^{\Diamond\prime}$

Results for real-world instances

Tables 3.5 to 3.7 display the average results for the multi-objective frameworks applied to real-world instances with a time limit of 20 hours, excluding preprocessing. Again, we perform 5 iterations per framework and instance. For the real-world instances, we distinguish between a simple-graph and a multi-graph setting. For the latter, we generate both the two and three shortest paths with maximum similarity of 80% (as explained previously). In Table 3.5 we compare MDLS, ECH, and EBSH against each other, while we compare MDLS and EBSH with EBSH* - a combination of the former two - in Table 3.7. Table 3.6 compares the effects of using a multi-graph for all four frameworks.

Table 3.5: Average results for the multi-objective frameworks applied to the real-world instances, where 10, 20, or 30 POIs must be visited. Average indicator values are reported. Column *Obj.* indicates whether the indicator aims for high values (↑), low values (↓), or closeness to 0. Best and second best values per indicator are displayed in bold and italic numbers, respectively. $\bullet$, $\Diamond$, $\prime$, $\star$ indicate significance according to the Wilcoxon signed-rank test.

			1 path			2 paths			3 paths		
Size	Indicator	Obj.	MDLS	ECH	EBSH	MDLS	ECH	EBSH	MDLS	ECH	EBSH
10	Range cov.	↑	**0.76**$^{\bullet}$	0.67	*0.72*	*0.85*	0.84	**0.89**	*0.9*	0.82	**0.94**
	Hypervol.	↑	0.38	*0.51*	**0.52**$^{\Diamond}$	0.34	*0.6*$^{\Diamond}$	**0.63**$^{\Diamond\star}$	0.26	*0.67*$^{\Diamond}$	**0.69**$^{\Diamond}$
	Multipl. Unary ϵ	↓	*1.33*	1.34	**1.3**	1.33	*1.22*$^{\bullet}$	**1.16**$^{\Diamond\star}$	1.38	*1.12*$^{\Diamond}$	**1.11**$^{\Diamond}$
	R3	0	*-1.06*	-1.15	**-0.93**	-0.85	*-0.36*$^{\Diamond}$	**-0.27**$^{\Diamond\star}$	-0.89	*-0.16*$^{\Diamond}$	**-0.05**$^{\Diamond\star}$
20	Range cov.	↑	**0.78**$^{\Diamond}$	0.62	*0.63*	**0.87**$^{\Diamond}$	0.68	*0.8*$'$	**0.85**$^{\Diamond}$	0.7	*0.75*
	Hypervol.	↑	0.37	*0.57*$^{\Diamond}$	**0.61**$^{\Diamond\star}$	0.32	*0.58*$^{\Diamond}$	**0.63**$^{\Diamond\star}$	0.3	*0.57*$^{\Diamond}$	**0.62**$^{\Diamond\star}$
	Multipl. Unary ϵ	↓	1.34	*1.3*	**1.27**	1.36	*1.27*$^{\Diamond}$	**1.21**$^{\Diamond}$	1.38	*1.26*	**1.22**$^{\Diamond}$
	R3	0	-0.85	*-0.80*	**-0.64**	-0.70	*-0.68*	**-0.44**$'$	-0.77	*-0.67*	**-0.49***
30	Range cov.	↑	**0.82**$^{\Diamond\star}$	*0.63*	0.6	**0.87**$^{\Diamond}$	0.66	*0.78*$'$	**0.85**$^{\Diamond}$	0.65	*0.78*$'$
	Hypervol.	↑	0.28	*0.5*$^{\Diamond}$	**0.58**$^{\Diamond\star}$	0.32	*0.49*$^{\Diamond}$	**0.54**$^{\Diamond}$	0.29	*0.48*$^{\Diamond}$	**0.53**$^{\Diamond\prime}$
	Multipl. Unary ϵ	↓	1.38	*1.33*	**1.27***	1.36	*1.31*	**1.26**$^{\Diamond\bullet}$	1.37	*1.32*	**1.27**$^{\Diamond}$
	R3	0	*-0.88*	-0.95	**-0.64**$^{\Diamond\star}$	**-0.68**$^{\Diamond}$	-0.92	*-0.70**	**-0.69**$^{\Diamond}$	-0.99	*-0.73**

The results for real-world instances keep the trend of ECH and EBSH being better regarding the hypervolume, multiplicative unary ϵ, and, except for larger instances, R3 indicator, while being worse regarding the range covering indicator, except for the smaller instances with multiple paths. This indicates that ECH and EBSH explored far better the middle part of the Pareto frontier yielding far superior results for the hypervolume indicator and still reasonable better results for the multiplicative unary ϵ indicator. At the same time, however, MDLS was better at exploring the outside parts of the Pareto frontier due to having better results for the range covering indicators. This is likely also the cause for the R3 indicator being less in favor of ECH

Table 3.6: Average results for the multi-objective frameworks applied to the real-world instances, where 10, 20, or 30 POIs must be visited. Average indicator values are reported. Column *Obj.* indicates whether the indicator aims for high values (↑), low values (↓), or closeness to 0. Best and second best values per indicator are displayed in bold and italic numbers, respectively. $\bullet$, $\Diamond$, $\prime$, $\star$ indicate significance according to the Wilcoxon signed-rank test.

			10 POIs			20 POIs			30 POIs		
Method	Indicator	Obj.	1 path	2 paths	3 paths	1 path	2 paths	3 paths	1 path	2 paths	3 paths
MDLS	Range cov.	↑	0.76	*0.85*$^{\bullet}$	**0.9**$^{\Diamond}$	0.78	**0.87**$^{\bullet}$	*0.85*$^{\bullet}$	0.82	**0.87**	*0.85*
	Hypervol.	↑	**0.38**	*0.34*	0.26	**0.37**	*0.32*	0.3	0.28	**0.32**	*0.29*
	Multipl. Unary ϵ	↓	**1.33**	*1.33*	1.38	**1.34**	*1.36*	1.38	1.38	**1.36**	*1.37*
	R3	0	-1.06	**-0.85**$^{\bullet}$	*-0.89*	-0.85	**-0.7**	*-0.77*	-0.88	**-0.68**$^{\bullet}$	*-0.69*
ECH	Range cov.	↑	0.67	**0.84**$^{\Diamond}$	*0.82*$^{\Diamond}$	0.62	*0.68*	**0.7**	0.63	**0.66**	*0.65*
	Hypervol.	↑	0.51	*0.6*$^{\Diamond}$	**0.67**$^{\Diamond\prime}$	0.57	**0.58**	*0.57*	**0.5**	*0.49*	0.48
	Multipl. Unary ϵ	↓	1.34	*1.22*$^{\Diamond}$	**1.12**$^{\Diamond\prime}$	1.3	*1.27*	**1.26**	1.33	**1.31**	*1.32*
	R3	0	-1.15	*-0.36*$^{\Diamond}$	**-0.16**$^{\Diamond\star}$	-0.8	*-0.68*	**-0.67**	*-0.95*	**-0.92**	-0.99
EBSH	Range cov.	↑	0.72	*0.89*$^{\Diamond}$	**0.94**$^{\Diamond\star}$	0.63	**0.8**$^{\bullet}$	*0.75*	0.6	*0.78*$^{\Diamond}$	**0.78**$^{\Diamond}$
	Hypervol.	↑	0.52	*0.63*$^{\Diamond}$	**0.69**$^{\Diamond}$	0.61	**0.63**$^{\bullet}$	*0.62*	**0.58***	*0.54*	0.53
	Multipl. Unary ϵ	↓	1.3	*1.16*$^{\Diamond}$	**1.11**$^{\Diamond}$	1.27	**1.21**$^{\Diamond}$	*1.22*	1.27	**1.26**	*1.27*
	R3	0	-0.93	*-0.27*$^{\Diamond}$	**-0.05**$^{\Diamond\prime}$	-0.64	**-0.44**$^{\Diamond}$	*-0.49*	**-0.64**	*-0.7*	-0.73
EBSH*	Range cov.	↑	0.73	*0.92*$^{\Diamond}$	**0.95**$^{\Diamond}$	0.7	*0.83*$^{\bullet}$	**0.93**$^{\Diamond\star}$	0.73	*0.86*$^{\bullet}$	**0.91**$^{\Diamond}$
	Hypervol.	↑	0.52	*0.62*$^{\Diamond}$	**0.65**$^{\Diamond\prime}$	*0.63*	**0.64***	0.61	*0.57*$^{\prime}$	**0.58***	0.53
	Multipl. Unary ϵ	↓	1.29	*1.15*$^{\Diamond}$	**1.13**$^{\Diamond}$	1.21	**1.15**	*1.15*	1.21	**1.15**	*1.19*
	R3	0	-0.91	*-0.25*$^{\Diamond}$	**-0.1**$^{\Diamond\star}$	-0.47	*-0.29*$^{\bullet}$	**-0.28**$^{\bullet}$	-0.47	**-0.33**	*-0.41*

Table 3.7: Average results for the multi-objective frameworks applied to the real-world instances, where 10, 20, or 30 POIs must be visited. Average indicator values are reported. Column *Obj.* indicates whether the indicator aims for high values (↑), low values (↓), or closeness to 0. Best and second best values per indicator are displayed in bold and italic numbers, respectively. $\bullet$, $\Diamond$, $\prime$, $\star$ indicate significance according to the Wilcoxon signed-rank test.

			1 path			2 paths			3 paths		
Size	Indicator	Obj.	MDLS	EBSH	EBSH*	MDLS	EBSH	EBSH*	MDLS	EBSH	EBSH*
10	Range cov.	↑	**0.76**	0.72	*0.73*	0.85	*0.89*	**0.92**	0.9	*0.94*	**0.95**
	Hypervol.	↑	0.38	*0.52*$^{\Diamond}$	**0.52**$^{\Diamond}$	0.34	**0.63**$^{\Diamond}$	*0.62*$^{\Diamond}$	0.26	**0.69**$^{\Diamond}$	*0.65*$^{\Diamond}$
	Multipl. Unary ϵ	↓	1.33	*1.3*	**1.29**	1.33	*1.16*$^{\Diamond}$	**1.15**$^{\Diamond}$	1.38	**1.11**$^{\Diamond}$	*1.13*$^{\Diamond}$
	R3	0	-1.06	*-0.93*	**-0.91**	-0.85	*-0.27*$^{\Diamond}$	**-0.25**$^{\Diamond}$	-0.89	**-0.05**$^{\Diamond}$	*-0.1*$^{\Diamond}$
20	Range cov.	↑	**0.78**	0.63	*0.7*	**0.87**	0.8	*0.83*	*0.85*	0.75	**0.93***
	Hypervol.	↑	0.37	*0.61*$^{\Diamond}$	**0.63**$^{\Diamond}$	0.32	*0.63*$^{\Diamond}$	**0.64**$^{\Diamond}$	0.3	**0.62**$^{\Diamond}$	*0.61*$^{\Diamond}$
	Multipl. Unary ϵ	↓	1.34	*1.27*	**1.21**$^{\Diamond\prime}$	1.36	*1.21*$^{\Diamond}$	**1.15**$^{\Diamond\star}$	1.38	*1.22*$^{\Diamond}$	**1.15**$^{\Diamond\star}$
	R3	0	-0.85	*-0.64*	**-0.47**$^{\Diamond\star}$	-0.7	*-0.44*	**-0.29**$^{\Diamond\star}$	-0.77	*-0.49*	**-0.28**$^{\Diamond\star}$
30	Range cov.	↑	**0.82**$^{\Diamond}$	0.6	*0.73*$^{\prime}$	**0.87**	0.78	*0.86**	*0.85*	0.78	**0.91***
	Hypervol.	↑	0.28	**0.58**$^{\Diamond}$	*0.57*$^{\Diamond}$	0.32	*0.54*$^{\Diamond}$	**0.58**$^{\Diamond}$	0.29	*0.53*$^{\Diamond}$	**0.53**$^{\Diamond}$
	Multipl. Unary ϵ	↓	1.38	*1.27*	**1.21**$^{\Diamond\prime}$	1.36	*1.26*$^{\Diamond}$	**1.15**$^{\Diamond\star}$	1.37	*1.27*$^{\Diamond}$	**1.19**$^{\Diamond}$
	R3	0	-0.88	*-0.64*$^{\Diamond}$	**-0.47**$^{\Diamond\star}$	*-0.68*	-0.7	**-0.33**$^{\Diamond\star}$	*-0.69*	-0.73	**-0.41**$^{\Diamond\star}$

and EBSH, once they do not achieve competitive results for the range covering indicator, due to reaching a high utility, when inconsistency is valued highly.

The outliers of MDLS not being the best for the range covering indicator when looking at the smallest instances and at the same time using the multi-graph can be explained by ECH and EBSH being more capable of exploiting the benefits of the multi-graph for the still simple, small instances due to the individual ALNS-runs of ECH and EBSH having far longer runtime than MDLS's.

When looking beyond the small instances, it can be examined that the multi-graph approach still increases the results for the range covering indicator to a certain degree. This is not too surprising due to the inherent, before already explained limit of the conversion and the inclusion of multiple paths allowing for more inconsistent solutions. When comparing the ECH and EBSH, EBSH improved far more for the medium and large instance, while being roughly equal for smaller ones. We explain this by the problem still being very simple for the smaller ones enabling the underlying ALNS for ECH and EBSH to converge against the same value within the given time limit, while not being able to do so for medium and larger ones. Our implementation of the ECH ends as soon as no new solution is found during an iteration, whereas the EBSH would continue with trying to split the largest remaining box. Therefore, our EBSH, when given a lot of computation time, often used more computation time in total allowing for more exploration. This furthermore explains the trend of EBSH being slightly better than ECH with the Wilcoxon signed-rank test still showing high significance even for larger instances. There is almost no difference and especially no high significance between using two or three paths. We explain this by the fact that there is not enough potential consistency increase, when adding the third path. However, since MDLS still had far better results compared to ECH and EBSH, these latter two methods still miss potential improvements. This implies that the additional complexity of the third path negated the potential gains.

The effect on the hypervolume, multiplicative unary ϵ, and R3 indicators seems even more mixed. When looking at ECH and EBSH, having multiple paths significantly improves the results for the smaller instances. Medium sizes ones, however, are only slightly improved and the Wilcoxon signed-rank test does not show such a high significance anymore, while larger instances are even partially worsened; with even a significance of 99.9% for one specific comparison. This confirms that small instances can easily be exploited even with the increased solution space, while the additional complexity of more paths destroys the benefit of the Pareto set of the multi-graph dominating the Pareto set of the simple graph. That this negative effect is more visible for the hypervolume, multiplicative unary ϵ likely stems from them being more complex indicators than range covering.

As the results indicate that the ϵ-based methods and MDLS have different strengths, it might be reasonable to combine these methods. When comparing the results of MDLS and EBSH (since EBSH and ECH are very similar in their design, while EBSH performing better), we observe the following: almost all solutions of MDLS that are not of lower costs than EBSH's solution or less consistent than EBSH's least consistent solution are dominated by one or more solutions of EBSH. Secondly, there are on average very few solutions that are still more cost-efficient than EBSH's (0.36) with an increasing but still small average for the larger instances (0.83). In comparison, the number of less consistent solutions is rather large with 2.16 and 3.80 on average for all instances and the larger instances, respectively. From these observations we conclude that the most promising combined method (denoted as EBSH*), is to (i) start with EBSH followed by (ii) short runs of MDLS's consistency-ALNS with the least consistent solutions found as potential starting solutions. While there might also be more cost-efficient solutions remaining, we consider it unlikely to find them within a short amount of time and only short ALNS runs. Instead it seems preferable to increase the time for generating the initial solution.

For the combined approach (EBSH*), we assign 19 hours of runtime for EBSH and 1 hour for MDLS's consistency-ALNS. Hence, the total runtime is equal compared to the experiments of the individual approaches. Table 3.7 compares EBSH* to MDLS and EBSH. For the smallest instances, the results of EBSH and EBSH* are almost the same. Thus, the added search for inconsistent solutions does not yield any large improvements. However, for these instances we did not observe a large gap regarding range for any of the methods, anyway. For larger instances, no significant changes are apparent when comparing the hypervolume indiciator of EBSH and EBSH*, whereas significant improvements are revealed for the other three indicators. This seems to stem from the fact that they are more influenced by a few less consistent solutions. However, even for the larger instances when using one or two paths, EBSH* is on average still not significantly better than MDLS for the range covering indicator. We assume that the significance when having three paths is due to the closeness towards the maximum value for the objectives of EBSH*'s approximation sets, since MDLS's approximation sets are still closer towards the minimum value. Therefore, MDLS is still better at finding the extreme solutions for the single-objective problems, with EBSH* closing the gap considerably.

3.5.4 Trade-off between cost and inconsistency

Table 3.8 shows the average improvement in inconsistency (i.e., decrease in consistency) if costs increase by a certain percentage p, relative to the cheapest solution $s_{cheapest}$ for an instance, which can be denoted as

$$\min_{s\in S}\Big\{\frac{s.cons - s_{cheapest}.cons}{s_{cheapest}.cons}|s.cost \leq s_{cheapest}.cost \times (1+p)\Big\}. \tag{3.6}$$

We only consider the non-dominated solutions of the 13 real-world instances. The first column shows the percentage increase, and of the lowest cost, while the remaining columns show the average reduction in consistency for the versions with 1 to 3 paths.

We observe that even a very small cost increase of 1% can lead to a considerable decrease in consistency, of at least 10%. Further cost increases lead to additional consistency decreases, though these trade-offs are weaker. Furthermore, the trade-offs are more beneficial with multiple paths, and the maximum consistency reduction is reached with three paths.

Table 3.8: Trade-off between cost increase and consistency decrease.

	Consistency decrease		
Cost inc.	1 path	2 paths	3 paths
1%	-10.58%	-13.18%	-13.69%
2%	-16.08%	-17.80%	-18.39%
5%	-21.93%	-24.03%	-25.22%
10%	-25.32%	-28.56%	-29.81%
No limit	-25.49%	-29.60%	-31.31%

3.6 Conclusion

We introduced the MO-P-MCGRP, which stems from vehicle routing problems for security guards and cash-in-transit operations. Objects and streets must be serviced or traversed such that cost and route inconsistency over multiple periods are minimized and maximized, respectively. We have formulated the MO-P-MCGRP mathematically and transformed it to an asymmetric CVRP. An ALNS-based search procedure, developed and benchmarked on single-objective MCGRP instances, was embedded into three multi-objective solution frameworks (MDLS, ECH, and EBSH) to generate solutions for the MO-P-MCGRP. We assessed the performance of these frameworks with four indicators taken from literature. For almost all instances, we found MDLS to be the best approach regarding the range covering indicator, while the ϵ-based methods were on average better for the other indicators. When comparing these two methods, we can additionally examine EBSH being significantly better than ECH in many cases, without cases being the other way round. A combined approach of iteratively applying MDLS's consistency-ALNS within the EBSH framework, did not show an effect on small instances, but slightly improved results for the larger instances. Furthermore, we observed that considering more than one shortest path between POIs might significantly increase solution quality. However, the additional computational effort of handling multi-graphs for larger instances caused them to have less benefits or even worse results, hinting strongly at a simple-graph being the method of choice for even larger or more complex instances. The trade-off between cost and inconsistency revealed that the desired inconsistency increases significantly with just a small cost increase, and these trade-offs grow even more beneficial when a multi-graph is used. All real-world instances are made publicly available in order to stimulate more research in this challenging and practically relevant area.

CHAPTER 4

A rolling horizon framework for the time-dependent multi-visit dynamic safe street snow plowing problem

Published in: *Networks*
A rolling horizon framework for the time-dependent multi-visit dynamic safe street snow plowing problem.
Georg E. A. Fröhlich, Margaretha Gansterer, Karl F. Doerner
https://doi.org/10.1002/net.22189

Abstract As a major real-world problem, snow plowing has been studied extensively. However, most studies focus on deterministic settings with little urgency and enough time to plan. In contrast, we assume a severe snowstorm with little known data and little time to plan. We introduce a novel time-dependent multi-visit dynamic safe street snow plowing problem and formulate it on a rolling-horizon-basis. To solve this problem, we develop an adaptive large neighborhood search as the underlying method and validate its efficacy on team orienteering arc routing problem benchmark instances. We create real-world-based instances for the city of Vienna and examine the effect of (i) different snowstorm movements, (ii) having perfect information, and (iii) different information-updating intervals and look-aheads for the rolling horizon method. Our findings show that different snowstorm movements have no significant effect on the choice of rolling horizon settings. They also indicate that (i) larger updating intervals are beneficial, if prediction errors are low, and (ii) larger look-aheads are better suited for larger updating intervals and vice versa. However, we observe that less look-ahead is needed when prediction errors are low.

4.1 Introduction

Snow plowing is a major real-world problem. It affects street safety, travel times, and traffic jams. Furthermore, some regions suffer considerable costs for efficient plowing and often go over budget, e.g., total costs of \$ 192 million in Montreal in 2018 (Scott (2019)). To deal with such problems, snow plowing has been studied extensively in the literature. Most studies, however, consider deterministic problems where arcs do not need repeated plowing and priorities between sets of arcs do exist. These models are good representations of situations after typical snowstorms with no major impact. In our opinion, however, these models are not too well suited for severe snowstorms that threaten to shut down the whole infrastructure and that require immediate care. Such a scenario was considered by Dussault et al. (2013). The snowstorm of their model warranted a shutdown and imposed impassable streets, even for a snow plowing truck. A problem as in the aforementioned paper could be evaded or mitigated, should effective plowing be performed before the accumulation of masses of snow.

We want to address this research gap by modeling a novel time-dependent multi-visit dynamic safe street snow plowing problem called the time-dependent multi-visit dynamic team orienteering arc routing problem (TDMVD-TOARP), which aims to keep the streets as safe as possible. The roads are deemed safe if the accumulated snow is below a specified threshold. Different priorities are assigned to streets according to classical characteristics (e.g., streets leading to critical infrastructure, streets located in heavily populated areas). In doing so, however, we do not create classes of priorities (e.g., Perrier et al. (2008)), where higher ones must be serviced before the lower ones; instead, we transform them into weights and include them in the objective function. Further, fleet costs are not considered, as the fleet is assumed to be available anyways, and we want to maximize the efficacy of the available resources. Snowfall intensity is not assumed to be constant or known ahead of time. This can result in some streets not requiring any service, while others require multiple plowings to be deemed safe over the entire planning horizon. These characteristics lead to a time-dependent problem wherein an earlier plowing can turn detrimental (see Section 4.3.5). A time-discretized mixed integer program (MIP) would be very challenging to solve, even if snowstorm movement is predictable; therefore, we address the problem on a rolling planning horizon. The proposed framework splits the problem into smaller problems and predicts the future snowfall and the state of the system before solving each subproblem. The subproblems are, hence, considered to be deterministic and solutions are generated using adaptive large neighborhood search (ALNS). Due to considering a real-world problem with dynamic changing data, the rolling horizon framework has only a very limited computation time to calculate a solution for a given subproblem, therefore a metaheuristic solution technique is used. During this time the solution for the previous subproblem is executed (e.g., the solution for the subproblem starting at minute 31 is determined during minutes 30 to 31; then, for minute 32, the solution will be determined during minutes 31 to 32). In the following, this available computation time will be called *updating interval*. In order to not be too myopic, subproblems might plan a little ahead of time, even if the latter part of the solution will not be used (e.g., the subproblem lasts from minute 31 to 32.5). This additional time (e.g., 0.5 minutes) will be called *look-ahead*.

The contribution of our study is fourfold:

1. We introduce a novel and challenging dynamic safe street snow plowing problem, model it as a TDMVD-TOARP, and tackle it in a rolling horizon framework.

2. To generate solutions, we propose an effective ALNS-based method with tailored destroy and repair operators.

3. In an extensive computational study, we examine the effects of different snowstorm movements, different levels of information, and forecasting quality.

4. We perform an in-depth analysis of the most important parameters for the rolling horizon frameworks (e.g., updating interval, look-ahead).

The remaining paper is structured as follows: Section 4.2 provides an overview of the related literature. Section 4.3 defines the problem and describes the transformation to the rolling horizon setting. Section 4.4 explains the proposed solution approach, while Section 4.5 presents the computational study. There, we first show the efficacy of the algorithm and then apply it to real-world-based instances. Finally, Section 4.6 includes concluding remarks and future research.

4.2 Literature Review

An extensive overview of the characteristics and problems related to snowfall can be found in the four-part review by Perrier et al. (2006a,b, 2007a,b). The authors classify the problems on strategic (e.g., facility location, fleet replacement scheduling), tactical (e.g., sector assignment to snow disposal sites, fleet size), and operational (e.g., vehicle routing and crew management) levels and extend them to real-time problems. The snow plowing problem is closely related to anti-icing, deicing, and abrasive spreading operations. All these problems typically belong to the operational or real-time class and are often closely related to the Chinese postman problem, rural postman problem, or capacitated arc routing problem. Commonly added characteristics are priorities or hierarchies for subsets of streets, turn penalties, variable speeds, and heterogeneous fleets. Priorities are deemed necessary as some streets (e.g., those leading to a hospital) are more important than others. Priorities can be included by weights or the requirement to serve arcs with a higher priority before those with lower ones. Turn penalties can be due to (i) street conditions leading to risky turns, or (ii) turns leading to snow accumulation on crossings. Within the models, these penalties can be tackled by forbidding very risky turns, adding time for turns, or adding penalties in the objective functions. Variable speeds are needed to carry out deadheading, which is typically faster than plowing. Regarding heterogeneous fleets, authors typically assume that companies use vehicles of different sizes (e.g., smaller ones for alleys), and vehicles for plowing or spreading.

Perrier et al. (2008) study a hierarchical rural postman problem with lexicographic objectives; the problem minimizes the makespan for each subset of streets of the same priority and includes heterogeneous vehicles, turn restrictions, and variable speeds. Different decompositions are applied. Salazar-Aguilar et al. (2012) focus on a synchronized arc routing problem with the aim to minimize the makespan. They require synchronization for streets that have multiple lanes in the same direction. This is due to the fact that a single plow would only create a large amount of snow on the neighboring lane. This problem is tackled by using a synchronized convoy of plows, with each plow pushing the snow further to the edge. They tackle it by solving a relaxed MIP and then improving it via ALNS. Dussault et al. (2013), in their windy postman problem, forbid the deadheading of non-plowed arcs, as their scenario assumes a snowstorm severe enough to warrant a shutdown. Their objective is cost minimization and they tackle the problem using a heuristic, which delivers results close to the lower bounds. Hajibabai and Ouyang (2016) study a stochastic snow plow fleet management problem with a weighted sum of deadheading and repositioning costs and rewards depending on service levels. The authors split the problem into multiple periods. They assign vehicles to snow routes such that the tasks can be served within one period. In each period, a truck can either (i) serve tasks in its current snow route, (ii) move to a different snow route, or (iii) remain idle. The tasks can arise randomly (following a Poisson process), and vehicles can fail randomly. In case of vehicle failure, the tasks that should have been serviced are added to the next period. To generate solutions, the authors propose adaptive dynamic programming, which is used to evaluate the effect of taking uncertainty into account. Quirion-Blais et al. (2017b) study a hierarchical plowing and spreading problem with heterogeneous fleets, variable speeds, and turn penalties. They use ALNS to solve it for varying fleet compositions and deduce bottlenecks. Quirion-Blais et al. (2017a) focus on a hierarchical rural postman problem with turn penalties, variable speeds, and heterogeneous vehicles with the objective to minimize a weighted sum of the makespan for each subset of streets of the same priority. The problem is transformed into an asymmetric traveling salesman problem and is solved using ALNS.

Holmberg (2019) investigates a snow plowing problem for a single snow remover, where streets need to be plowed up to four times. The reasoning is that some cities with broader streets require additional plowing between the lanes, as plows are too narrow. While some types of plowing need to be performed in a specific direction, others can be performed in either direction. The problem is reformulated into an asymmetric traveling salesman problem and solved using a heuristic. Hajizadeh and Holmberg (2020) extend this study by creating a hybrid heuristic and a MIP relaxation. The authors use branching techniques in both methods, which allows sharing of information between the hybrid heuristic and the MIP relaxation. Castro Campos et al. (2020) prove that the problem in Dussault et al. (2013) can be solved in polynomial time when costs are symmetric or metric, i.e., uphill service costs are, at most, equal to downhill service costs and two uphill deadheadings. This assumption applies to several real-world problems. Additionally, the authors also create heuristics for this problem, which make use of several of their theoretical findings.

Xu and Kwon (2021) address service levels based on users' driving times. The objective is to minimize the weighted sum of travel times and the makespan of the snow plowing problem. Travel times are time-dependent, as cleared streets can be traversed faster. Accordingly, the following two user behaviors are tested in this problem: (i) users follow their usual path and (ii) users adapt to street conditions and make use of alternative street segments. The authors propose to tackle the problem using tabu search. Ahabchane et al. (2021) address a hierarchical salt spreading model, where priorities are included in the objective function. The latter consists of the total routing costs and the sum product of weights given according to the priorities and the times they are serviced. While the original problem is a mixed capacitated general routing problem, the authors transform it into a node routing problem. Furthermore, the demands are not deterministic, which they tackle using a robust formulation by accounting for demand variation. The authors create a heuristic based on simulated annealing (SA) and evaluate the performance using Monte Carlo simulations.

Through this work, we extend the research on dynamic *real-time* snowplowing problems, where complexity and, particularly, the underlying time-dependencies require a new solution approach. We propose to tackle the problem within a rolling horizon framework and evaluate the updating intervals as well as look-aheads based on prediction accuracy.

4.3 Problem Formulation

Our problem formulation is split into multiple parts. First, we define the main problem of keeping the streets as safe as possible. However, as it is too large to be solved and, for the rolling framework, is later decomposed into smaller problems, a formal definition for these subproblems is given. Due to the subproblems' limited time frame lengthy arcs might pose problems and require special consideration, explained in Section 4.3.3. Afterwards the information handling - what information is available, what is only assumed - for the rolling horizon framework is explained. Finally, a section is dedicated to the calculation of changes to the solution.

4.3.1 Integrative model

The TDMVD-TOARP can be represented on a graph $G = (V, A)$, where V is the set of vertices and A is the set of arcs in which each arc a is of different priority w_a. Arcs with high w_a typically include locations that are (i) critical infrastructure (e.g., hospitals), (ii) situated within highly populated areas, or (iii) considered arterial roads. The time horizon is denoted as $\mathcal{T}$. During $\mathcal{T}$, a snowstorm passes through the system (the city), causing snow to accumulate, while a fleet of vehicles K, which starts at a depot, clears the streets and tries to keep them as safe as possible.

When an arc a is affected by the snowstorm at time t, its level of snow l_{at} increases with the storm's intensity i_t ($l_{at} = l_{a,t-1} + i_t$). Should an arc a be serviced by a vehicle at time t, the level is reset to 0 ($l_{at} = 0$). Whenever an arc's level of snow is below a threshold H, it is deemed safe ($s_{at} = 1$); otherwise, it is deemed not safe ($s_{at} = 0$).

The objective is to maximize the weighted time streets are safe. Here, cost minimization is not an objective, as the fleet of vehicles is considered to be available.

To depict the problem as a MIP, each arc is represented twice: once in set $A_S = \{a_1^s, ..., a_n^s\}$ and once in set $A_T = \{a_1^t, ..., a_n^t\}$, where every arc corresponds to (i) traversing and servicing or (ii) only traversing the arc, respectively. The duration needed for each arc is denoted by d_a. Traversal and service of an arc takes longer than just traversal ($d_{a_i^s} > d_{a_i^t}$). The starting and ending nodes for an arc are given by σ_a and ω_a. The decision variable x_{atk} equals 1 if an arc a's traversal (and possibly service) starts at time t by vehicle k. Similarly, p_{at} equals 1 if arc a's service starts at time t.

$$max \sum_{a \in A_S} \sum_{t \in \mathcal{T}} s_{at} w_a \tag{4.1}$$

$$\sum_{a \in A_S \cup A_T | \sigma_a \in D} x_{a,0,k} = 1 \qquad \forall k \in K \tag{4.2}$$

Table 4.1: Decision variables and constants

Sets	
A_S	set of arcs, where traversal includes service
A_T	set of arcs, where only traversal occurs
D	set of nodes, where depots are located
K	set of vehicles
T	set of discretized time units
Parameters	
w_a	weight (priority) of arc a
d_a	duration for arc a (depending on the arc it can either be service and traversal, if $a \in A_S$, or only traversal, if $a \in A_T$)
i_{at}	intensity of snowfall on arc a at time t
σ_a	start node of arc a
ω_a	end node of arc a
e_k	node from which vehicle k starts
t_k^s	time at which vehicle k starts
H	threshold of snow that has to be met
M	large number
Decision variables	
x_{atk}	equals 1 if arc a is traversed by vehicle k starting at time t; otherwise, 0
s_{at}	equals 1, if arc a is unsafe at time t; otherwise, 0
p_{at}	equals 1, if arc a is serviced starting at time t; otherwise, 0
l_{at}	level of snow on arc a at time t

$$\sum_{a \in A_S \cup A_T} x_{atk} \leq 1 \qquad \forall t \in T \quad \forall k \in K \tag{4.3}$$

$$\omega_a x_{atk} = \sum_{j \in A_S \cup A_T} x_{j,t+d_a} \sigma_j \qquad \forall a \in A_S \cup A_T \quad \forall t \in T \quad \forall k \in K \tag{4.4}$$

$$(1 - x_{atk})M \geq \sum_{j=t+1}^{t+d_a-1} x_{ajk} \qquad \forall a \in A_S \cup A_T \quad \forall t \in T \quad \forall k \in K \tag{4.5}$$

$$p_{at} = \sum_{k \in K} x_{atk} \qquad \forall a \in A_S \quad \forall t \in T \tag{4.6}$$

$$l_{at} \geq l_{a,t-1} + i_{at} - p_{at} M \qquad \forall a \in A_S \quad \forall t \in T \tag{4.7}$$

$$l_{at} \geq 0 \qquad \forall a \in A_S \quad \forall t \in T \tag{4.8}$$

$$l_{at} < H + (1 - s_{at})M \qquad \forall a \in A_S \quad \forall t \in T \tag{4.9}$$

$$x_{atk} \in \{0, 1\} \qquad \forall a \in A_S \cup A_T \quad \forall t \in T \quad \forall k \in K \tag{4.10}$$

$$s_{at} \in \{0, 1\} \qquad \forall a \in A_S \quad \forall t \in T \tag{4.11}$$

Equation (4.1) defines the objective function, while Constraints (4.2) force every vehicle to traverse an arc connected to a depot at time 0. Returning to the depot, however, is not required. At any time, a vehicle may only start the traversal of a single arc due to Constraints (4.3). Constraints (4.4) enforce vehicles to never be idle and to choose a connecting arc. If arc a's traversal started at time t, and therefore, finishes its traversal at $t + d_a$, a new traversal must start on an arc i, right where a ended. Constraints (4.5) guarantee that no additional traversals take place during this time; with $d_a - 1$ being sufficiently large big-Ms. The service of an arc is determined by Constraints (4.6), while the levels of snow are calculated using Constraints (4.7) and (4.8). Constraints (4.9) define the safety of the arcs. The largest value needed for the big-Ms in Constraints (4.7) for a specific arc a and time t is $M = l_{a0} + \sum_{n=0}^{t} i_{an}$, as it enables clearing the largest amount of snow that would be possible on arc a at time t. Similarly, for Constraints (4.9) with arc a and time t, a big-M of $M > l_{a0} + \sum_{n=0}^{t} i_{an} - H$ is sufficient. In addition, when only using a single big-M, the smallest sufficient value for both types of constraints would be $M = max_{a \in A_S}\{l_{a0} + \sum_{t \in T} i_{at}\}$, which is the largest amount of snow that can accumulate on a single arc if no plowing is performed.

4.3.2 Rolling horizon adaptation

If the problem is embedded within a rolling horizon framework, the following conditions are required:

- Vehicles do not necessarily start at the depot, but at the node they stopped previously (e_k).

- Instead of the time frame $\mathcal{T}$, a reduced time frame T_i' is considered. This reduced time frame consists of the updating interval T_i and the look-ahead E_i. All vehicles must end their movement within T_i'.

- Due to the look-ahead component, vehicles may be forced to first finish a previously started movement and, consequently, start later (t_k^s). For example, a vehicle started with arc a in the previous subproblem during T_i but finished its movement in E_i.

These adaptions can be represented in the MIP by replacing Constraints (4.2)–(4.5) and Constraints (4.10) with the following constraints. All other constraints would replace the set T with T_i', as a reduced time frame T_i' would be used instead of the whole time frame.

$$\sum_{a \in A_S \cup A_T | \sigma_a = e_k} x_{a,t_k^s,k} = 1 \qquad \forall k \in K \tag{4.12}$$

$$\sum_{a \in A_S \cup A_T} x_{atk} \leq 1 \qquad \forall t \in T_i' | t \geq t_k^s \quad \forall k \in K \tag{4.13}$$

$$\omega_a x_{atk} = \sum_{j \in A_S \cup A_T} x_{j,t+d_a,k} \sigma_j \qquad \forall a \in A_S \cup A_T \quad \forall t \in T_i' | t \geq t_k^s \quad \forall k \in K \tag{4.14}$$

$$(1 - x_{atk})M \geq \sum_{j=t+1}^{t+d_a-1} x_{ajk} \qquad \forall a \in A_S \cup A_T \quad \forall t \in T_i'|t \geq t_k^s \quad \forall k \in K \tag{4.15}$$

$$x_{atk} = 0 \qquad \forall a \in A_S \cup A_T \quad \forall t \in T_i'|t < t_k^s \quad \forall k \in K \tag{4.16}$$

$$x_{atk} \in \{0, 1\} \qquad \forall a \in A_S \cup A_T \quad \forall t \in T_i'|t \geq t_k^s \quad \forall k \in K \tag{4.17}$$

Constraints (4.12) and (4.16) handle the vehicle finishing any previous movement and starting after the arc of the previous movement. Constraints (4.13)-(4.15) and (4.17) are adapted to start after the previous movement is done ($\forall t \in T_i'|t \geq t_k^s$). The calculations of the snow level and penalties (cf. Constraints (4.6)-(4.9) and (4.11)) would not be influenced by the previous movement ($\forall t \in T_i'$).

The following stopping criteria might be applied for the rolling horizon optimization: (i) all streets are safe for a sufficient period of time, and (ii) the snowstorm has ended. For the first case, no additional steps are required. Vehicles would return to the depot(s); but due to costs not being considered in the model, the details of the return would not matter. For the second case, however, unsafe arcs must be plowed. As every arc requires at most one plowing operation during this phase, a different problem formulation would be required for the end of the horizon phase. Nevertheless, the rolling horizon framework could be used until all streets are cleared.

4.3.3 Splitting of arcs

Some arcs cannot be traversed, as their traversal or service time is larger than T_i', whereas other arcs may be so long that they cannot be combined with any other arc. E.g., consider three arcs with service durations of 1.5 times, 0.9 times, and 0.4 times the time frame T_i' (cf., the arcs on the left graph of Figure 4.1). The arc with a duration of $1.5T_i'$ clearly cannot be serviced within the time frame T_i'. The other two arcs could be serviced, but only individually. To enable the servicing of every arc in this problem and enable more combinations, they are remodeled into several smaller arcs, each within $T_i'/2$. Assuming an arc a has a traversal time t, the number n of required segments can be calculated by $n = \lceil 2t/T_i' \rceil$. The arc is then split into n arcs of equal length. For all arcs except the last one, the only successor is the directly following arc ($Succ(a_i) = \{a_{i+1}\} \forall i < n$). Note that within the updating framework, either all arcs $a_1, ..., a_n$ must be serviced consecutively or not serviced at all (e.g., service of a_2, but not a_1, is prohibited). Figure 4.1 provides an example with one arc of each type. While the arcs representing the arc that previously had a duration of $1.5T_i'$ still could not be serviced within T_i', the arc can now be serviced over two periods (e.g., first $t_{1,1}$ and $t_{1,2}$, and in the next period $t_{1,3}$). Similarly, the service of the arc with a duration $0.9T_i'$ could follow the arc with a duration $0.4T_i'$, by starting with the first segment in the first period and finishing the second segment in the second period.

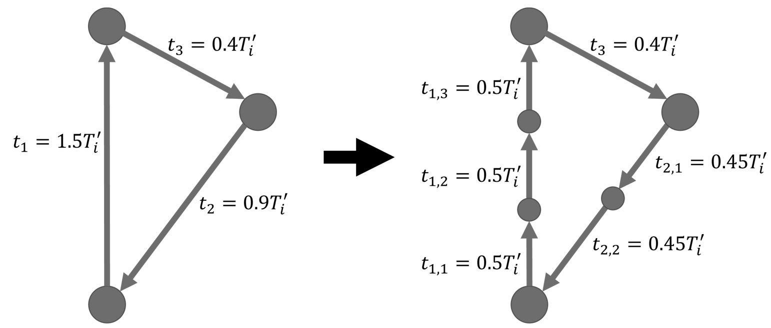


Figure 4.1: Illustration of the splitting of arcs. On the left side, t_1 could not be scheduled at all, while t_2 could not be combined with any other arc. On the right side, all arcs are split into smaller arcs with times of at most $T_i'/2$.

4.3.4 Information handling for the rolling horizon

Figure 4.2 and Table 4.3 illustrate the sequencing within the rolling horizon framework. During each updating interval T_i, while the solution Sol_{T_i} for the current period is executed (see Figure 4.2), solution $Sol_{T_{i+1}}$ for the next period is determined. Available data includes the current state of the system S_{T_i}, the solution Sol_{T_i} for the current period, and the previous snowstorm movements $M_{T_0}, ..., M_{T_{i-1}}$. In all, four steps are performed. First, the previous snowstorm movements are used to predict the upcoming snowstorm movements $\widehat{M}_{T_i}$ and $\widehat{M}_{T_{i+1}}$ (see step 1 in Table 4.3). Next, the current state and the solution for the current period are combined with the predicted snowstorm movements for the current period, resulting in the predicted starting state for the next period $\widehat{S}_{T_{i+1}}$ (see step 2 in Table 4.3). Based on the predicted starting state for the next period and the predicted snowstorm movements, the solution $Sol_{T_{i+1}}$ for the next time period is determined (see step 3 in Table 4.3). Finally, time passes to the next period and the real state $S_{T_{i+1}}$ and snowstorm movement M_{T_i} become known (see step 4 in Table 4.3).

Three criteria are used for predicting the snowstorm: angular speed, acceleration, and intensity change. After every iteration of the rolling horizon, the true movement of the snowstorm during the period is read and saved for the three criteria. Individual entries correspond to one second. Furthermore, entries that are more than $1,000$ seconds in the past are deleted, as their likelihood of still being correlated to the current diminishes. Based on the remaining entries, the expected values are calculated by moving averages for the three criteria. They are further used for the angular speed, acceleration, and intensity change of the predicted snowstorm.

Table 4.2 depicts a small example of such a prediction. Given are the measured centers of the snowstorm and snowstorm intensities over the first 4 time units. From them, the speed, angle, acceleration, angular speed, and intensity change are determined. The expected acceleration, angular speed, and intensity change are calculated by moving average (e.g., $(0.414+0) \div 2 = 0.207$ for the acceleration), and are then used to calculate the predicted values via linear equations (e.g., $45° + 22.5° \cdot 1 = 67.5°$). For calculating a snowstorm's center (x, y) based on speed s and angle g, circle equations are used; $(x, y) = (x_{old} + s\sin(g), y_{old} + s\cos(g))$. Arcs within a limited range of the predicted snowstorm centers (e.g., at most one kilometer away) are predicted to accumulate snow according to the intensities.

Table 4.2: Example for predicting the snowstorm.

	Measured values				Predicted values	
Time	1	2	3	4	5	6
Snowstorm's center	(0, 0)	(0, 1)	(1, 2)	(2, 3)	(3.5, 3.62)	(5.33, 3.62)
Speed	-	1	1.414	1.414	1.621	1.828
Angle	-	0°	45°	45°	67.5°	90°
Intensity	-	10	8	9	8.5	8
Acceleration	-	-	0.414	0	0.207	0.207
Angular speed	-	-	45°	0°	22.5°	22.5°
Intensity change	-	-	-2	1	-0.5	-0.5

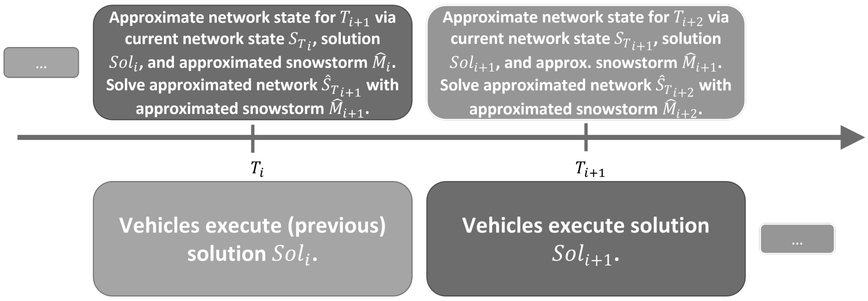


Figure 4.2: Updating process within the rolling horizon framework. The previous solution Sol_i is executed at the start of interval T_i and until the start of T_{i+1}. The next solution is determined with all available information.

Table 4.3: Sequences of information becoming available within the rolling horizon framework.

Time	Step	Known	Predicted	Unknown	Step
T_{i-1}	...	...	...	...	...
T_i	1	S_{T_i}, Sol_{T_i}, M_{T_0}, ..., $M_{T_{i-1}}$	-	$Sol_{T_{i+1}}$, $S_{T_{i+1}}$, M_{T_i}, $M_{T_{i+1}}$	$predictMovement(M_{T_0}, ..., M_{T_{i-1}}) \rightarrow \{\widehat{M}_{T_i} \widehat{M}_{T_{i+1}}\}$
T_i	2	S_{T_i}, Sol_{T_i}, M_{T_0}, ..., $M_{T_{i-1}}$	$\widehat{M}_{T_i}$, $\widehat{M}_{T_{i+1}}$	$Sol_{T_{i+1}}$, $S_{T_{i+1}}$	$predictState(S_{T_i}, Sol_{T_i}, \widehat{M}_{T_i}) \rightarrow \widehat{S}_{T_{i+1}}$
T_i	3	S_{T_i}, Sol_{T_i}, M_{T_0}, ..., $M_{T_{i-1}}$	$\widehat{S}_{T_{i+1}}$, $\widehat{M}_{T_i}$, $\widehat{M}_{T_{i+1}}$	$Sol_{T_{i+1}}$	$performALNS(\widehat{S}_{T_{i+1}}, \widehat{M}_{T_{i+1}}) \rightarrow Sol_{T_{i+1}}$
T_i	4	S_{T_i}, Sol_{T_i}, M_{T_0}, ..., $M_{T_{i-1}}$, $Sol_{T_{i+1}}$	$\widehat{S}_{T_{i+1}}$, $\widehat{M}_{T_i}$, $\widehat{M}_{T_{i+1}}$	-	$timePasses() \rightarrow \{S_{T_{i+1}}, M_{T_i}\}$
T_{i+1}	1	$S_{T_{i+1}}$, $Sol_{T_{i+1}}$, M_{T_0}, ..., M_{T_i}	-	$Sol_{T_{i+2}}$, $S_{T_{i+2}}$, $M_{T_{i+1}}$, $M_{T_{i+2}}$	$predictMovement(M_{T_0}, ..., M_{T_i}) \rightarrow \{\widehat{M}_{T_{i+1}} \widehat{M}_{T_{i+2}}\}$
T_{i+1}	...	...	...	...	...

4.3.5 Calculating changes in safety

While the calculation of weighted safety is not difficult, it can take some time should a straightforward approach be used. For every arc and every discretized time unit, a check whether the accumulated snow is too much must be performed. Should the solution be changed by inserting or removing sequences of arcs, any later parts of the solution would change as well due to being plowed later/earlier resulting in more/less snow than before. A straightforward, but slow approach would again be simply making a conditional check for every affected arc.

However, by preprocessing the benefits b_{at} of plowing an arc a at time t these calculations can be sped up significantly. Should multiple plowings of an arc be allowed during time frame T_i', such precalculations would be problematic, as the plowings would influence one another. E.g., assume an unsafe arc, for which snow accumulates at a rate that it is unsafe again after 5 minutes. Individually, plowing it at $t = 3$ would keep it safe for 5 minutes, and plowing it at $t = 5$ would keep it safe for 5 minutes. When plowing it at $t = 3$ and at $t = 5$ it would be safe from $t = 3$ to $t = 10$; only 7 minutes. Fortunately, servicing every arc at most once in T_i' is a reasonable assumption given the relatively small time frames. If the arc is unsafe at time t, and enough snow for it to be unsafe again is accumulated at time t', the plowing has yielded a benefit of $b_{at} = w_a(t' - t)$. But if the arc was safe at the time of plowing and instead had become unsafe at t', the new time of it becoming unsafe t'' must be determined. This then yields the benefit of $b_{at} = w_a(t'' - t')$.

Those benefits can be used to efficiently calculate the benefit/detriment of changes to a route. For instance, assume a route with services at $t_1, ..., t_n$ on arcs $a_1, ...a_n$. The current benefit then is $\sum_{i=1}^{n} b_{a_i,t_i}$. When rerouting and inserting a new serviced arc a^* after arc a_j at time t^*, by replacing the path $\{a_j, b_1, ..., b_m, a_{j+1}\}$ with $\{a_j, c_1, ..., c_k, a^*, c_{k+1}, ..., c_l, a_{j+1}\}$, all following services are postponed by $\delta = d_{a^*} + \sum_{i=1}^{l} d_{c_i} - \sum_{i=1}^{m} d_{b_i}$. This results in the benefit of $(\sum_{i=1}^{j} b_{a_i,t_i}) + b_{a^*,t^*} + (\sum_{i=j+1}^{n} b_{a_i,t_i+\delta})$. Should a consecutive sequence of service be inserted at $t_1^*, ...t_m^*$, the benefit would be $(\sum_{i=1}^{j} b_{a_i,t_i}) + (\sum_{i=1}^{m} b_{a_i^*,t_i^*}) + (\sum_{i=j+1}^{n} b_{a_i,t_i+\delta})$.

While not directly related to these calculations, we also want to point out that early plowing of an arc a may decrease the benefit. For instance, assume a steady snowstorm causing arcs to be unsafe within 20 minutes, and two different plowing schedules. In the first hour-long schedule, arc a is plowed every 30 minutes, once at $t = 0$ and then at $t = 30$. This results in arc a being safe in the intervals $[0, 20]$ and $[30, 50]$; in total 40 minutes. In the second schedule, the second plowing is brought forward to $t = 10$, resulting in a safe interval $[0, 30]$; in total 30 minutes. This prohibits schemes of optimizing the benefit by simply performing everything as soon as possible.

4.4 Solution Methods

Due to the underlying problem's complexity, we propose to embed ALNS into the rolling horizon framework. ALNS is a renowned metaheuristic that was introduced by Pisinger and Ropke (2007). It has been applied to a multitude of related problems (e.g., Salazar-Aguilar et al. (2012); Quirion-Blais et al. (2017b,a)). To the best of our knowledge, we are the first to apply it to the introduced dynamic stochastic snow plowing problem. Additionally, we also propose problem-specific operators.

4.4.1 Basic component: Adaptive large neighborhood search

ALNS iteratively selects pairs of destroy and repair operators (d and r), which are applied to the incumbent solution s. The selection is based on scores, which are updated depending on the operator pairs' performance during the last interval of v iterations. Every pair begins each interval with a score of 1, which increases for a good performance (i.e., if an operator pair finds a new best solution s^* or a new incumbent solution s). After v iterations, the old scores are overwritten with the currently accumulated scores. The new solution s' is further improved by a variable neighborhood descent (VND), which strengthens it further. An acceptance test using SA is performed to determine whether the solution is accepted as a new incumbent solution or not. SA includes a *temperature* and *cooling* parameters. As long as the temperature is high, the likelihood of accepting worse solutions is also high ($exp(\frac{currentSolutionValue - incumbentSolutionValue}{temperature})$).

Cooling, on the other hand, indicates how much the temperature is lowered in each iteration. The whole process is iterated until it reaches the given stopping criterion. For the pseudocode, see Algorithm 7.

The following standard and problem-specific operators are used:

1. *[d_1] Random cluster*: Consecutive sequences of serviced arcs are randomly chosen and only traversed.

2. *[d_2] Random cluster and shorten*: Consecutive sequences of serviced arcs are randomly chosen and only traversed. If beneficial, paths between serviced arcs are replaced with the shortest path.

3. *[d_3] Greedy cluster*: Consecutive sequences of serviced arcs, whose traversal yields the greatest benefit, are only traversed.

4. *[d_4] Isolated*: Serviced arcs that are the most isolated from the previous and the following serviced arcs, are only traversed.

5. *[d_5] Individual saving cluster*: Consecutive sequences of serviced arcs, whose service, without considering other arcs, yields the smallest benefit, are only traversed.

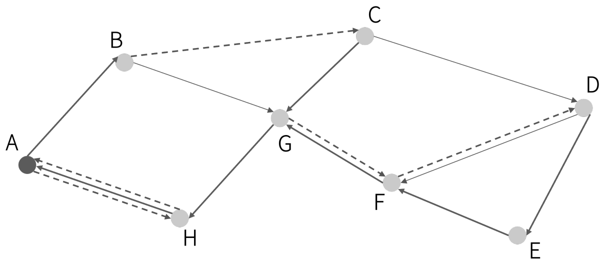
6. *[r_1] Random*: Unserviced arcs are randomly chosen and assigned to a random vehicle. If the vehicle's current route already includes a traversal of the arc, service takes place at that time. Otherwise, *Random path change* is executed.

7. *[r_2] Random path change*: Unserviced arcs are inserted into random routes at a random position by removing a sequence of pure traversals and reconnecting the route with two shortest paths.

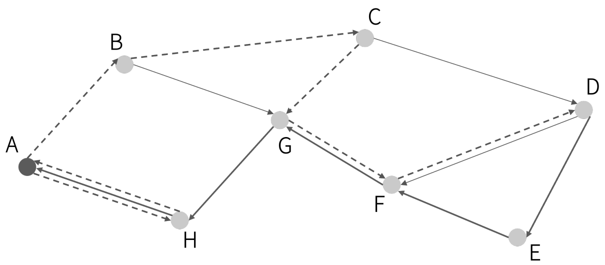
8. *[r_3] Greedy insertion*: Unserviced arcs with the greatest insertion benefits are inserted at their best positions.

9. *[r_4] k-regret*: Unserviced arcs with the highest regret value, when comparing their best and k-best option (where $k \leq 3$), are inserted at their best positions. (E.g., the benefit of best option 100, the benefit of second best option 70, the benefit of third best option 20 $\Rightarrow$ 2-regret of 30, 3-regret of 80.)

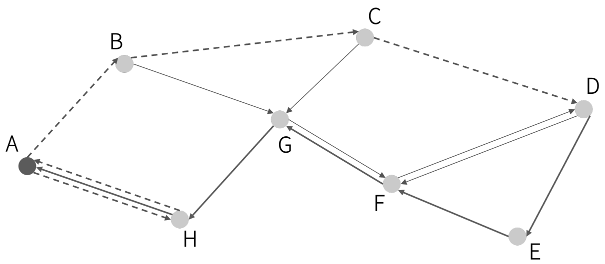
Figure 4.3 illustrates some movements possible with these operators. From the initial solution in Figure 4.3a the operators d_1, d_3, and d_5 could remove two consecutive services (arcs AB & CG), by changing them to pure traversal (cf. Figure 4.3b). The route itself would not change through these operators. In contrast to d_1, d_2 could change the route. In Figure 4.3c, it not only removes arcs AB & CG, but also replaces the path between the services HA & DE with a shorter path (ABCD instead of ABCGFD). All repair operators (except for r_1, should it not call r_2) can change the route as they insert services regardless of whether the arc was already part of the route. E.g., in Figure 4.3d, where the service BG is added to the route between services HA & DE by replacing the path ABCD with ABGFD.



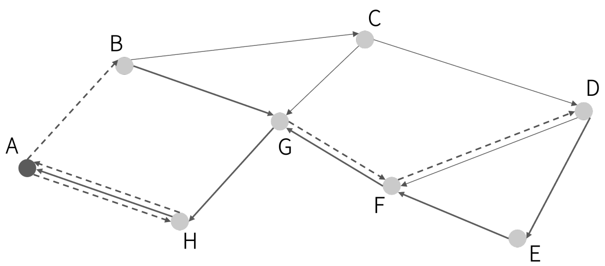
(a) Base route starting at A servicing: HAB, CG, DEFGH. [*Full route AHABCGFDEFGHA*]



(b) Removing a cluster of two consecutive services (AB & CG): HA, DEFGH. (Possible via d_1, d_3, d_5.) [*Full route AHABCGFDEFGHA*]



(c) Shortening the route by exchanging the current path from HA to DE with the shortest path. (Via d_2.) [*Full route AHABCDEFGHA*]



(d) Inserting a service between two consecutive services (HA & DE), which changes the path between them. (Possible via r_2, r_3, r_4.) [*Full route AHABGFDEFGHA*]

Figure 4.3: Illustration of possible changes to a route via the ALNS. The routes start and end at A. Bold lines are traversed and serviced, while dashed lines are only traversed. Thin lines are unused arcs.

For all operators except the random ones (d_1, d_2, r_1, r_2), we include noise factors u, which we randomly draw from an interval $U = [-\psi, +\psi]$. The values used to select the arcs are multiplied by $1 + u$, which increases diversification.

In addition to the destroy and repair operators, we use VND operators that only change when an arc is only traversed or traversed as well as serviced. This is why these operators do not impact the routing. Further, due to limited computational effort, VND operators can be applied whenever a new solution is generated. The following operators are proposed:

1. *[v_1] Beneficial traversals to services*: A previously traversed arc is serviced.

2. *[v_2] Single arc traversal-service-switch on same route*: For an arc that was traversed multiple times during a route and also serviced, the service is moved to a different traversal (e.g., given traversals at t_1 and t_2 and service at t_1, the changes will be that the service is scheduled for t_2 instead of t_1).

3. *[v_3] Single arc traversal-service-switch on different routes*: For an arc that was traversed multiple times and also serviced (on any route), the service is moved to a different traversal.

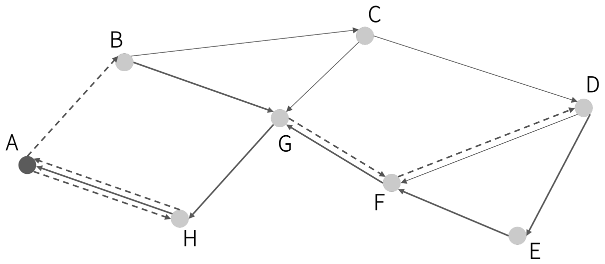
4. *[v_4] Two arcs traversal-service-switch on same route*: A previously serviced arc is only traversed, while a previously only traversed arc is serviced; both arcs are on the same route.

The operators are applied in the order listed above. Should an operator not find an improvement, the next operator is used (e.g., v_3 after v_2). However, if an improvement is found, the VND continues with v_1. The VND ends once v_4 was applied without any improvement. For the pseudocode, see Algorithm 8.

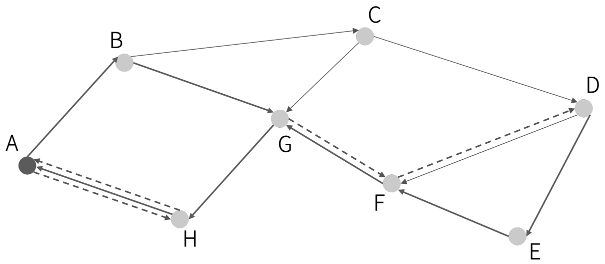
Figure 4.4 illustrates some movements possible with these operators. An initial solution is given in Figure 4.4a. Operator v_1 changes BG to be not only traversed but serviced as well (cf. Figure 4.4b). Operator v_2 causes HA, which was traversed twice and serviced during the first traversal, to be serviced on the second traversal instead (cf. Figure 4.4c). Lastly, operator v_4 causes the previously only traversed arc FD to now be serviced, while the inverse holds true for arc FG (cf. Figure 4.4d).

If no new best solution result is found for w iterations, operators d_1 and d_2 are applied to the incumbent solution several times before the solution is improved using the VND operators (see Algorithm 7, lines 7-10). Whether d_1 or d_2 is used, is determined randomly. The number of times it is applied depends on the iterations since the last best solution was found, and this is also decided randomly (e.g., if it was found recently, it will be applied once; but if the last best is very far in the past, it will be applied 3 to 5 times).

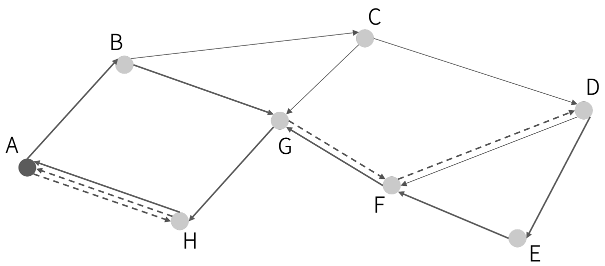
The algorithmic parameters (see Table 4.4) were set similarly to previous work of Fröhlich et al. (2020). Scores were awarded for finding (i) a new best solution, (ii) a solution better than the current incumbent solution but not a new best solution, and (iii) a solution worse than the current incumbent but sufficiently good to be accepted via SA.



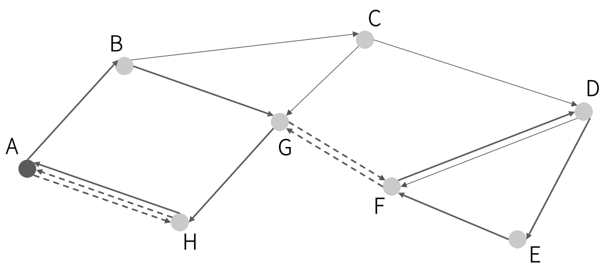
(a) Base route. [*Full route AHABGFDEFGHA*]



(b) Including the service of AB, which was previously just traversed. (Via v_1.) [*Full route AHABGFDEFGHA*]



(c) Servicing HA at the end of the route instead of at the beginning. (Via v_2.) [*Full route AHABGFDEFGHA*]



(d) Including the service of FD, which was previously just traversed, while not servicing FG and instead only traversing it. (Via v_4.) [*Full route AHABGFDEFGHA*]

Figure 4.4: Illustration of possible changes to a route via the VND. The routes start and end at A. Bold lines are traversed and serviced, while dashed lines are only traversed. Thin lines are unused arcs.

Algorithm 7 ALNS

$\{s', s, s^*\}$ ←generateInitialSolution()
while *stoppingCriterionNotReached* **do**
    $\{d, r\}$ ←selectDestroyAndRepairOperators($D, R, scores$)
    $s' \leftarrow r(d(s))$
    $s' \leftarrow$ VND(s')
    $\{s, s^*, scores'\} \leftarrow$
     acceptanceTestViaSimulatedAnnealing($s', s, s^*, scores', d, r$)
    if *iterationsWithoutImprovementSinceLastKick*$= w$ **then**
        s ←randomDestroyAndRepair(s)
        s ←VND(s)
    end if
    $n \leftarrow n + 1$
    if $n = v$ **then**
        $\{scores, scores'\}$ ←updateScores($scores, scores'$)
        $n \leftarrow 0$
    end if
end while
return s^*

Algorithm 8 VND

$i \leftarrow 0$
while $i \neq 4$ **do**
    $\{s, improvementBoolean\}$ ←performNeighborhood(s, i)
    if *improvementBoolean* **then**
        $i \leftarrow 0$
    else
        $i \leftarrow i + 1$
    end if
end while
return s

4.4.2 Basic component: Rolling horizon framework

Given a total time horizon $\mathcal{T}$, the rolling horizon solves subproblems, each with a time horizon of T (equaling the updating interval) and an extra horizon E (equaling the look-ahead). The steps are identical to Section 4.3.4 for all but the first iteration, which requires a slight modification. First, the previously measured snowstorm movements ($M_{-1}, ..., M_{i-1}$) are used to predict the next snowstorm movements ($\widehat{M_i}$ and $\widehat{M_{i+1}}$). Then the next predicted state $\widehat{S}_{i+1}$ is created from the current state S_i, the expected snowstorm movement $\widehat{M_i}$ and solution Sol_i for this updating interval. Based on the next predicted state $\widehat{S}_{i+1}$ and snowstorm movement $\widehat{M_{i+1}}$, the next solution Sol_{i+1} is created using ALNS. At the end of the updating interval, the current state is updated, by using the actual snowstorm movement M_i, instead of the prediction $\widehat{M_i}$. The deviations for the first iteration are that there might not be any snowstorm data available for predicting the movements and that no solution is available for creating the predicted state and updating the actual state. The pseudocode is provided in Algorithm 9. More details on the generation of predictions are provided in Section 4.3.4. If all the streets are safe, the vehicles can return to the depot.

Table 4.4: List of algorithmic parameters

Parameter	Values
Noise ψ	0.05
Scores for ALNS	{10, 3, 1}
Iterations before updating scores v	40
Iterations without improvement before random destroy-and-repair w	1000
Initial temperature	Initial solution value
Cooling factor SA	0.975
Termination criterion (time in seconds)	3600

Algorithm 9 Rolling horizon framework

$\{\widehat{M}_{-1}, \widehat{M}_0\} \leftarrow$predictInitialSnowstormMovements()
$\widehat{S}_0 \leftarrow$createInitialPredictedState$(S_{-1}, \widehat{M}_{-1})$
$Sol_0 \leftarrow$ALNS$(\widehat{S}_0, \widehat{M}_0)$
$S_0 \leftarrow$updateState(S_{-1}, M_{-1})
$i \leftarrow 0$
while $i + 1 < \mathcal{T}/T$ **do**
    $\{\widehat{M}_i, \widehat{M}_{i+1}\} \leftarrow$updateAndCreateNextPredictedMovements$(M_{-1}, ..., M_i)$
    $\widehat{S}_{i+1} \leftarrow$createNextPredictedState$(S_i, \widehat{M}_i, Sol_i)$
    $Sol_{i+1} \leftarrow$ALNS$(\widehat{S}_{i+1}, \widehat{M}_{i+1})$
    $S_{i+1} \leftarrow$updateState(S_i, M_i, Sol_i)
end while

4.5 Computational Study

Our computational study, which was performed on the Vienna Scientific Cluster 4 (3.1 GHz), consists of two parts. First, we compare the proposed ALNS to the optimal results on benchmark instances. Second, we present a newly generated set of real-world-based instances, which we use to derive computational and managerial findings.

4.5.1 Results on benchmark instances

To validate the efficacy of the proposed ALNS, we apply it to the team orienteering arc routing problem (TOARP), as this problem class is arc-based and does not require all arcs to be visited. The main difference is its need for vehicles to return to the depot and a time-independent reward for service. However, as the time-independence is similar to an arc starting with snow but not being further affected by the snowstorm, this characteristic is covered by our proposed ALNS. Therefore, only the need to return to the depot is added.

All experiments are conducted with a runtime of 1 hour on the *Hertz_p0_d_2* and *Hertz_p0_d_3* instances, which have either two or three vehicles and networks with 96–846 arcs, including 10–121 profitable arcs. We benchmark the proposed ALNS against the optimal results available in Archetti et al. (2014), who also had a time limit of 1 hour. For each instance, we perform five runs. Table 4.5 lists the average percentage ratios (i.e., we divide our results by the state-of-the art result, which are optimal solutions in most cases), standard deviations, worst-case ratios, and the number of instances solved to optimality.

We observe an average ratio of 98.46% (i.e., an average percentage gap of less than 1.6%), and a worst-case ratio of about 95%. More detailed results are available in Table C.1 in the

Table 4.5: Benchmarking ALNS against the optimal results available in Archetti et al. (2014).

Instance set	Average ratio	Standard deviation	Worst-case ratio	# of optimal solutions	# of opt. sol. in all runs
Hertz_p0_d_2	98.73%	1.37%	95.39%	11 from 27	7 from 27
Hertz_p0_d_3	98.19%	1.78%	94.18%	8 from 27	7 from 27

Appendix. Furthermore, we examine the ratios after 30 seconds, as this is the smallest updating interval for the rolling horizon framework. Even for the most difficult instances, an average ratio of 94.02% is achieved.

4.5.2 Real-world-based instances

To gain computational and managerial insights, we generate a new set of instances based on data from the city of Vienna. In the following sections, we first describe the details of data generation and then provide detailed results and discussions. As the instances are far too large for the MIP-formulation to deliver results within a reasonable time frame, no comparison to (near) optimal solutions are performed. The instances are made available at *https://bda.univie.ac.at/research/data-and-instances/vehicle-routing-problems/*.

Instance generation

We generate eight instances based on the street network of Vienna (denoted as *vienna*-instances); five smaller ones, and three larger ones. For each district, the set of arcs S initially consists of all arcs starting or ending in a district. However, as Vienna has many one-way streets, S would only be partially connected and some arcs or whole sets may be unreachable. Hence, for every arc a in S that could not be reached when using only S, the shortest path P between any other arc in S and a using the whole street network of Vienna is inserted ($S := S \cup P$). We use Cartesian coordinates to calculate the arcs' centers and normalize them to have 0 as the smallest value (e.g., the smallest x-coordinate in S is $2,000$; therefore, the x-coordinate of $2,500$ is normalized to 500). Traversal times are set according to the length and speed limits.

Further, snowstorm movements are generated according to different shapes (straight line (*straight*), wavy line (*snake*), full circle (*clock*), and half circle that is backtracked (*half-half*)). Figure C.1 in the appendix depicts the movements. Detailed results are not provided as the shape of the snowstorm did not lead to significantly different findings.

Computational results

For each setting, consisting of updating interval, look-ahead, instance, snowstorm movement, and prediction type, we conduct 8 runs of the proposed rolling horizon framework with a duration of $3,600$ seconds (e.g., the problem is split into 120 subproblems, when the updating interval is 30 seconds, and 12 subproblems, when the updating interval is 300 seconds). Table 4.6 lists the applied parameter settings. Table 4.7 lists the number of arcs for the *vienna*-instances, when performing the splitting according to Section 4.3.3. However, as all 16 combinations of updating intervals and look-aheads would have been too confusing, we only list four combinations. Instances *vienna_6* - *vienna_8* were used in later tests, to examine whether observations change for larger instances. *Normal* prediction corresponds to the procedure described in Section 4.3.4, while *perfect* prediction simply reads the input file for the snowstorm and replicates it, and thus has no prediction errors.

Table 4.6: List of parameters

Parameter	**Values**
Updating interval	30, 60, 120, 300
Look-ahead	50, 100, 200, 300
Instance	vienna_1, ..., vienna_5
Snowstorm movement	snow_1, ..., snow_4
Prediction	Normal, perfect

Table 4.7: Information on real-world-based instances

	Arcs when split to be equal or less				
Instance	∞	300	160	80	40
vienna_1	1778	1826	1925	2194	2763
vienna_2	499	509	521	595	739
vienna_3	1670	1675	1727	1900	2251
vienna_4	1469	1469	1472	1526	1647
vienna_5	1214	1246	1296	1465	1855
vienna_6	3637	3668	3765	4183	5180
vienna_7	2498	2567	2725	3259	4495
vienna_8	3828	3840	3884	4084	4795

Tables 4.8-4.11 report the results for the binomial tests that determine whether the results show significant differences based on the applied setting. Detailed results for different movements of snowstorms are provided in Tables C.2 and C.3 in the Appendix. Tables 4.8 and 4.9 compare the influence of the updating interval. Their binomial tests had identical look-aheads (e.g., a comparison of T=30, E=50 with T=60, E=50; but not a comparison of T=30, E=50 with T=60, E=100). The first row denotes the updating interval and the following rows contain the results for different instances. Every result x/y lists the number of snowstorm movements, for which the base setting yields significantly better (x) and significantly worse (y) according to binomial tests (e.g., 0/2 in Table 4.8 for vienna_2 denotes that the updating interval of 120 never performs significantly better, but performs significantly worse than the updating interval of 300 in 2 cases; thus, in the remaining 2 cases, there is no significant difference). Tables 4.10 and 4.11 compare the effect of a look-ahead with different fixed updating intervals (30, 60, 120, and 300). For Tables 4.8 and 4.9, every binomial test consists of 32 comparisons (4 different look-aheads with 8 runs each). For Tables 4.10 and 4.11, every binomial test consists of 40 comparisons (5 different instances with 8 runs each).

Table 4.8 shows a clear picture of shorter updating intervals performing worse than the larger ones, when predictions are imperfect. This is an overall trend, but it is strongest when $T = 30$ and $T = 60$ are compared to the larger intervals. Further, $T = 30$ and $T = 60$ are significantly worse in 57 tests out of 60 (95%) and 33 tests out of 40 tests (82.5%), respectively. The effect decreases between $T = 120$ and $T = 300$ with 7 tests out of 20 (35%) being significantly worse. We explain this observation by the rolling horizon framework effectively splitting the problem into smaller subproblems. Another aspect to be considered is that a smaller T results in less

Table 4.8: Aggregated results for the binomial test comparing updating intervals, when *predicting* snowstorm movements. The first number indicates the number of times a setting performed significantly better ($\alpha = 5\%$) than the other setting; the second number of times is performed significantly worse.

	T=30 compared to			T=60 compared to			T=120 compared to			T=300 compared to		
	T=60	T=120	T=300	T=30	T=120	T=300	T=30	T=60	T=300	T=30	T=60	T=120
vienna_1	0/4	0/1	0/4	4/0	0/0	0/3	1/0	0/0	0/1	4/0	3/0	1/0
vienna_2	0/4	0/4	0/4	4/0	0/4	0/3	4/0	4/0	0/2	4/0	3/0	2/0
vienna_3	0/4	0/4	0/4	4/0	0/4	0/4	4/0	4/0	0/4	4/0	4/0	4/0
vienna_4	0/4	0/4	0/4	4/0	0/4	0/4	4/0	4/0	0/0	4/0	4/0	0/0
vienna_5	0/4	0/4	0/4	4/0	0/4	0/3	4/0	4/0	0/0	4/0/	3/0	0/0
Total	**0/20**	**0/17**	**0/20**	**20/0**	**0/16**	**0/17**	**17/0**	**16/0**	**0/7**	**20/0**	**17/0**	**7/0**

Table 4.9: Aggregated results for the binomial test comparing updating intervals, when *perfectly predicting* snowstorm movements. The first number indicates the number of times a setting performed significantly better ($\alpha = 5\%$) than the other setting; the second number of times is performed significantly worse.

	T=30 compared to			T=60 compared to			T=120 compared to			T=300 compared to		
	T=60	T=120	T=300	T=30	T=120	T=300	T=30	T=60	T=300	T=30	T=60	T=120
vienna_1	0/4	0/4	0/4	4/0	0/4	0/4	4/0	4/0	0/4	4/0	4/0	4/0
vienna_2	0/4	0/4	0/4	4/0	0/4	0/4	4/0	4/0	0/4	4/0	4/0	4/0
vienna_3	0/4	0/4	0/4	4/0	0/4	0/4	4/0	4/0	0/4	4/0	4/0	4/0
vienna_4	0/4	0/4	0/4	4/0	0/3	0/4	4/0	3/0	0/4	4/0	4/0	4/0
vienna_5	0/1	0/4	0/4	1/0	0/4	0/4	4/0	4/0	0/4	4/0	4/0	4/0
Total	**0/17**	**0/20**	**0/20**	**17/0**	**0/19**	**0/20**	**20/0**	**19/0**	**0/20**	**20/0**	**20/0**	**20/0**

Table 4.10: Aggregated results for the binomial test comparing look-aheads, when *predicting* snowstorm movements. The first number indicates the number of times a setting performed significantly better ($\alpha = 5\%$) than the other setting; and the second refers to the number of times a setting performed significantly worse.

	E=50 compared to			E=100 compared to			E=200 compared to			E=300 compared to		
	E=100	E=200	E=300	E=50	E=200	E=300	E=50	E=100	E=300	E=50	E=100	E=200
T=30	0/0	4/0	4/0	0/0	4/0	4/0	0/4	0/4	4/0	0/4	0/4	0/4
T=60	0/1	4/0	4/0	1/0	4/0	4/0	0/4	0/4	4/0	0/4	0/4	0/4
T=120	0/4	0/0	3/0	4/0	0/0	4/0	0/0	0/0	3/0	0/3	0/4	0/3
T=300	0/3	0/3	0/2	3/0	0/4	0/1	3/0	4/0	3/0	2/0	1/0	0/3

Table 4.11: Aggregated results for the binomial test comparing look-aheads, when *perfectly predicting* snowstorm movements. The first number indicates the number of times a setting performed significantly better ($\alpha = 5\%$) than the other setting; and the second refers to the number of times a setting performed significantly worse.

	E=50 compared to			E=100 compared to			E=200 compared to			E=300 compared to		
	E=100	E=200	E=300	E=50	E=200	E=300	E=50	E=100	E=300	E=50	E=100	E=200
T=30	0/4	0/0	4/0	4/0	4/0	4/0	0/0	0/4	4/0	0/4	0/4	0/4
T=60	0/4	0/1	4/0	4/0	2/0	4/0	1/0	0/2	4/0	0/4	0/4	0/4
T=120	0/4	0/0	3/0	4/0	3/0	4/0	0/0	0/3	3/0	0/3	0/4	0/3
T=300	0/1	2/0	3/0	1/0	2/0	3/0	0/2	0/2	0/0	0/3	0/3	0/0

computation time for each problem. As Table 4.9 shows, these findings are similar, when the snowstorm movement is known ahead of time/predicted perfectly (57 out of 60 tests for $T = 30$ and 39 out of 40 tests for $T = 120$). The exception is $T = 300$, compared to $T = 120$, performing even better than previously, as it has significantly better results for all 20 binomial tests.

Table 4.10 shows a clear trend that the look-ahead should increase with the updating interval, when predictions are imperfect. For small T, the small look-aheads $E = 50$ and $E = 100$ perform best, as they dominate $E = 200$ and $E = 300$ in all tests. This can be explained by the computational effort required for long look-aheads. However, $E = 100$ performs better than $E = 50$ when T increases to 60 and 120, and $E = 200$ dominates in case of very long updating intervals ($T = 300$). These observations can be explained by the following: (i) a lower increase in problem size - and therefore computational effort - compared to smaller T (e.g., an increase by 71.4% for $T = 300$ when extending $E = 50$ to $E = 300$), which is outweighed by (ii) larger look-aheads enabling less myopic solutions, and (iii) smoother transitions between sub-problems occurring due to more arcs being available as the “connecting arc” (the last arc starting within the updating interval and ending within the look-ahead).

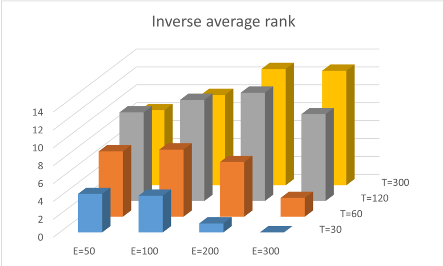
Table 4.11 shows slightly different results when predictions are perfect. The small look-ahead ($E = 100$) proves best for all updating intervals. However, this effect diminishes (e.g., it is significantly better for $T = 30$ in 12 out of 12 tests, but it is only better in 6 out of 12 tests for $T = 300$).

To validate the results, we increase the available computation time by 900% (e.g., a setting with $T = 30$ and $E = 120$ runs with 300 instead of 30 seconds). Regarding updating intervals, the observed trends are supported for $T = 30$ and $T = 60$. Hence, myopic behavior leads to poor solutions. However, the trend of $T = 300$ proving better than $T = 120$ (as it performed significantly better in 7 out of 20 settings and never worse, previously) does not hold as there is even an instance, where $T = 120$ performs better. This suggests that once enough computation time is available, prediction errors can be sufficiently large enough to result in smaller updating intervals (in this case $T = 120$) performing better. Regarding look-aheads, the additional computation time did not bring new insights. All observations, particularly the fact that larger look-aheads are more suitable for larger updating intervals when predictions are imperfect, are confirmed.

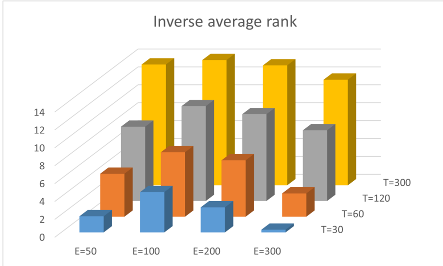
To further examine whether observations change with problem size, we conduct experiments on large instances with 2498–3828 arcs (before splitting). Note that *vienna_1* to *vienna_5* have 499–1778 arcs before splitting. Once more, larger updating intervals perform significantly better, especially when using perfect information.

For $T = 30$ and $T = 60$, without perfect forecasting of the snowstorm, the look-ahead $E = 200$ performs best. However, when using $T = 300$, a trend towards shorter look-aheads is noticeable. This can be explained by the problem becoming too large when using larger look-aheads, as a similar effect was already observed for the largest instance of $T = 200$ and $T = 300$.

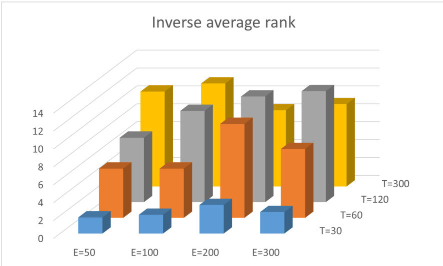
Results are depicted in Figure 4.5, where we show the performance of individual settings on the smaller and larger instances when using imperfect and perfect predictions. The height of the bars in each subfigure corresponds to the inverse average rank of the respective pair of updating interval and look-ahead. For instance, if a pair yields the best average performance



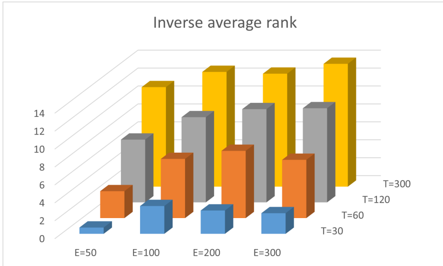
(a) Results for small instances with imperfect prediction. (*Top 3:* 300/200, 300/300, 120/200)



(b) Results for small instances with perfect prediction. (*Top 3:* 300/100, 300/50, 300/200)



(c) Results for large instances with imperfect prediction. (*Top 3:* 120/300, 120/200, 300/100)



(d) Results for large instances with perfect prediction. (*Top 3:* 300/300, 300/100, 300/200)

Figure 4.5: Inverse average ranks for different settings of updating interval T and look-ahead E. The best and worst possible values are 15 and 0, respectively.

in 18 out of 20 aggregated runs, and the second best performance in the remaining runs, it results in the average rank of $(1 \cdot 18 + 2 \cdot 2)/20 = 1.1$ and, as 16 pairs are compared, an inverse average rank of $16 - 1.1 = 14.9$. For perfect predictions, the updating interval is clearly the most impactful parameter, as in Figures 4.5b and 4.5d, the four best average ranks were achieved by $T = 300$. For imperfect predictions (see Figures 4.5a and 4.5c), the updating interval still seems to be more important. However, a poorly chosen look-ahead can be detrimental. When instances are large, the sum of updating interval and look-ahead should not be too large (see Figure 4.5c). However, when instances are small, both large updating intervals and look-aheads yield good results. Lastly, we want to point out that these findings depend on the method's relative efficacy. A more effective method might find larger instances to deliver results more similar to smaller instances, and vice versa for a less effective method. Nonetheless, should these findings prove valuable for a decision maker appropriately assessing their method.

4.6 Conclusion

While a large body of literature on problems related to snow plowing does exist, a gap regarding more efficient and adaptive methods exists, as well. Possible reasons for this may be the technical difficulties in predicting the street conditions. However, as measuring systems and data communication have improved and become more affordable, such adaptive methods have become of interest.

In this study, we examined a novel snow plowing problem based on several challenging characteristics, such as ongoing stochastic snowfall, the necessity for arcs to be plowed repeatedly, and a time-dependent objective function where earlier service could be detrimental. As these characteristics make it difficult to apply exact methods, we tackled the problem using a rolling horizon framework with embedded ALNS. The ALNS was benchmarked against optimal solutions on available TOARP instances.

Our computational study on real-world-based instances examined the effect of updating intervals as well as the effect of look-aheads within the rolling horizon framework. The results suggested that the loss of optimality is low when the horizon is split into larger parts, but that larger problems suffer from imperfect predictions that lead to inefficient moves. The derived results also pointed to the importance of sufficient look-aheads, as well as to the fact that the connection of individual horizons is important. Further, managerial findings included that large look-aheads are more suited for large updating intervals, and vice versa. Additionally, they concluded that small updating intervals should not be paired with very large look-aheads, even when sufficient computational time is available.

Future improvements of the model can be made by adding other common snow plowing considerations such as turn restrictions or heterogeneous fleets. Since the necessity for turn restrictions typically arises from street conditions, they should be assumed to be time-dependent. This would not only make the problem more challenging, but it could also render some insertions or removals infeasible. Another future research direction would be examining different partitioning schemes for the rolling horizon framework.

CHAPTER 5

Conclusion

This thesis studies and reviews multiple vehicle routing problems with safety and security aspects. It gives an extensive overview, and classification on the available literature, while also providing guidance for further research directions. It also studies two novel problems within this field of research - the multi-objective periodic mixed capacitated general routing problem (MO-P-MCGRP) and the time-dependent multi-visit dynamic team orienteering arc routing problem (TMDVD-TOARP). Methodologically, this thesis includes a qualitative literature review, the usage of non-trivial problem-conversions, and the application of state-of-the-art metaheuristics (once arc-based, once node-based), of multi-objective frameworks, and of a rolling horizon framework.

Chapter 2 is a literature review and gives an extensive overview of research done in the field of vehicle routing problems with safety and security problems. We classify the literature into five main categories: (i) transportation of hazardous materials, (ii) patrol routing, (iii) cash-in-transit, (iv) dissimilar routing problems, and (v) modeling of multi-graphs. For each category, details on risk measures, methods, objectives, and commonly found constraints are given. Furthermore, suggestions for promising adaptations and expansions within the fields of hazardous material transportation, patrol routing, and cash-in-transit are given. Details on instances used with safe and secure routing are also provided.

In Chapter 3 a novel vehicle routing problem with security aspects is examined - the MO-P-MCGRP. Nodes, edges, and arcs must be serviced, while minimizing routing costs and maximizing route inconsistency, which is measured by how often street segments are used and by how often service sequences are repeated. The problem's MCGRP-component is remodelled into a capacitated vehicle routing problem (CVRP). We address the possibility that the remodelled problem's non-dominated solutions might be dominated by solutions of the original problem, and therefore determine not just the shortest path between nodes for the CVRP, but the k-shortest paths with a specified diversity. We solve the CVRP with state-of-the-art methods (adaptive large neighborhood search), and examine the efficacy of different multi-objective frameworks (multi directional local search (MDLS), ϵ-box splitting heuristic (EBSH), ϵ-constraint heuristic (ECH)). We test our method and framework on real-world instances, and compare them via multi-objective indicators (range covering, hypervolume, multiplicative unary ϵ, R3). Our results indicate that EBSH delivers on average the best solutions for all multi-objective indicators but one, while MDLS is delivering on average the best for the one remaining indicator. Therefore, we decide to combine EBSH and MDLS into a new multi-objective framework: EBSH*. This new framework improves the results for larger instances slightly. We also examine the effect of considering more than just the shortest path for the CVRP. While considering multiple paths improves the results for smaller instances, it does not show any benefits or even worses results for larger instances, the cause being additional computational effort of handling multi-graphs. We also examine the trade-off between cost and inconsistency. Our finding: that a small cost increase yields a significant inconsistency increase.

In Chapter 4 a novel snow plowing problem is introduced - the TMDVD-TOARP. It has several challenging characteristics: ongoing stochastic snowfall, arcs requiring repeated plowing, a time-dependent objective function where earlier service can be detrimental. We tackle the problem by using an online rolling horizon framework with an embedded arc-based ALNS. We study how results are influenced by the updating intervals and look-aheads - both parameters of the rolling horizon framework. We also examine the difference between perfect and imperfect snowstorm predictions regarding solution quality. As the results of an online heuristic, where the computation time for subproblems might only be 30 seconds, depend even more on the methods efficiency and/or computational power, we also performed runs with tenfold increased computation times. This is equivalent to having a tenfold increase in efficiency/computational power. The main findings are: (i) that larger horizons suffer less from a loss of optimality, but more from imperfect snowstorm predictions, leading in retrospect to inefficient moves, (ii) that large look-ahead are more suited for large updating intervals, and vice versa, (iii) that small updating intervals should not be paired with very large look-aheads, even if the computational power is increased significantly.

Bibliography

Ageev, A. A., Baburin, A., and Gimadi, E. K. (2007). A 3/4-approximation algorithm for finding two disjoint Hamiltonian cycles of maximum weight. *Journal of Applied and Industrial Mathematics*, 1:142–147.

Ageev, A. A. and Pyatkin, A. V. (2009). A 2-approximation algorithm for the metric 2-peripatetic salesman problem. *Diskretnyi Analiz i Issledovanie Operatsii*, 16(4):3–20.

Ahabchane, C., Langevin, A., and Trépanier, M. (2021). Robust optimization for the hierarchical mixed capacitated general routing problem applied to winter road maintenance. *Computers & Industrial Engineering*, 158:107396.

Alinaghian, M. and Naderipour, M. (2016). A novel comprehensive macroscopic model for time-dependent vehicle routing problem with multi-alternative graph to reduce fuel consumption: A case study. *Computers & Industrial Engineering*, 99:210–222.

Allahyari, S., Yaghoubi, S., and Van Woensel, T. (2021). A novel risk perspective on location-routing planning: An application in cash transportation. *Transportation Research Part E: Logistics and Transportation Review*, 150:102356.

Alves, M. J. and Almeida, M. (2007). MOTGA: A multiobjective Tchebycheff based genetic algorithm for the multidimensional knapsack problem. *Computers & Operations Research*, 34(11):3458–3470.

Andelmin, J. and Bartolini, E. (2017). An exact algorithm for the green vehicle routing problem. *Transportation Science*, 51(4):1288–1303.

Androutsopoulos, K. N. and Zografos, K. G. (2010). Solving the bicriterion routing and scheduling problem for hazardous materials distribution. *Transportation Research Part C: Emerging Technologies*, 18(5):713–726.

Androutsopoulos, K. N. and Zografos, K. G. (2012). A bi-objective time-dependent vehicle routing and scheduling problem for hazardous materials distribution. *EURO Journal on Transportation and Logistics*, 1(1-2):157–183.

Archetti, C. and Speranza, M. G. (2014). A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization*, 2(4):223–246.

Archetti, C., Speranza, M. G., Corberán, Á., Sanchis, J. M., and Plana, I. (2014). The team orienteering arc routing problem. *Transportation Science*, 48(3):442–457.

Augerat, P., Belenguer, J., Benavent, E., Corberán, A., Naddef, D., and Rinaldi, G. (1995). Computational results with a branch and cut code for the capacitated vehicle routing problem research report 949-m. *Universite Joseph Fourier, Grenoble, France.*

Baburin, A., Della Croce, F., Gimadi, E. K., Glazkov, Y. V., and Paschos, V. T. (2009). Approximation algorithms for the 2-peripatetic salesman problem with edge weights 1 and 2. *Discrete Applied Mathematics*, 157(9):1988–1992.

Bach, L., Hasle, G., and Wøhlk, S. (2013). A lower bound for the node, edge, and arc routing problem. *Computers & Operations Research*, 40(4):943–952.

bda.univie.ac.at. Bda: Data and instances for vehicle routing problems. https://bda.univie.ac.at/research/data-and-instances/vehicle-routing-problems/. Accessed: 2025-07-30.

Behnke, M. and Kirschstein, T. (2017). The impact of path selection on ghg emissions in city logistics. *Transportation Research Part E: Logistics and Transportation Review*, 106:320–336.

Bektaş, T., Ehmke, J. F., Psaraftis, H. N., and Puchinger, J. (2019). The role of operational research in green freight transportation. *European Journal of Operational Research*, 274(3):807–823.

Ben Ticha, H., Absi, N., Feillet, D., Quilliot, A., and Van Woensel, T. (2019). A branch-and-price algorithm for the vehicle routing problem with time windows on a road network. *Networks*, 73(4):401–417.

Borodin, A. and El-Yaniv, R. (2005). *Online computation and competitive analysis*. cambridge university press.

Bosco, A., Laganà, D., Musmanno, R., and Vocaturo, F. (2013). Modeling and solving the mixed capacitated general routing problem. *Optimization Letters*, 7:1451–1469.

Bosco, A., Laganà, D., Musmanno, R., and Vocaturo, F. (2014). A matheuristic algorithm for the mixed capacitated general routing problem. *Networks*, 64(4):262–281.

Bozkaya, B., Salman, F. S., and Telciler, K. (2017). An adaptive and diversified vehicle routing approach to reducing the security risk of cash-in-transit operations. *Networks*, 69(3):256–269.

Bula, G. A., Prodhon, C., Gonzalez, F. A., Afsar, H. M., and Velasco, N. (2017). Variable neighborhood search to solve the vehicle routing problem for hazardous materials transportation. *Journal of Hazardous Materials*, 324:472–480.

Cao, Y., Wei, W., Bai, Y., and Qiao, H. (2019). Multi-base multi-UAV cooperative reconnaissance path planning with genetic algorithm. *Cluster Computing*, 22:5175–5184.

Capar, I., Keskin, B. B., and Rubin, P. A. (2015). An improved formulation for the maximum coverage patrol routing problem. *Computers & Operations Research*, 59:1–10.

Castro Campos, R. A., Rodriguez Villalobos, C. A., and Zaragoza Martinez, F. J. (2020). Plowing with precedence in polynomial time. *Networks*, 76(4):451–466.

Chen, H., Cheng, T., and Shawe-Taylor, J. (2018). A balanced route design for min-max multiple-depot rural postman problem (MMMRPP): a police patrolling case. *International Journal of Geographical Information Science*, 32(1):169–190.

Chen, X. (2012). Fast patrol route planning in dynamic environments. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 42(4):894–904.

Christofides, N. (1979). The vehicle routing problem. *Combinatorial optimization.*

Constantino, M., Mourão, M. C., and Pinto, L. S. (2017). Dissimilar arc routing problems. *Networks*, 70(3):233–245.

Cordeau, J.-F., Gendreau, M., and Laporte, G. (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30(2):105–119.

Cordone, R. and Wolfler Calvo, R. (2001). A heuristic for the vehicle routing problem with time windows. *Journal of Heuristics*, 7:107–129.

Data.gv.at (2019). Data.gv.at.: Straßengraph wien - data.gv.at. https://www.data.gv.at/katalog/dataset/1039ed7e-97fb-435f-b6cc-f6a105ba5e09. Accessed: 2019-04-04.

De Brey, M. and Volgenant, A. (1997). Well-solved cases of the 2-peripatetic salesman problem. *Optimization*, 39(3):275–293.

De Kort, J. (1992). Bounds for the symmetric 2-peripatetic salesman problem. *Optimization*, 23(4):357–367.

De Kort, J. B. (1993). A branch and bound algorithm for symmetric 2-peripatetic salesman problems. *European Journal of Operational Research*, 70(2):229–243.

De Kort, J. B. J. M. and Volgenant, A. (1994). On the generalized 2-peripatetic salesman problem. *European Journal of Operational Research*, 73(1):175–180.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Dell'Amico, M., Díaz, J. C. D., Hasle, G., and Iori, M. (2016). An adaptive iterated local search for the mixed capacitated general routing problem. *Transportation Science*, 50(4):1223–1238.

Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271.

Doerner, K. F., Gutjahr, W. J., Hartl, R. F., Strauss, C., and Stummer, C. (2006). Pareto ant colony optimization with ILP preprocessing in multiobjective project portfolio selection. *European Journal of Operational Research*, 171(3):830–841.

Duchenne, É., Laporte, G., and Semet, F. (2005). Branch-and-cut algorithms for the undirected m-peripatetic salesman problem. *European Journal of Operational Research*, 162(3):700–712.

Duchenne, É., Laporte, G., and Semet, F. (2007). The undirected m-peripatetic salesman problem: polyhedral results and new algorithms. *Operations Research*, 55(5):949–965.

Duchenne, É., Laporte, G., and Semet, F. (2012). The undirected m-capacitated peripatetic salesman problem. *European Journal of Operational Research*, 223(3):637–643.

Dussault, B., Golden, B., Groër, C., and Wasil, E. (2013). Plowing with precedence: A variant of the windy postman problem. *Computers & Operations Research*, 40(4):1047–1059.

Eglese, R. and Bektaş, T. (2014). Green vehicle routing. In *Vehicle Routing: Problems, Methods, and Applications, Second Edition*, pages 437–458. SIAM.

Erkut, E. and Ingolfsson, A. (2000). Catastrophe avoidance models for hazardous materials route planning. *Transportation Science*, 34(2):165–179.

Erkut, E., Tjandra, S. A., and Verter, V. (2007). Hazardous materials transportation. *Handbooks in Operations Research and Management Science*, 14:539–621.

Fisher, M. L. (1994). Optimal solution of vehicle routing problems using minimum k-trees. *Operations Research*, 42(4):626–642.

Fleischmann, B. (1985). A cutting plane procedure for the travelling salesman problem on road networks. *European Journal of Operational Research*, 21(3):307–317.

Fröhlich, G. E., Doerner, K. F., and Gansterer, M. (2020). Secure and efficient routing on nodes, edges, and arcs of simple-graphs and of multi-graphs. *Networks*, 76(4):431–450.

Fröhlich, G. E., Gansterer, M., and Doerner, K. F. (2023). Safe and secure vehicle routing: a survey on minimization of risk exposure. *International Transactions in Operational Research*, 30(6):3087–3121.

Fröhlich, G. E., Gansterer, M., and Doerner, K. F. (2024). A rolling horizon framework for the time-dependent multi-visit dynamic safe street snow plowing problem. *Networks*, 83(2):236–255.

Gansterer, M. and Hartl, R. F. (2018). Collaborative vehicle routing: A survey. *European Journal of Operational Research*, 268(1):1–12.

Garaix, T., Artigues, C., Feillet, D., and Josselin, D. (2010). Vehicle routing problems with alternative paths: An application to on-demand transportation. *European Journal of Operational Research*, 204(1):62–75.

Ghilas, V., Demir, E., and Van Woensel, T. (2016). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows and scheduled lines. *Computers & Operations Research*, 72:12–30.

Gimadi, E. K. (2015). Efficient algorithms with performance guarantees for some problems of finding several discrete disjoint subgraphs in complete weighted graph. *Applied Mathematics and Computation*, 255:84–91.

Gimadi, E. K., Glazkov, Y. V., and Glebov, A. (2009). Approximation algorithms for solving the 2-peripatetic salesman problem on a complete graph with edge weights 1 and 2. *Journal of Applied and Industrial Mathematics*, 3(1):46–60.

Gimadi, E. K., Glebov, A. N., Skretneva, A., Tsidulko, O. Y., and Zambalaeva, D. Z. (2015). Combinatorial algorithms with performance guarantees for finding several Hamiltonian circuits in a complete directed weighted graph. *Discrete Applied Mathematics*, 196:54–61.

Gimadi, E. K., Istomin, A., and Rykov, I. (2014). On the m-capacitated peripatetic salesman problem with capacity restrictions. *Journal of Applied and Industrial Mathematics*, 8:40–52.

Gimadi, E. K., Istomin, A. M., and Tsidulko, O. Y. (2016). On asymptotically optimal approach to the m-peripatetic salesman problem on random inputs. In *Discrete Optimization and Operations Research: 9th International Conference, DOOR 2016, Vladivostok, Russia, September 19-23, 2016, Proceedings 9*, pages 136–147. Springer.

Gimadi, E. K. and Ivonina, E. (2012). Approximation algorithms for the maximum 2-peripatetic salesman problem. *Journal of Applied and Industrial Mathematics*, 6(3):295–305.

Gimadi, E. K. and Tsidulko, O. Y. (2017). An asymptotically optimal algorithm for the m-peripatetic salesman problem on random inputs with discrete distribution. *Journal of Applied and Industrial Mathematics*, 11:354–361.

Gimadi, E. K. and Tsidulko, O. Y. (2018). Asymptotically optimal algorithm for the maximum m-peripatetic salesman problem in a normed space. In *International Conference on Learning and Intelligent Optimization*, pages 402–410. Springer.

Glebov, A. and Zambalaeva, D. Z. (2012a). An approximation algorithm for the minimum two peripatetic salesmen problem with different weight functions. *Journal of Applied and Industrial Mathematics*, 6:167–183.

Glebov, A. and Zambalaeva, D. Z. (2012b). A polynomial algorithm with approximation ratio 7/9 for the maximum two peripatetic salesmen problem. *Journal of Applied and Industrial Mathematics*, 6:69–89.

Glebov, A., Zambalaeva, D. Z., and Skretneva, A. (2015). A polynomial algorithm with approximation ratio 2/3 for the asymmetric maximum 2-peripatetic salesman problem. *Journal of Applied and Industrial Mathematics*, 9:61–67.

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

Golden, B. L., Wasil, E. A., Kelly, J. P., and Chao, I.-M. (1998). The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets, and computational results. In *Fleet management and logistics*, pages 33–56. Springer.

Haimes, Y. (1971). On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, (3):296–297.

Hajibabai, L. and Ouyang, Y. (2016). Dynamic snow plow fleet management under uncertain demand and service disruption. *IEEE Transactions on Intelligent Transportation Systems*, 17(9):2574–2582.

Hajibabai, L. and Saha, D. (2019). Patrol route planning for incident response vehicles under dispatching station scenarios. *Computer-Aided Civil and Infrastructure Engineering*, 34(1):58–70.

Hajizadeh, R. and Holmberg, K. (2020). A branch-and-dive heuristic for single vehicle snow removal. *Networks*, 76(4):509–521.

Halvorsen-Weare, E. E. and Savelsbergh, M. W. (2016). The bi-objective mixed capacitated general routing problem with different route balance criteria. *European Journal of Operational Research*, 251(2):451–465.

Hamacher, H. W., Pedersen, C. R., and Ruzika, S. (2007). Finding representative systems for discrete bicriterion optimization problems. *Operations Research Letters*, 35(3):336–344.

Holmberg, K. (2019). The (over)zealous snow remover problem. *Transportation Science*, 53(3):867–881.

Hoogeboom, M. and Dullaert, W. (2019). Vehicle routing with arrival time diversification. *European Journal of Operational Research*, 275(1):93–107.

Hsieh, Y.-C., You, P.-S., Lee, P.-J., and Lee, Y.-C. (2015). A novel encoding scheme based evolutionary approach for the bi-objective grid patrol routing problem with multiple vehicles. *Scientia Iranica*, 22(4):1576–1585.

Huang, Y., Zhao, L., Van Woensel, T., and Gross, J.-P. (2017). Time-dependent vehicle routing problem with path flexibility. *Transportation Research Part B: Methodological*, 95:169–195.

Keskin, B. B., Li, S. R., Steil, D., and Spiller, S. (2012). Analysis of an integrated maximum covering and patrol routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):215–232.

Krarup, J. (1995). The peripatetic salesman and some related unsolved problems. In *Combinatorial Programming: Methods and Applications: Proceedings of the NATO Advanced Study Institute held at the Palais des Congrès, Versailles, France, 2–13 September, 1974*, pages 173–178. Springer.

Lacomme, P., Prins, C., Prodhon, C., and Ren, L. (2015). A multi-start split based path relinking (MSSPR) approach for the vehicle routing problem with route balancing. *Engineering Applications of Artificial Intelligence*, 38:237–251.

Lai, D. S., Demirag, O. C., and Leung, J. M. (2016). A tabu search heuristic for the heterogeneous vehicle routing problem on a multigraph. *Transportation Research Part E: Logistics and Transportation Review*, 86:32–52.

Letchford, A. N., Nasiri, S. D., and Oukil, A. (2014). Pricing routines for vehicle routing with time windows on road networks. *Computers & Operations Research*, 51:331–337.

Li, L. Y. and Eglese, R. W. (1996). An interactive algorithm for vehicle routeing for winter—gritting. *Journal of the Operational Research Society*, 47(2):217–228.

Li, S. and Keskin, B. B. (2014). Bi-criteria dynamic location-routing problem for patrol coverage. *Journal of the Operational Research Society*, 65(11):1711–1725.

Lin, C., Choy, K. L., Ho, G. T., Chung, S. H., and Lam, H. (2014). Survey of green vehicle routing problem: past and future trends. *Expert Systems with Applications*, 41(4):1118–1138.

Liu, H., Jin, C., Yang, B., and Zhou, A. (2017). Finding top-k shortest paths with diversity. *IEEE Transactions on Knowledge and Data Engineering*, 30(3):488–502.

Liu, W.-Y., Lin, C.-C., Chiu, C.-R., Tsao, Y.-S., and Wang, Q. (2014). Minimizing the carbon footprint for the time-dependent heterogeneous-fleet vehicle routing problem with alternative paths. *Sustainability*, 6(7):4658–4684.

López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L. P., Birattari, M., and Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58.

Ma, H., Cheang, B., Lim, A., Zhang, L., and Zhu, Y. (2012). An investigation into the vehicle routing problem with time windows and link capacity constraints. *Omega*, 40(3):336–347.

Margolis, J. T., Song, Y., and Mason, S. J. (2022). A multi-vehicle covering tour problem with speed optimization. *Networks*, 79(2):119–142.

Martí, R., Laguna, M., and Campos, V. (2005). Scatter search vs. genetic algorithms: An experimental evaluation with permutation problems. *Metaheuristic Optimization via Memory and Evolution*, pages 263–282.

Matl, P., Hartl, R. F., and Vidal, T. (2017). Heuristic rectangle splitting: Leveraging single-objective heuristics to efficiently solve multi-objective problems. *arXiv preprint arXiv:1705.10174*.

Michallet, J., Prins, C., Amodeo, L., Yalaoui, F., and Vitry, G. (2014). Multi-start iterated local search for the periodic vehicle routing problem with time windows and time spread constraints on services. *Computers & Operations research*, 41:196–207.

Mohri, S. S., Asgari, N., Farahani, R. Z., Bourlakis, M., and Laker, B. (2020). Fairness in hazmat routing-scheduling: A bi-objective Stackelberg game. *Transportation Research Part E: Logistics and Transportation Review*, 140:102006.

Ngueveu, S. U., Prins, C., and Wolfler Calvo, R. (2010a). A hybrid tabu search for the m-peripatetic vehicle routing problem. *Matheuristics*, pages 253–266.

Ngueveu, S. U., Prins, C., and Wolfler Calvo, R. (2010b). Lower and upper bounds for the m-peripatetic vehicle routing problem. *4OR*, 8:387–406.

Ngueveu, S. U., Prins, C., and Wolfler Calvo, R. (2013). New lower bounds and exact method for the m-pvrp. *Transportation Science*, 47(1):38–52.

Nolz, P. C., Semet, F., and Doerner, K. F. (2011). Risk approaches for delivering disaster relief supplies. *OR Spectrum*, 33:543–569.

Oliveira, W. A. d., Moretti, A. C., and Reis, E. F. (2015). Multi-vehicle covering tour problem: building routes for urban patrolling. *Pesquisa Operacional*, 35(3):617–644.

Orloff, C. S. (1974). A fundamental problem in vehicle routing. *Networks*, 4(1):35–64.

Oyola, J. and Løkketangen, A. (2014). GRASP-ASP: An algorithm for the CVRP with route balancing. *Journal of Heuristics*, 20:361–382.

Pan, S., Trentesaux, D., Ballot, E., and Huang, G. Q. (2019). Horizontal collaborative transport: survey of solutions and practical implementation issues. *International Journal of Production Research*, 57(15-16):5340–5361.

Perrier, N., Langevin, A., and Amaya, C.-A. (2008). Vehicle routing for urban snow plowing operations. *Transportation Science*, 42(1):44–56.

Perrier, N., Langevin, A., and Campbell, J. F. (2006a). A survey of models and algorithms for winter road maintenance. Part I: system design for spreading and plowing. *Computers & Operations Research*, 33(1):209–238.

Perrier, N., Langevin, A., and Campbell, J. F. (2006b). A survey of models and algorithms for winter road maintenance. Part II: system design for snow disposal. *Computers & Operations Research*, 33(1):239–262.

Perrier, N., Langevin, A., and Campbell, J. F. (2007a). A survey of models and algorithms for winter road maintenance. Part III: Vehicle routing and depot location for spreading. *Computers & Operations Research*, 34(1):211–257.

Perrier, N., Langevin, A., and Campbell, J. F. (2007b). A survey of models and algorithms for winter road maintenance. Part IV: Vehicle routing and fleet sizing for plowing and snow disposal. *Computers & Operations Research*, 34(1):258–294.

Pisinger, D. and Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403–2435.

Pradhananga, R., Taniguchi, E., and Yamada, T. (2010). Optimization of vehicle routing and scheduling problem with time window constraints in hazardous material transportation. *Journal of the Eastern Asia Society for Transportation Studies*, 8:790–803.

Pradhananga, R., Taniguchi, E., Yamada, T., and Qureshi, A. G. (2014). Bi-objective decision support system for routing and scheduling of hazardous materials. *Socio-Economic Planning Sciences*, 48(2):135–148.

Prieto Curiel, R. and Bishop, S. (2016). A measure of the concentration of rare events. *Scientific Reports*, 6(1):32369.

Prins, C. (2009). A GRASP × evolutionary local search hybrid for the vehicle routing problem. In *Bio-inspired algorithms for the vehicle routing problem*, pages 35–53. Springer.

Prins, C. and Bouchenoua, S. (2005). A memetic algorithm solving the VRP, the CARP and general routing problems with nodes, edges and arcs. In *Recent advances in memetic algorithms*, pages 65–85. Springer.

Prischink, M., Kloimüllner, C., Biesinger, B., and Raidl, G. R. (2016). Districting and routing for security control. In *Hybrid Metaheuristics: 10th International Workshop, HM 2016, Plymouth, UK, June 8-10, 2016, Proceedings 10*, pages 87–103. Springer.

Quirion-Blais, O., Langevin, A., Lehuédé, F., Péton, O., and Trépanier, M. (2017a). Solving the large-scale min-max k-rural postman problem for snow plowing. *Networks*, 70(3):195–215.

Quirion-Blais, O., Langevin, A., and Trépanier, M. (2017b). A case study of combined winter road snow plowing and de-icer spreading. *Canadian Journal of Civil Engineering*, 44(12):1005–1013.

Radojičić, N., Djenić, A., and Marić, M. (2018). Fuzzy GRASP with path relinking for the risk-constrained cash-in-transit vehicle routing problem. *Applied Soft Computing*, 72:486–497.

Reinelt, G. (1991). TSPLIB—A traveling salesman problem library. *ORSA Journal on Computing*, 3(4):376–384.

Saint-Guillain, M., Paquay, C., and Limbourg, S. (2021). Time-dependent stochastic vehicle routing problem with random requests: Application to online police patrol management in Brussels. *European Journal of Operational Research*, 292(3):869–885.

Salazar-Aguilar, M. A., Langevin, A., and Laporte, G. (2012). Synchronized arc routing for snow plowing operations. *Computers & Operations Research*, 39(7):1432–1440.

Schilde, M., Doerner, K. F., Hartl, R. F., and Kiechle, G. (2009). Metaheuristics for the bi-objective orienteering problem. *Swarm Intelligence*, 3:179–201.

Scott, M. (2019). Montreal snow clearing costs went up by $10 million in 2018. https://montrealgazette.com/news/local-news/snow-clearing-costs-up-by-10-million. Accessed: 2025-06-09.

Setak, M., Habibi, M., Karimi, H., and Abedzadeh, M. (2015). A time-dependent vehicle routing problem in multigraph with FIFO property. *Journal of Manufacturing Systems*, 35:37–45.

Setak, M., Izadi, S., and Tikani, H. (2017a). A time-dependent vehicle routing problem for disaster response phase in multi-graph-based network a time-dependent vehicle routing problem for disaster response phase in multi-graph-based network. *International Journal of Industrial Engineering & Production Research*, 28(4):377–387.

Setak, M., Shakeri, Z., and Patoghi, A. (2017b). A time dependent pollution routing problem in multi-graph. *International Journal of Engineering*, 30(2):234–242.

Shenmaier, V. (2014). Asymptotically optimal algorithms for geometric max TSP and max m-PSP. *Discrete Applied Mathematics*, 163:214–219.

Sintef.no (2012). Sintef.no: NEARP / MCGRP. https://www.sintef.no/projectweb/top/nearp/. Accessed: 2020-05-15.

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

Soriano, A., Gansterer, M., and Hartl, R. F. (2018). The two-region multi-depot pickup and delivery problem. *OR Spectrum*, 40(4):1077–1108.

Soriano, A., Vidal, T., Gansterer, M., and Doerner, K. (2020). The vehicle routing problem with arrival time diversification on a multigraph. *European Journal of Operational Research*, 286(2):564–575.

Sugiura, S., Kurauchi, F., and Takagi, A. (2015). Optimal patrol planning using sten for road maintenance. *Journal of the Eastern Asia Society for Transportation Studies*, 11:1654–1665.

Sun, X. and Wang, J. (2018). Routing design and fleet allocation optimization of freeway service patrol: Improved results using genetic algorithm. *Physica A: Statistical Mechanics and its Applications*, 501:205–216.

Sun, X., Wang, J., Wu, W., and Liu, W. (2018). Genetic algorithm for optimizing routing design and fleet allocation of freeway service overlapping patrol. *Sustainability*, 10(11):4120.

Taillard, É. (1993). Parallel iterative search methods for vehicle routing problems. *Networks*, 23(8):661–673.

Taillard, É. D. (1999). A heuristic column generation method for the heterogeneous fleet vrp. *RAIRO-Operations Research*, 33(1):1–14.

Talarico, L., Sörensen, K., and Springael, J. (2015a). The k-dissimilar vehicle routing problem. *European Journal of Operational Research*, 244(1):129–140.

Talarico, L., Sörensen, K., and Springael, J. (2015b). Metaheuristics for the risk-constrained cash-in-transit vehicle routing problem. *European Journal of Operational Research*, 244(2):457–470.

Talarico, L., Sörensen, K., and Springael, J. (2017a). A biobjective decision model to increase security and reduce travel costs in the cash-in-transit sector. *International Transactions in Operational Research*, 24(1-2):59–76.

Talarico, L., Springael, J., Sörensen, K., and Talarico, F. (2017b). A large neighbourhood metaheuristic for the risk-constrained cash-in-transit vehicle routing problem. *Computers & Operations Research*, 78:547–556.

Tarantilis, C. D. and Kiranoudis, C. T. (2001). Using the vehicle routing problem for the transportation of hazardous materials. *Operational Research*, 1:67–78.

Taslimi, M., Batta, R., and Kwon, C. (2020). Medical waste collection considering transportation and storage risk. *Computers & Operations Research*, 120:104966.

Ticha, H. B., Absi, N., Feillet, D., and Quilliot, A. (2017). Empirical analysis for the VRPTW with a multigraph representation for the road network. *Computers & Operations Research*, 88:103–116.

Ticha, H. B., Absi, N., Feillet, D., and Quilliot, A. (2019). Multigraph modeling and adaptive large neighborhood search for the vehicle routing problem with time windows. *Computers & Operations Research*, 104:113–126.

Tikani, H. and Setak, M. (2019). Efficient solution algorithms for a time-critical reliable transportation problem in multigraph networks with FIFO property. *Applied Soft Computing*, 74:504–528.

Tikani, H., Setak, M., and Demir, E. (2021). A risk-constrained time-dependent cash-in-transit routing problem in multigraph under uncertainty. *European Journal of Operational Research*, 293(2):703–730.

Tricoire, F. (2012). Multi-directional local search. *Computers & Operations Research*, 39(12):3089–3101.

Uchoa, E., Pecin, D., Pessoa, A., Poggi, M., Vidal, T., and Subramanian, A. (2017). New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research*, 257(3):845–858.

University of Vienna (2021). Datenbank Universität Wien. https://usearch.univie.ac.at/primo-explore/dbsearch?query=contains,dbcategory,&tab=jsearch_slot&sortby=title&vid=UWI&databases=category,Wirtschaft. Accessed: 2021-06-03.

Vidal, T. (2017). Node, edge, arc routing and turn penalties: Multiple problems—one neighborhood extension. *Operations Research*, 65(4):992–1010.

Vidal, T., Crainic, T. G., Gendreau, M., Lahrichi, N., and Rei, W. (2012). A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research*, 60(3):611–624.

Vidal, T., Crainic, T. G., Gendreau, M., and Prins, C. (2014). A unified solution framework for multi-attribute vehicle routing problems. *European Journal of Operational Research*, 234(3):658–673.

Wallar, A., Plaku, E., and Sofge, D. A. (2015). Reactive motion planning for unmanned aerial surveillance of risk-sensitive areas. *IEEE Transactions on Automation Science and Engineering*, 12(3):969–980.

Wang, H., Lee, Y., and Lee, Y. (2014). Two-stage particle swarm optimization algorithm for the time dependent alternative vehicle routing problem. *Journal of Applied & Computational Mathematics*, 3(4):1–9.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.

Willemse, E. J. and Joubert, J. W. (2012). Applying min-max k postmen problems to the routing of security guards. *Journal of the Operational Research Society*, 63(2):245–260.

Wolfler Calvo, R. and Cordone, R. (2003). A heuristic approach to the overnight security service problem. *Computers & Operations Research*, 30(9):1269–1287.

Xu, S. and Kwon, T. J. (2021). Optimizing snowplow routes using all-new perspectives: Road users and winter road maintenance operators. *Canadian Journal of Civil Engineering*, 48(8):959–968.

Yan, S., Wang, S.-S., and Chang, Y.-H. (2014). Cash transportation vehicle routing and scheduling under stochastic travel times. *Engineering Optimization*, 46(3):289–307.

Yan, S., Wang, S.-S., and Wu, M.-W. (2012). A model with a solution algorithm for the cash transportation vehicle routing and scheduling problem. *Computers & Industrial Engineering*, 63(2):464–473.

Yassen, E. T., Arram, A., Ayob, M., and Nazri, M. Z. A. (2017). A constructive heuristic for police patrol routing problems. *Pertanika J. Sci. Technol*, 25(S6).

Zajac, S. (2016). The bi-objective k-dissimilar vehicle routing problem. In *Computational Logistics: 7th International Conference, ICCL 2016, Lisbon, Portugal, September 7-9, 2016, Proceedings 7*, pages 306–320. Springer.

Zajac, S. (2018). On a two-phase solution approach for the bi-objective k-dissimilar vehicle routing problem. *Journal of Heuristics*, 24(3):515–550.

Zhang, Q. and Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731.

Zhao, J. and Zhu, F. (2016). A multi-depot vehicle-routing model for the explosive waste recycling. *International Journal of Production Research*, 54(2):550–563.

Ziztler, E., Laumanns, M., and Thiele, L. (2002). SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. *Evolutionary Methods for Design, Optimization, and Control*, pages 95–100.

Zografos, K. G. and Androutsopoulos, K. N. (2008). A decision support system for integrated hazardous materials routing and emergency response decisions. *Transportation Research Part C: Emerging Technologies*, 16(6):684–703.

Zografros, K. G. and Samara, S. (1989). Combined location-routing model for hazardous waste transportation and disposal. *Transportation Research Record*, (1245).

APPENDIX A

Appendix for Chapter 2

We have provided a brief summary of the used instances in the papers from Tables 2.3, 2.4, and 2.5. Instances that are well-known are abbreviated according to the common prefix/suffix. Other instances are described via *RWI* (RWIs), *RGI* (randomly generated instances), and *Art. inst.* (artificial instance). While all instances could be seen as artificially created, we classified instances that are built strongly on real-world data (e.g., a specific street network, demands given by companies) as RWIs . Instances that do not belong to the RWI category and that had some random generation mechanism (e.g., random node placement, random distances) are classified as randomly generated instances or RGI. The rest was declared as "Art. inst." Due to many RWIs being very specific, we have omitted their descriptions and would like to point the interested reader to the specific articles, instead.

Instances

- sol, Solomon (1987): For the VRPTW with cartesian coordinates. Instances have either (i) 25, 50, or 100 customers, (ii) narrow or wide time windows, and (iii) have customers clustered, randomly distributed, or a mix thereof.

  - sol for hazmat, Ma et al. (2012): They were extended with population exposure for arcs according to a grid network, where neighboring grids had positively correlated population sizes.
  - sol for patrol routing, Yassen et al. (2017): Four were simulated with each containing 50 hotspots and 5 patrols, with 2 having randomly distributed hotspots and 2 having clusters.
  - sol for cash-in-transit, Michallet et al. (2014): The instances with 100 customers were extended for the periodic VRP with time spread constraints by adding multiple periods and a spread value ϵ.
  - sol for military patrol routing, Margolis et al. (2022): Two instances based on sol were created for the multi-vehicle covering tour problem with time windows. A subset of

20 waypoints was selected, and some of the other 80 customers were randomly set as targets.

- kelly, Golden et al. (1998): For the VRP, with cartesian coordinates. Instances have 200 to 483 customers. Some instances also have a limitation on tour lengths.
- tai, Taillard (1993): For the VRP, with cartesian coordinates. Instances have either 75, 100, 150, or 385 customers.
- egl, Li and Eglese (1996): For the CARP. Instances have either (i) 77 or 140 nodes, (ii) one of three combinations of fleet and vehicle capacity, and (iii) sometimes just required arcs. Required arcs are between 51 and 190.
  - md-egl, Chen et al. (2018): For the min-max multiple depot rural postperson problem. Demands were dropped from egl, and 10 additional depots were evenly added across the instances.
- vrpnc, Christofides (1979): For the VRP with cartesian coordinates. Instances have either 50, 75, 100, 120, 150, or 199 customers.
- TSPLIB, Reinelt (1991): A set of several different traveling salesman problem with lower and upper bounds.
- A, Augerat et al. (1995): For the VRP with cartesian coordinates. Instances have between 31 and 79 customers.
- B, Augerat et al. (1995): For the VRP with cartesian coordinates. Instances have between 30 and 77 customers.
- F, Fisher (1994): For the VRP with cartesian coordinates. Instance have 44, 71, or 134 customers.
- L, Talarico et al. (2017a): For the risk-constrained cash-in-transit VRP with multiple depots and cartesian coordinates. They are a modification of the multi-depot VRP instances by Cordeau et al. (1997) and have between 48 and 360 nodes.
- O, Talarico et al. (2017b): For the risk-constrained cash-in-transit VRP with cartesian coordinates. Instances have between 9 and 336 customers and were generated in such a way that the optimal solutions were known ahead of time. Tours take the shape of an obelisk.
- P, Augerat et al. (1995): For the VRP with cartesian coordinates. Instances have between 15 and 100 customers.
- R, Talarico et al. (2015a): For the risk-constrained cash-in-transit VRP with cartesian coordinates. Instances have (i) between 3 and 19 customers, (ii) one of five risk levels, (iii) demands generated with one of four standard deviations, and (iv) one of five risk thresholds.

- S, Talarico et al. (2017b): For the risk-constrained cash-in-transit VRP with cartesian coordinates. Instances have between 9 and 336 customers and generated in way that the optimal solutions became known ahead of time. Tours take the shape of a spiral.
- V, Talarico et al. (2015a): For the risk-constrained cash-in-transit VRP. They were modified from the VRP library by adding one of five different risk threshold for each. They have between 21 and 300 customers.
- X, Uchoa et al. (2017): For the VRP with cartesian coordinates. Instances have (i) between 100 and 1000 customers, (ii) have customers clustered, randomly distributed, or a mix thereof, (iii) have the depot placed in the center, in a corner, or randomly, and (iv) one of five demand distributions.
- Bula et al. (2017): Instance from Golden et al. (1984) and Taillard (1999) for the heterogeneous fleet VRP are modified to include risk for the traversed arcs.
- Zografros and Samara (1989): A network consisting of 16 nodes for the tri-objective location routing problem.
- Capar et al. (2015): A set of 480 randomly generated instances with 20 for each combination of (i) 10, 20, or 40 hotspots and (ii) one to eight state trooper cars.
- Chen (2012): A 4x4-grid for patrol routing with randomly generated patrol rewards for each of the 80 time units and areas is used.
- Hsieh et al. (2015): A 8x8-grid for patrol routing with 16 nodes to be patrolled between 1 and 3 times is examined.
- Keskin et al. (2012): A set of 480 randomly generated instances with 20 for each combination of (i) 10, 20, or 40 hotspots and (ii) one to eight state trooper cars.
- Sugiura et al. (2015): A 3x3 grid for patrol routing with 48 links that have to either be patrolled once or twice is examined.
- Duchenne et al. (2005): A set of 60 randomly generated instances with five for each combination of (i) 10, 20, 30, 40, 50, and 60 nodes and (ii) random distances or Euclidean distances.
- Duchenne et al. (2007): A set of randomly generated instances ranging from 10 to 190 nodes, and Euclidean distances is used.
- Duchenne et al. (2012): A set of randomly generated instances ranging from 10 to 120 nodes, and Euclidean distances is used.

APPENDIX B

Appendix for Chapter 3

Besides using heuristics, we tried to use mixed integer programming to gain exact results; aiming to use them as benchmarks for the heuristics. However, due to the high complexity of the MO-P-MCGRP, the complete Pareto-frontier could only be calculated for two of the smallest MGGDB instances within the time limit of 72 hours.

For the mathematical formulation of the transformed MO-P-MCGRP, let N_0/N, K_{ij}, P, and L denote the set of POIs with/without a depot, non-dominated paths between POIs i and j, periods, and arcs and edges of the original MCGRP, respectively. For the paths, let x_{ij}^{kp} be the decision variable indicating whether path k between POIs i and j has been used in period p and let c_{ij}^{kp} indicate the costs. Furthermore, t_{ijk}^{s} and t_{ijk}^{e} are binary values that denote the orientation of the POI at the start and end of the path, respectively. For example, $t_{ijk}^{e} = 1$ would indicate that POI j has orientation 1 after arriving there through x_{ij}^{kp}, and $t_{jik}^{s} = 0$ indicates that j is of orientation 0 after leaving it through x_{ji}^{kp}. Let q_{ip} be the remaining capacity after visiting POI i in period p, d_{ip} denote the demand, and Q represent the maximum capacity for a given set of vehicles V. For inconsistency in regard of paths, edges or arcs, decision variables v_{ij} and v_l are introduced. Via v_{ij} the number of times POI i is serviced directly before j except for the first time is determined, while v_l determines the number of times the edges and arcs of the original MCGRP are used beyond the first time. Finally, w_l is the weight of arc or edge l, and b_{ij}^{kl} indicates whether l is part of path k between POIs i and j.

$$\min \sum_{i \in N_0} \sum_{j \in N_0} \sum_{k \in K} \sum_{p \in P} c_{ij}^{kp} x_{ij}^{kp} \tag{B.1}$$

$$\min \sum_{i \in N_0} \sum_{j \in N_0} v_{ij} + \sum_{l \in L} v_l w_l \tag{B.2}$$

$$\sum_{i \in N_0} \sum_{k \in K_{ij}} x_{ij}^{kp} = 1 \qquad \forall j \in N, p \in P \tag{B.3}$$

$$\sum_{i \in N_0} \sum_{k \in K_{ij}} x_{ij}^{kp} (t_{ijk}^{e} + 1) = \sum_{i \in N_0} \sum_{k \in K_{ji}} x_{ji}^{kp} (t_{jik}^{s} + 1) \qquad \forall j \in N_0, p \in P \tag{B.4}$$

$$\sum_{k \in K_{ij}} \sum_{p \in P} x_{ij}^{kp} \leq v_{ij} + 1 \qquad \forall i \in N_0, j \in N_0 \tag{B.5}$$

$$\sum_{i \in N_0} \sum_{j \in N_0} \sum_{k \in K} \sum_{p \in P} x_{ij}^{kp} b_{ij}^{kl} \leq v_l + 1 \qquad \forall l \in L \tag{B.6}$$

$$\sum_{j \in N_0} \sum_{k \in K_{0j}} x_{0j}^{kp} \leq V \qquad \forall p \in P \tag{B.7}$$

$$q_{ip} + M - d_{jp} - q_{jp} - \sum_{k \in K_{ij}} x_{ij}^{kp} M \geq 0 \qquad \forall i \in N_0, j \in N, p \in P \tag{B.8}$$

$$q_{0p} = Q \qquad \forall p \in P \tag{B.9}$$

The objectives, namely minimization of cost and consistency, which is equivalent to maximizing inconsistency, are formulated in (B.1) and (B.2), respectively. Constraints (B.3) denote that every POI has to be serviced exactly once by requiring to have exactly one in-going path. This works together with constraints (B.4), since it causes the left side to be the orientation after arriving from j's predecessor and the right side to be the orientation after leaving from j to its successor. By increasing all t_{ijl}^{e} and t_{ijl}^{s} by one, it furthermore guarantees that every node is left as well. The violations regarding the sequence of service are determined in (B.5). This is done by adding up the number of times POI j was serviced directly after i. Similarly, constraints (B.6) adds up the number of times edges and arcs of the original MCGRP were used to determine violations regarding street segments. The number of vehicles is restricted in constraints (B.7) via the sum of outgoing arcs from the depot. We handle loading restrictions and also connectivity in constraints (B.8), where the remaining capacity of the vehicle after servicing POI j is calculated, and constraints (B.9), where the base capacity for the vehicles is set.

APPENDIX C

Appendix for Chapter 4

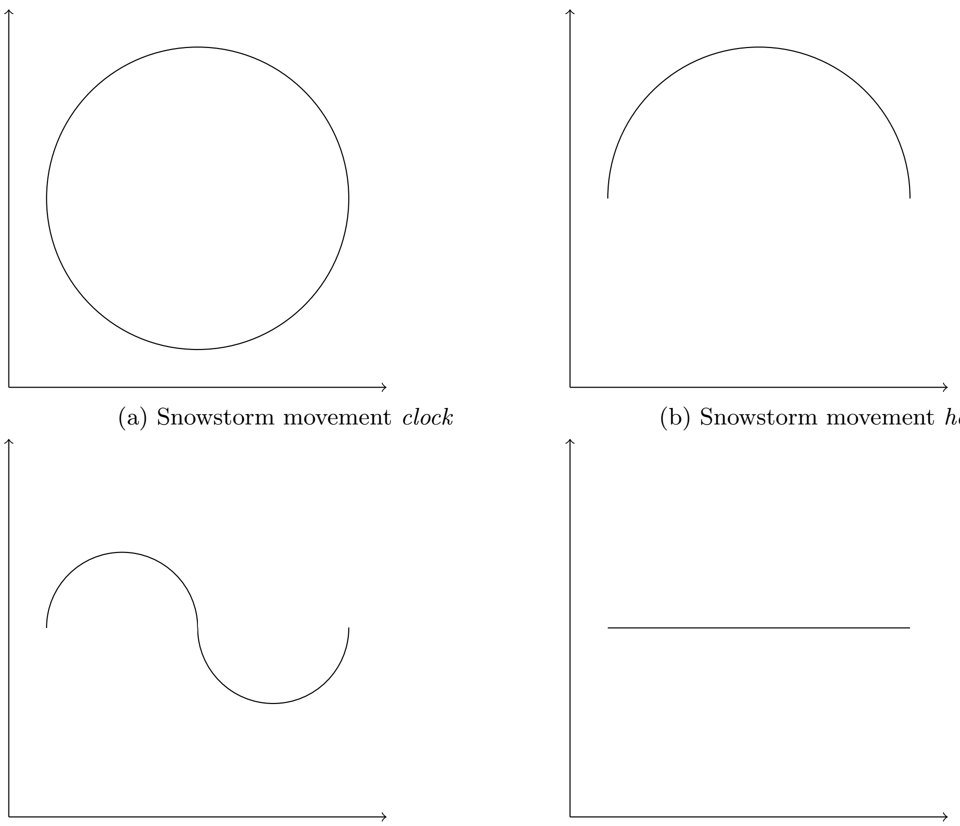
Table C.1 lists the ratios ($\frac{OurResult}{LiteratureResult}$) of individual runs for the Hertz_p0_d2 and Hertz_p0_d3 instances, which have 2 and 3 vehicles, respectively. The networks have 96–846 arcs including 10–121 profitable arcs.

Figure C.1 depicts the 4 types of snowstorm movements. We tried more realistic movements and also more severe ones, to test whether any effect become be visible. Movements *straight* and *snake* seem to be the most natural ones. For *straight*, the wind would not turn at all, while for *snake*, the wind would gradually change its direction and then gradually change its direction back into the original direction. The movement *half-half* might seem natural, when looking at the figure, but after it has moved in a half-circle once, it perfectly backtracks the half-circle. It, therefore, assumes that the wind gradually turns right, then makes a 180-degree turn, and finally gradually turns left. Similarly atypical, *clock* would be described by a wind that is permanently turning into the same direction.

Tables C.2 and C.3 list the results for binomial tests, when having imperfect predictions. In Table C.2, the influence of the updating interval on different snowstorm movements and instances is compared. The first row denotes the comparison of updating intervals. The following rows contain similar results for different instances. Every four rows correspond to one instance (*vienna_1*, ..., *vienna_5*). The superscript $^+$ denotes that the base setting has performed significantly better, for example, for the *clock*-movement, the $T = 60$ setting performs better than $T = 30$. Table C.3 compares the effect of a look-ahead with different fixed updating intervals.

Table C.2 clearly shows smaller updating intervals performing worse than larger updating intervals. The results are consistent for *vienna_2*–*vienna_5* and also do not significantly deviate for specific snowstorm movements. For *vienna_1*, which is the largest instance of the five, updating interval $T = 120$ delivers mixed results, which were better on average, but not generally dominating.

Table C.2 shows not only that smaller look-aheads are desirable for smaller updating intervals, but also the independence of these results from the snowstorm movement. For the two smallest updating intervals, small look-aheads ($E = 50$ and $E = 100$) are desirable for all snowstorm movements, as all entries show that they perform significantly better. For $T = 120$, the look-ahead of $E = 100$ still performs significantly better than the other setting, independent from the



(a) Snowstorm movement *clock* (b) Snowstorm movement *half-half*

(c) Snowstorm movement *snake* (d) Snowstorm movement *straight*

Figure C.1: Illustration of the four different snowstorm movements used.

Table C.1: Results for benchmark tests on TOARP instances using the proposed ALNS.

Instance	**Run 1**	**Run 2**	**Run 3**	**Run 4**	**Run 5**	**Instance**	**Run 1**	**Run 2**	**Run 3**	**Run 4**	**Run 5**
hertzd10_d2	1	1	1	1	1	hertzd10_d3	1	1	1	1	1
hertzd11_d2	1	1	1	1	1	hertzd11_d3	1	1	1	1	1
hertzd12_d2	1	1	1	1	1	hertzd12_d3	0.9762	0.9762	0.9762	0.9762	0.9762
hertzd13_d2	1	1	1	1	1	hertzd13_d3	1	1	1	1	1
hertzd14_d2	1	1	1	1	1	hertzd14_d3	1	1	1	1	1
hertzd15_d2	0.9758	1	0.9653	0.9811	0.9811	hertzd15_d3	1	1	1	1	1
hertzd16_d2	0.9811	0.9811	0.9811	0.9811	0.9811	hertzd16_d3	0.9957	0.9957	0.9957	0.9957	0.9957
hertzd17_d2	1	1	1	1	0.9951	hertzd17_d3	0.9552	0.9552	0.9552	0.9552	0.9552
hertzd18_d2	0.9907	0.9907	0.9907	0.9907	0.9907	hertzd18_d3	0.9844	0.9932	0.9932	0.9932	0.9932
hertzd19_d2	1	0.9785	1	0.9785	1	hertzd19_d3	0.9886	0.9886	0.9943	0.9943	0.9943
hertzd20_d2	0.9795	0.9795	0.9966	0.9966	0.9795	hertzd20_d3	0.9953	0.9953	0.9953	1	1
hertzd21_d2	1	1	1	1	1	hertzd21_d3	0.9514	0.9514	0.9514	0.9514	0.9514
hertzd22_d2	0.9728	0.9728	0.9728	0.9747	0.9728	hertzd22_d3	0.986	0.9686	0.9674	0.9942	0.9686
hertzd23_d2	0.9981	0.9981	0.9981	0.9981	0.9907	hertzd23_d3	1	1	1	1	1
hertzd24_d2	0.9983	0.9983	0.9983	0.994	0.9983	hertzd24_d3	0.9537	0.966	0.9528	0.9528	0.9537
hertzd25_d2	1	0.981	0.9744	0.9744	0.9905	hertzd25_d3	0.9898	0.9898	0.9898	0.9898	0.9898
hertzd26_d2	0.9969	0.9939	0.9977	0.9939	0.9969	hertzd26_d3	1	1	1	1	1
hertzd27_d2	0.98	0.9736	0.9703	0.9736	0.9714	hertzd27_d3	0.9726	0.975	0.9726	0.9726	0.975
hertzd28_d2	0.9762	0.9955	0.9955	0.9955	0.9762	hertzd28_d3	0.9713	0.9713	0.9809	0.974	0.9809
hertzd29_d2	0.9981	0.9981	0.9893	0.9981	0.9845	hertzd29_d3	0.945	0.9881	0.9916	0.9904	0.9916
hertzd30_d2	0.9732	0.9844	0.9844	0.9732	0.9732	hertzd30_d3	0.9565	0.9565	0.9565	0.9565	0.9552
hertzd31_d2	0.9606	0.9606	0.9606	0.9619	0.9606	hertzd31_d3	0.9926	0.989	0.9845	0.9948	0.9897
hertzd32_d2	1	1	1	1	1	hertzd32_d3	0.9859	0.9792	0.9933	0.9963	0.9908
hertzd33_d2	0.9762	0.9612	0.9721	0.9721	0.9721	hertzd33_d3	0.9775	0.9628	0.9675	0.9775	0.9628
hertzd34_d2	0.9539	0.9544	0.9544	0.9539	0.9539	hertzd34_d3	0.9676	0.9878	0.9692	0.9708	0.9718
hertzd35_d2	0.9737	0.9761	0.9972	0.9681	0.9756	hertzd35_d3	0.9596	0.9444	0.9633	0.9523	0.9418
hertzd36_d2	0.9805	0.9805	0.9921	0.9782	0.994	hertzd36_d3	0.956	0.9513	0.9658	0.9902	0.9814

Note: We provide ratios found within 1 h run times.

snowstorm movement. Similar effects can be observed for $T = 300$ and $E = 200$.

Table C.2: Results for the binomial test comparing updating intervals when predicting the snowstorm movement (in %)

T=30	T=60	T=120	T=300	T=60	T=30	T=120	T=300	T=120	T=30	T=60	T=300	T=300	T=30	T=60	T=120
Clock	$\mathbf{0.02}^-$	50	$\mathbf{0}^-$	Clock	$\mathbf{0.02}^+$	27.06	$\mathbf{0}^-$	Clock	50	27.06	$\mathbf{0.03}^-$	Clock	$\mathbf{0}^+$	$\mathbf{0}^+$	$\mathbf{0.03}^+$
HH	$\mathbf{0}^-$	25.17	$\mathbf{0}^-$	HH	$\mathbf{0}^+$	33.88	$\mathbf{0}^-$	HH	25.17	33.88	30.36	HH	$\mathbf{0}^+$	$\mathbf{0}^+$	30.36
Snake	$\mathbf{0.02}^-$	$\mathit{4.81}^-$	$\mathbf{0.21}^-$	Snake	$\mathbf{0.02}^+$	33.88	32.38	Snake	$\mathit{4.81}^+$	33.88	11.33	Snake	$\mathbf{0.21}^+$	32.38	11.33
Straight	$\mathbf{0.81}^-$	14.31	$\mathbf{0.12}^-$	Straight	$\mathbf{0.81}^+$	7.58	$\mathbf{0.01}^-$	Straight	14.31	7.58	19.38	Straight	$\mathbf{0.12}^+$	$\mathbf{0.01}^+$	19.38
Clock	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0.03}^-$	Clock	$\mathbf{0}^+$	$\mathbf{0}^-$	11.48	Clock	$\mathbf{0}^+$	$\mathbf{0}^+$	57.22	Clock	$\mathbf{0.03}^+$	11.48	57.22
HH	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0.04}^-$	HH	$\mathbf{0}^+$	$\mathbf{0}^-$	$\mathit{1.73}^-$	HH	$\mathbf{0}^+$	$\mathbf{0}^+$	20.24	HH	$\mathbf{0.04}^+$	$\mathit{1.73}^+$	20.24
Snake	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0}^-$	Snake	$\mathbf{0}^+$	$\mathbf{0}^-$	$\mathbf{0}^-$	Snake	$\mathbf{0}^+$	$\mathbf{0}^+$	$\mathit{3.84}^-$	Snake	$\mathbf{0}^+$	$\mathbf{0}^+$	$\mathit{3.84}^+$
Straight	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0}^-$	Straight	$\mathbf{0}^+$	$\mathbf{0}^-$	$\mathbf{0}^-$	Straight	$\mathbf{0}^+$	$\mathbf{0}^+$	$\mathbf{0}^-$	Straight	$\mathbf{0}^+$	$\mathbf{0}^+$	$\mathbf{0}^+$
Clock	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0}^-$	Clock	$\mathbf{0}^+$	$\mathbf{0.01}^-$	$\mathbf{0}^-$	Clock	$\mathbf{0}^+$	$\mathbf{0.01}^+$	$\mathbf{0.17}^-$	Clock	$\mathbf{0}^+$	$\mathbf{0}^+$	$\mathbf{0.17}^+$
HH	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0}^-$	HH	$\mathbf{0}^+$	$\mathbf{0}^-$	$\mathbf{0}^-$	HH	$\mathbf{0}^+$	$\mathbf{0}^+$	$\mathbf{0.03}^-$	HH	$\mathbf{0}^+$	$\mathbf{0}^+$	$\mathbf{0.03}^+$
Snake	$\mathbf{0.01}^-$	$\mathbf{0}^-$	$\mathbf{0}^-$	Snake	$\mathbf{0.01}^+$	$\mathbf{0.11}^-$	$\mathbf{0.01}^-$	Snake	$\mathbf{0}^+$	$\mathbf{0.11}^+$	$\mathit{2.51}^-$	Snake	$\mathbf{0}^+$	$\mathbf{0.01}^+$	$\mathit{2.51}^+$
Straight	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0}^-$	Straight	$\mathbf{0}^+$	$\mathit{1}^-$	$\mathbf{0}^-$	Straight	$\mathbf{0}^+$	$\mathit{1}^+$	$\mathit{1}^-$	Straight	$\mathbf{0}^+$	$\mathbf{0}^+$	$\mathit{1}^+$
Clock	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0}^-$	Clock	$\mathbf{0}^+$	$\mathbf{0}^-$	$\mathit{1}^-$	Clock	$\mathbf{0}^+$	$\mathbf{0}^+$	18.85	Clock	$\mathbf{0}^+$	$\mathit{1}^+$	18.85
HH	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0}^-$	HH	$\mathbf{0}^+$	$\mathbf{0}^-$	$\mathbf{0.11}^-$	HH	$\mathbf{0}^+$	$\mathbf{0}^+$	43	HH	$\mathbf{0}^+$	$\mathbf{0.11}^+$	43
Snake	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0}^-$	Snake	$\mathbf{0}^+$	$\mathbf{0}^-$	$\mathit{2.51}^-$	Snake	$\mathbf{0}^+$	$\mathbf{0}^+$	10.77	Snake	$\mathbf{0}^+$	$\mathit{2.51}^+$	10.77
Straight	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0}^-$	Straight	$\mathbf{0}^+$	$\mathbf{0}^-$	$\mathbf{0.11}^-$	Straight	$\mathbf{0}^+$	$\mathbf{0}^+$	43	Straight	$\mathbf{0}^+$	$\mathbf{0.11}^+$	43
Clock	$\mathbf{0.02}^-$	$\mathbf{0}^-$	$\mathbf{0.02}^-$	Clock	$\mathbf{0.02}^+$	$\mathbf{0}^-$	7.48	Clock	$\mathbf{0}^+$	$\mathbf{0}^+$	50	Clock	$\mathbf{0.02}^+$	7.48	50
HH	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0}^-$	HH	$\mathbf{0}^+$	$\mathbf{0}^-$	$\mathit{1}^-$	HH	$\mathbf{0}^+$	$\mathbf{0}^+$	50	HH	$\mathbf{0}^+$	$\mathit{1}^+$	50
Snake	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0.33}^-$	Snake	$\mathbf{0}^+$	$\mathbf{0}^-$	$\mathit{1.13}^-$	Snake	$\mathbf{0}^+$	$\mathbf{0}^+$	27.06	Snake	$\mathbf{0.33}^+$	$\mathit{1.13}^+$	27.06
Straight	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0.33}^-$	Straight	$\mathbf{0}^+$	$\mathbf{0}^-$	$\mathbf{0.01}^-$	Straight	$\mathbf{0}^+$	$\mathbf{0}^+$	41.94	Straight	$\mathbf{0.33}^+$	$\mathbf{0.01}^+$	41.94

Note: Bold numbers indicate strong significance according to the binomial test, while italic numbers indicate weak significance. Superscripts $^+$ denotes that the setting of the block performs better than the setting of the column, while $^-$ denotes the inverse.

Table C.3: Results for the binomial test comparing look-aheads with real time horizons fixed at 30, 60, 120, and 300 when predicting the snowstorm movement (in %)

E=50	E=100	E=200	E=300	E=100	E=50	E=200	E=300	E=200	E=50	E=100	E=300	E=300	E=50	E=100	E=200
Clock	36.01	$\mathbf{0}^+$	$\mathbf{0}^+$	Clock	36.01	$\mathbf{0}^+$	$\mathbf{0}^+$	Clock	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0}^+$	Clock	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0}^-$
HH	6.07	$\mathbf{0}^+$	$\mathbf{0}^+$	HH	6.07	$\mathbf{0}^+$	$\mathbf{0}^+$	HH	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0}^+$	HH	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0}^-$
Snake	36.01	$\mathbf{0}^+$	$\mathbf{0}^+$	Snake	36.01	$\mathbf{0}^+$	$\mathbf{0}^+$	Snake	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0.03}^+$	Snake	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0.03}^-$
Straight	43.4	$\mathbf{0}^+$	$\mathbf{0}^+$	Straight	43.4	$\mathbf{0}^+$	$\mathbf{0}^+$	Straight	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0}^+$	Straight	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0}^-$
Clock	8.14	$\mathbf{0.07}^+$	$\mathbf{0}^+$	Clock	8.14	$\mathbf{0}^+$	$\mathbf{0}^+$	Clock	$\mathbf{0.07}^-$	$\mathbf{0}^-$	$\mathbf{0}^+$	Clock	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0}^-$
HH	56.27	$\mathbf{0.32}^+$	$\mathbf{0}^+$	HH	56.27	$\mathbf{0.01}^+$	$\mathbf{0}^+$	HH	$\mathbf{0.32}^-$	$\mathbf{0.01}^-$	$\mathbf{0}^+$	HH	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0}^-$
Snake	$\mathit{1.19}^-$	$\mathbf{0.05}^+$	$\mathbf{0}^+$	Snake	$\mathit{1.19}^+$	$\mathbf{0.01}^+$	$\mathbf{0}^+$	Snake	$\mathbf{0.05}^-$	$\mathbf{0.01}^-$	$\mathbf{0}^+$	Snake	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0}^-$
Straight	56.27	$\mathbf{0.17}^+$	$\mathbf{0}^+$	Straight	56.27	$\mathbf{0.01}^+$	$\mathbf{0}^+$	Straight	$\mathbf{0.17}^-$	$\mathbf{0.01}^-$	$\mathbf{0}^+$	Straight	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0}^-$
Clock	$\mathit{2.66}^-$	9.98	$\mathbf{0.04}^+$	Clock	$\mathit{2.66}^+$	31.79	$\mathbf{0.01}^+$	Clock	9.98	31.79	$\mathbf{0.35}^+$	Clock	$\mathbf{0.04}^-$	$\mathbf{0.01}^-$	$\mathbf{0.35}^-$
HH	$\mathbf{0.69}^-$	12.79	23.66	HH	$\mathbf{0.69}^+$	31.79	$\mathbf{0.01}^+$	HH	12.79	31.79	18.85	HH	23.66	$\mathbf{0.01}^-$	18.85
Snake	$\mathbf{0.01}^-$	9.98	$\mathbf{0}^+$	Snake	$\mathbf{0.01}^+$	50	$\mathbf{0}^+$	Snake	9.98	50	$\mathbf{0}^+$	Snake	$\mathbf{0}^-$	$\mathbf{0}^-$	$\mathbf{0}^-$
Straight	$\mathit{1.92}^-$	31.79	$\mathit{1}^+$	Straight	$\mathit{1.92}^+$	21.48	$\mathbf{0}^+$	Straight	31.79	21.48	$\mathit{1}^+$	Straight	$\mathit{1}^-$	$\mathbf{0}^-$	$\mathit{1}^-$
Clock	$\mathit{1.21}^-$	$\mathbf{0}^-$	$\mathit{4.48}^-$	Clock	$\mathit{1.21}^+$	$\mathbf{0.01}^-$	5.51	Clock	$\mathbf{0}^+$	$\mathbf{0.01}^+$	$\mathbf{0.25}^+$	Clock	$\mathit{4.48}^+$	5.51	$\mathbf{0.25}^-$
HH	15.37	$\mathit{1}^-$	$\mathit{1}^-$	HH	15.37	$\mathbf{0}^-$	$\mathbf{0.01}^-$	HH	$\mathit{1}^+$	$\mathbf{0}^+$	56.27	HH	$\mathit{1}^+$	$\mathbf{0.01}^+$	56.27
Snake	$\mathit{3.84}^-$	9.46	14.31	Snake	$\mathit{3.84}^+$	$\mathit{1.13}^-$	7.58	Snake	9.46	$\mathit{1.13}^+$	$\mathit{4.94}^+$	Snake	14.31	7.58	$\mathit{4.94}^-$
Straight	$\mathbf{0.03}^-$	$\mathbf{0}^-$	33.18	Straight	$\mathbf{0.03}^+$	$\mathit{1.13}^-$	27.06	Straight	$\mathbf{0}^+$	$\mathit{1.13}^+$	$\mathit{2.35}^+$	Straight	33.18	27.06	$\mathit{2.35}^-$

Note: Bold numbers indicate strong significance according to the binomial test, while italic numbers indicate weak significance. Superscripts $^+$ denotes that the setting of the block performs better than the setting of the column, while $^-$ denotes the inverse.