Quantitative Methods for Controlling the Spread of Invasive Species

by

Samuel M. Fischer

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Applied Mathematics

Department of Mathematical and Statistical Sciences

University of Alberta

Abstract

Invasive species and infectious diseases cause significant ecological and economic harm all over the world. Therefore, substantial effort is made across the globe to prevent the spread and decrease the impact of biological invasions and epidemics. To optimize policy and make control efforts more effective, managers need risk assessment and decision support tools providing them with practical management advice. The development of such tools is the objective of this thesis.

A major vector for many invasive species and infectious diseases is human traffic and trade through road networks. Therefore, reliable predictions of road traffic are needed to facilitate optimal invasion and disease control. Traffic estimates can be used to determine where new invasions and infections are most likely to occur and to optimize prevention measures reducing the introduction of propagules and pathogens to uninfested areas. A challenge, however, is the vast number of potential routes that road travellers could take to reach their destinations. This challenge can make both traffic estimates and effective spread control difficult.

In this thesis, I develop a set of tools to both assess the traffic of potential invasive species or disease vectors and to optimize road-side control measures hindering the propagation of biological invasions and epidemics. I introduce a novel method to compute routes that potential vectors might reasonably take and incorporate the resulting paths in a hybrid gravity and route choice model for vector traffic. The hybrid model accounts for both the travel incentive and the route choice of potential vectors. This hybrid approach makes it possible to fit the model to survey data collected at roads and to determine the major pathways of potential vectors. Fitting the model to road-side survey data facilitates more

accurate traffic estimates and permits the construction of large-scale traffic models, which were difficult to fit with traditional methods. The road-specific traffic estimates, in turn, can be used to determine the best locations to control potentially infested vectors, and I develop a management support tool for this task. The decision support tool can account for location-specific management constraints and provides specific management advice.

I introduce a number of statistical tools to test model assumptions and to assess the credibility of parameter estimates and predictions. In particular, I develop a robust and efficient algorithm to compute profile likelihood confidence intervals. The new algorithm is applicable even in situations in which earlier methods regularly fail or return erroneous results.

I apply all methods developed in this thesis to prediction and control of the transport of zebra and quagga mussels (*Dreissena spp.*) to the Canadian province British Columbia. Dreissenid mussels are invasive in North America and have various negative effects on both ecosystems and human well-being. A major spread mechanism for zebra and quagga mussels is traffic of boaters transporting their watercraft from invaded to uninvaded waterbodies. I apply the newly developed management support tools to optimize placement and operation of watercraft inspection stations, where watercraft are screened for invasive mussels and decontaminated if potentially infested. Considering different management scenarios, I identify general principles for optimal invasive species and disease management.

Preface

This thesis is an original work by S. M. Fischer. Some of the research included in this thesis was conducted as part of collaborations detailed below. M. A. Lewis was the supervisory author in chapters 3-5 and was involved with concept formation and manuscript composition in all chapters.

A version of chapter 2 is currently under review in *Transportation Research Part B: Methodological* as Fischer, S. M. Locally optimal routes for route choice sets.

A version of chapter 3 is currently under review in *Royal Society Open Science* as Fischer, S. M., Beck, M., Herborg, L.-M., and Lewis, M. A. A hybrid gravity and route choice model to assess vector traffic in large-scale road networks. All authors conceived the project; S. M. Fischer conceived the methods jointly with M. A. Lewis and prepared the data for the analysis jointly with M. Beck. S. M. Fischer conducted the mathematical analysis, implemented the model, and wrote the manuscript. All authors revised the manuscript.

A version of chapter 4 will soon be submitted for publication as Fischer, S. M., Beck, M., Herborg, L.-M., and Lewis, M. A. Managing Aquatic Invasions: optimal locations and operating times for watercraft inspection stations. All authors conceived the project; S. M. Fischer conceived the methods jointly with M. A. Lewis, and prepared the data for the analysis jointly with M. Beck. S.M. Fischer conducted the mathematical analysis, implemented the model, and wrote the manuscript. All authors revised the manuscript.

A version of chapter 5 will soon be submitted for publication as Fischer, S. M. and Lewis, M. A. A robust and efficient algorithm to find Profile Likelihood Confidence Intervals. S. M. Fischer and M. A. Lewis jointly conceived the project; S. M. Fischer conceived the algorithm,

conducted the mathematical analysis, implemented the algorithm, and wrote the manuscript. M. A. Lewis revised the manuscript.

A large part of the data used in this thesis were kindly provided by Martina Beck, BC Ministry of Environment and Climate Change Strategy, Conservation Science Section, Victoria, BC.

Acknowledgements

I would like to give thanks to Dr. Mark Lewis, my honoured supervisor, teacher, and mentor, who provided me with guidance, helpful feedback, encouragement, and the freedom to go off the beaten paths. He was an example for me in many ways, lifted my eyes consistently towards the broader questions, and changed the way I think of science. Through his leadership, he created a supportive atmosphere of joint learning, teamwork, and friendship in his lab. I will always be grateful for the opportunity to be part of this lab and to study under his supervision.

I would also like to thank my co-supervisor Dr. Leif-Matthias Herborg and my supervisory committee member Dr. Michael Li for their advice, support, helpful comments, and encouragement. Despite the mysterious abundance of technical glitches that kept afflicting us, I always looked forward to our supervisory committee meetings.

I furthermore want to give thanks to Martina Beck, who constantly supported me with her expertise in invasive species management and provided me with data and resources without which this thesis would not have been possible. In addition, I want to thank Kimberley Wilke-Budinski, who made administrative matters something I would look forward to and who had a large role in making the University of Alberta a home for me.

In the last five and a half years that I spent in my PhD program, I was blessed in so many ways and by so many people that it will be hard to list them all. It is often the little things that make a big difference. I want to thank all the awesome people (listed in no particular order) who have discussed research problems with me; provided me with constructive feedback; took me out of my working bubble to the mountains, to cultural

events, or to the choir; shared a lunch or coffee break with me; gave me spiritual support; or facilitated my physical and mental recreation by joining me (or letting me join) in soccer, badminton, football, and board games. Furthermore, I want to thank my lovely landlady June Hunt and her family, who provided me with a place to live, which became home very quickly.

My transition to the graduate program at the University of Alberta, and thus this thesis, originated from the simple idea to spend some time studying somewhere abroad. Looking back, I deem myself incredibly fortunate that I ended up being admitted to the University of Alberta. This lucky circumstance I largely owe to my respected undergraduate supervisor Dr. Horst Malchow and his advice and dedicated support. I also want to thank the German Academic Foundation (Studienstiftung) and the German Academic Exchange Service (DAAD), who supported me with personal advice, seminars, and funding before and during the first two years of my program in Edmonton. In line with these agencies, I want to acknowledge the Canadian Aquatic Invasive Species Network (CAISN) and the Natural Sciences and Engineering Research Council of Canada (NSERC), who made this research possible through their funding.

Among all my supporters, my family was outstanding. They bore with me, built me up, encouraged me, cared for me despite the distance; they reminded me that there is a life beyond PhD research. Among the uncounted things I have been blessed with, my family is likely the greatest gift.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

General introduction

Human traffic and trade are major vectors for infectious diseases and invasive species (Karesh et al., 2005; Kimball, 2006; Hulme, 2009), which have significant effects on human well-being, economy, and the functioning of ecosystems (Pimentel et al., 2005; Kimball, 2006; Pejchar and Mooney, 2009). Examples include human diseases such as the Severe Acute Respiratory Syndrome (SARS) distributed via airplane travellers (Kimball, 2006), animal diseases such as avian flu spreading via trade of infected birds (Van Den Berg, 2009) and invasive species such as zebra and quagga mussels (*Dreissena spp.*) "hitchhiking" on trailered watercraft (Johnson et al., 2001). Due to the vast damages infectious diseases and invasive species cause, significant efforts are made to prevent their spread and reduce their impact (Shine et al., 2010; Johnson et al., 2017; Turbelin et al., 2017).

A large body of research is concerned with developing tools to make the control of infectious diseases and invasive species most effective (Lewis et al., 2016). Two major toolsets are needed to optimize infectious disease and invasive species management: tools to gain an understanding of the epidemic or invasion process, and tools to optimize management actions given this information. This thesis seeks to address both these tasks. Methods to estimate the traffic of potential disease or invasive species vectors are developed as well as a management support tool using these results to optimize control. Hence, this thesis provides, within its range of applicability, a specific but comprehensive toolset to minimize the spread of infectious diseases and invasive species.

Specifically, this thesis considers the scenario in which a disease or invasive species spreads by means of human road traffic and is managed via road-side control measures. This scenario is challenging from a modelling perspective, as the behaviour of road travellers may vary among individuals and could be affected by a variety of factors. Hence, sophisticated modelling tools are needed to address this management problem. The main result of this thesis is a management support tool that (1) facilitates risk assessment by estimating vector pressure to entities of management concern and (2) provides specific management advice on when and where to apply control actions.

Though the mechanisms behind the spread of infectious diseases and invasive species are often similar – making relevant theory applicable to both issues alike – I will focus on management of invasive species in this thesis. In particular, I will apply the developed methods to the control of zebra and quagga mussels in the Canadian province British Columbia (BC). Zebra and quagga mussels are invasive in North America and cause severe economic and ecological damages (Rosaen et al., 2012; Karatayev et al., 2015b).

Below, I give a brief introduction on the issues surrounding invasive species management, providing background on the ecology of invasive species, their dispersal, impact, and management, and review research on optimal management. As modelling plays an integral role in this thesis, I furthermore discuss earlier work on invasive species modelling and the challenges associated with this task. Finally, I provide some background on zebra and quagga mussels and give an overview of the structure of this thesis.

1.1 Biological invasions

1.1.1 A working definition of invasive species

An important step towards effective invasive species management is to gain a general understanding of biological invasions. The very first challenge is thereby to find a clear definition of the term "invasive". Though the militaristic connotation of the word may evoke the unam-

biguous image of a species aggressively conquering habitat outside its native range, defining the term "invasive" has proven difficult and sparked a controversial debate among ecologists (Lockwood et al., 2013). For example, it can be unclear how far from its home range a species is considered "non-native" (Colautti and MacIsaac, 2004; Valéry et al., 2009), and how the invasion process can be distinguished from "natural" range extension and colonization (Davis et al., 2001). Similarly, arguments have been made on to what extent the impact of a species determines its status as invasive (Richardson et al., 2000; Daehler, 2001). Even the term "invasive" in itself has been criticized, as its negative connotation may convey a value statement that may neither be accurate (Davis et al., 2011) nor even justifiable without stepping outside the realm of science (Simberloff, 2003; Chew and Carroll, 2011).

In this thesis, I consider biological invasions from the management perspective. This resolves potential issues with a negative connotation of the word "invasive", as a management desire – and thus a value statement justifying the control of the species – is already presumed. In a similar manner, this perspective permits the characterization of invasive species on a functional level, taking some liberty in aspects of a general definition. Within this thesis, I will consider a species invasive if (1) it is not present in some suitable habitat, (2) there is a significant chance that the species will be introduced to this habitat, (3) the species has the potential to establish and spread within and from this habitat, and (4) the species has a (negative) impact potentially motivating management.

This working definition coincides largely with the invasion process formalized by Lockwood et al. (2013). Note, however, that some species may fulfill this definition even though they belong to the general native species pool of the "invaded" habitat and would therefore not be considered invasive by many ecologists. Similarly, species that have not yet invaded any foreign habitat (and are thus not invasive in the classical sense) could fall in the range of this definition due to their *potential* to invade. Furthermore, as the definition includes aspects variable in space, time, and human judgment and actions, no species is considered inherently invasive. For example, a species stopping to disperse because it has invaded all suitable habi-

tat looses its status as invasive species even though it could still be (rightfully) considered alien based on its dispersal history. For these reasons, the given working definition is not suited as a general definition of invasive species. Nonetheless, important mechanisms behind biological invasions can be understood based on the four given characteristics, especially in the context of management.

1.1.2 The mechanisms behind biological invasions

To understand the mechanisms behind biological invasions, we may consider the four defining properties of invasive species. With regard to the first and the third property, we may ask how suitable habitat can be characterized, and which factors facilitate and hinder establishment of an invasive species. Habitat suitability can depend on a variety of factors. These may include abiotic factors, such as climate and topography (Hirzel and Le Lay, 2008), and biotic factors, such as presence of predators, prey, competitors, and mutualists (Levine and D'Antonio, 1999; Simberloff and Von Holle, 1999). Furthermore, establishment can both be hindered and facilitated by disturbances, such as natural disasters (Hobbs and Huenneke, 1992). Finally, invadable habitat must be sufficiently segregated from the considered species' home range, as the new habitat would already have been invaded otherwise (Seebens et al., 2013).

Looking at the second property, dispersal of invasive species can be classified as human-aided or natural and, in the former case, intentional or unintentional (Lockwood et al., 2013). A major driver for human-aided dispersal of invasive species is global trade (Hulme, 2009). Thereby, propagules may be transported attached to carriers (Kolar, 2002; Von der Lippe and Kowarik, 2007) or along with goods (Johnson et al., 2001; Koch et al., 2012; Drake and Mandrak, 2014). Intentional distribution may happen through import and release of food or game species (Lockwood, 1999; Mack, 2003). The nature of the dispersal mechanism determines where, how frequently, and how abundantly propagules are introduced to new

habitat. Therefore, knowing the spread mechanism is key to understanding the dynamics of an invasion.

Considering the fourth property, the impacts of invasive species are diverse in magnitude and nature. The arguably most direct impact of invasive species is their effect on the invaded ecosystems. As invasive species may be lacking enemies limiting their spread, invasives often grow quickly in abundance (Keane, 2002). Since the host systems are often poorly adapted to the foreign species, species loss (Bellard et al., 2016) and loss in ecosystem function (Pejchar and Mooney, 2009) can be the consequence. As a result, the impact of invasive species can go far beyond the realm of ecology. For example, direct economic damage may be encountered if invasive pests decrease the yields of crop harvests (Pimentel et al., 2000). Less direct effects include decreasing touristic value of sites, e.g. due to decreasing water quality in lakes (Rosaen et al., 2012). Hence, invasive species cause tremendous costs to economies all over the world (Pimentel et al., 2005). However, the cultural impact of invasive species, e.g. through loss of culturally important species, may be just as significant (Pfeiffer and Voeks, 2008).

In the past century and present, anthropogenic influences have increased the prevalence of each of the defining characteristics of invasive species. For example, climate change alters abiotic conditions so that species can invade habitats in which they were previously unable to establish (Hellmann et al., 2008). The rapid increase in traffic and trade opens new pathways for the spread of invasions and has led to a boost in dispersed propagules (Hulme, 2009). At the same time, anthropogenic ecosystem disturbances can pave the way for new invasions and extend their effects (Didham et al., 2007). This increases the need for effective invasive species management.

1.2 Management of invasive species

1.2.1 Management options and current policy

The typical goal of invasive species management is to reduce some impact of invasive species. To mitigate this impact, management can target different stages of the invasion process: transport, establishment and spread, and impact. If no live propagules are introduced to new habitat, they cannot establish and spread; if a species cannot establish and spread, it will likely have no effect; if a species does not have an effect, the management incentive is lost.

Transport of propagules can be divided into three substages: uptake of propagules, transport of live propagules, and release of propagules into new habitat (Carlton and Ruiz, 2005). Each of these substages can be addressed with management. Pickup of propagules may, for example, be reduced by limiting access to donor regions; survival of propagules may be decreased by treatment of carriers or transported goods (Briski et al., 2013); and release of propagules may be inhibited by imposing import restrictions (Johnson et al., 2017). Establishment and spread in recipient habitat may be counteracted by eradicating new populations (Pluess et al., 2012). Finally, the impact of invasives may be mitigated by decreasing the species density in infested habitat (Yokomizo et al., 2009) or even adjustment to the new circumstances (McDermott et al., 2013; Marbuah et al., 2014).

Theoretical and empirical studies suggest that it is often more cost-effective to prevent the introduction of live propagules rather than managing established populations of invasive species (Leung et al., 2002; Lodge et al., 2006; Pluess et al., 2012). Following such insights, many governments have issued regulations restricting the import of particular goods and require treatment of potentially infested freight and carriers (Shine et al., 2010; Johnson et al., 2017; Turbelin et al., 2017). To facilitate rapid response measures and timely eradication, government agencies have furthermore established early detection and information sharing

networks (Simpson et al., 2009). In addition, there have been coordinated attempts to eradicate established invasive species (Wilson et al., 2013; Jones et al., 2016).

1.2.2 Optimizing management

In line with the significant efforts made to control invasive species, considerable research effort has been made to identify optimal management strategies. Many studies on optimal invasive species management consider bio-economic models pairing an ecological invasion model with an economic model for invasion and control costs (Potapov and Lewis, 2008; Potapov et al., 2008; Finnoff et al., 2010; Carrasco et al., 2010; Epanchin-Niell and Wilen, 2012). Though these models were often developed with focus on specific species, some general insights were consistent throughout studies. For example, several results emphasize the importance of managing invasions early, when they are spatially contained (Finnoff et al., 2010; Blackwood et al., 2010; Epanchin-Niell and Wilen, 2012). In later stages of invasions, when most suitable habitat patches are invaded already and act as secondary propagule sources, it can be more cost-effective to cease control efforts completely (Potapov et al., 2008; Finnoff et al., 2010). Several studies furthermore highlighted the benefit of controlling invasive species with actions variable in time and space (Albers et al., 2010; Finnoff et al., 2010; Epanchin-Niell and Wilen, 2012).

A major challenge involved with bio-economic modelling is the necessity to estimate the costs of invasions. Though attempts have been made to estimate the economic damages caused by invasive species (Pimentel et al., 2000, 2005; Rosaen et al., 2012), it is difficult to put cost labels on cultural and aesthetic loss and to incorporate ethical considerations in economic models. Therefore, bio-economic approaches cannot provide an "objectively" optimal solution to invasive species management.

A second limitation of many theoretical studies on optimal invasive species management is their relatively high level of abstraction. Though many of the developed tools could be applied to real management scenarios, it is challenging to model real ecosystems, manage-

ment options, and management constraints spatially explicit and detailed enough to yield *specific* management advice. Knowing of these challenges and the uncertainty inherent to invasion models, managers often rely on qualitative models and decision support tools despite the abundance of developed quantitative tools. Therefore, it remains an important task for researchers to develop quantitative methods that are easily applicable by managers and provide concrete management advice in specific situations. This will be the major goal of this thesis.

1.2.3 Solving high-dimensional management problems

Typically, optimizing management strategies on a detailed level requires the consideration of many control options and constraints. Solving such high-dimensional problems is rarely possible with analytical techniques, and even numerical solutions can be difficult to obtain. These challenges can be overcome if the objective function and constraints are convex or, even better, linear functions. Convex problems with thousands of variables and constraints can be solved within seconds via interior point methods. Consequently, convex and linear programming has also been used in the context of invasive species management (Hastings et al., 2006).

Despite the computational advantages of convex optimization, convex and linear programming may not be directly applicable if the optimized control policy involves discrete decisions, such as the choice of control locations. Though it is often possible to model decision problems with convex functions, the integer constraints required to model discrete choices make these problems NP-hard in general (Conforti et al., 2014). Although there is an established theory to solve convex or linear integer problems (Conforti et al., 2014), it is often impossible to find optimal solutions in reasonable time, and approximate solutions may be the best possible result (Ageev and Sviridenko, 2004). Nonetheless, since any feasible solution satisfying integer constraints is also admissible if these constraints are removed, convex and linear programming can be used to *bound* the optimal solution of convex and

linear integer problems. This is helpful when a highly optimal solution is sought (Ageev and Sviridenko, 2004).

1.3 Modelling the spread of invasive species

An important tool for informed invasive species management are invasion models. These models can be used to predict the spread of invasive species, thus facilitating early detection and rapid response measures, and to assess how management actions affect the rate and impact of future invasions. Furthermore, models can be used as a research tool to increase our understanding of invasive species and their dispersal. This understanding, in turn, may facilitate management later.

Invasion models differ in their level of abstraction and the invasion stages they cover. A comprehensive invasion model would need to consider both transport of invasive species (involving uptake, survival, and release of propagules) and their chance to establish in the new habitat (Lewis et al., 2016). Modelling propagule pressure and establishment jointly is particularly important if Allee effects strongly impact the dynamics of small populations (see e.g. Potapov and Lewis, 2008). Though joint propagule transport and establishment models have been constructed for a variety of species and scenarios (Leung et al., 2004; Potapov and Lewis, 2008; Seebens et al., 2013), developing, fitting, and analyzing such comprehensive models can be challenging or, if data are missing, infeasible. Therefore, many studies focus on modelling either the introduction of propagules or the suitability of habitat for invaders (Lewis et al., 2016).

1.3.1 Modelling traffic and trade

Many models for invasive species spread focus on one particular vector driving the spread of the considered species at the considered scale. As human traffic and trade are major vectors for invasive species (Lockwood et al., 2013), models for invasive species spread must often account for the social and economic factors influencing human decisions. While high-

quality data on human trade are often available (see e.g. Seebens et al., 2013), modelling human behaviour on an individual-based level is more challenging in general, because many different factors can affect individuals' choices, and individual-specific data are often difficult to collect.

As road traffic is a major vector for several invasive species (Johnson et al., 2001; Von der Lippe and Kowarik, 2007; Koch et al., 2012; Drake and Mandrak, 2014), modelling road traffic is important to understand, predict, and manage invasions. Models for road traffic often consist of two components, one modelling the travel incentive (who drives where how often), and one modelling the route choice. In many instances, it is sufficient to model the travel incentive, as the route choice may not affect the distribution of invasive species. However, incorporating the route choice is necessary if invasive species can invade habitat surrounding roads (e.g. weeds, Von der Lippe and Kowarik, 2007), if control measures are applied at roads, or if data obtained at road sides shall be used to fit the models.

1.3.2 Modelling travel incentive

Often, the travel incentive of road travellers is modelled with so-called gravity models (Bossenbroek et al., 2001; Leung et al., 2004; Potapov et al., 2010; Mari et al., 2011; Muirhead and MacIsaac, 2011). These models, originally developed in the context of economics (Anderson, 2011), build on the assumption that the mean traffic flow between an origin and a destination is proportional to the "repulsiveness" of the origin (e.g. the number of potential travellers living at the origin) the "attractiveness" of the destination (e.g. number of touristic facilities), and a power of the distance between the origin and destination. To increase the mechanistic validity of gravity models, researchers often introduce additional constraints, for example to ensure that the estimated outbound traffic flows do not exceed the number of individuals residing at an origin (Wilson, 1970; Muirhead et al., 2011). Other models account for unknown underlying processes by understanding traffic as a stochastic process (Flowerdew and Aitkin, 1982; Potapov et al., 2010). However, as an alternative to gravity models, travel

incentive could also be modelled mechanistically, e.g. with random utility models (Siderelis and Moore, 1998; Chivers and Leung, 2012).

Models for travel incentive are often fitted via traveller counts collected at origins and destinations, or mail-out surveys collecting details on past and planned trips from potential travellers. Both these data sources, however, may represent only small fractions of the traffic. Origin/destination based sampling is difficult if many origins and destinations are considered or if origins and destinations have many access points. Mail-out surveys, in turn, can rarely cover the complete set of potential vectors and suffer from high sampling error if only few respondents make trips relevant for the study. These challenges are most prevalent in large-scale systems, in which, first, many and large origins and destinations are considered and, second, the number of travelling vectors is small in comparison to the number of vectors who could *potentially* start a trip. For example, millions of people in North America could potentially travel to British Columbia, but only few will actually do so. Identifying the latter individuals can be difficult. Due to these challenges, most models for the traffic of invasive species vectors rely on a single, large survey. This, in turn, makes it difficult to discern systemic stochasticity from modelling error.

A potential solution to this problem is to survey travellers at roads used by many individuals. Often, long-distance traffic concentrates on a small set of major highways (see Abraham et al., 2010), which would therefore be promising locations for traffic samples. Sampling traffic in multiple time intervals permits assessment of stochastic traffic variations and thus a thorough model validation. For these reasons, this approach will be pursued in this thesis. Note, however, that travellers could travel along various (and also unmonitored) routes. Therefore, survey data obtained at road sides are of limited use unless combined with a route choice model.

1.3.3 Modelling route choice

Modelling route choice is a challenging task due to the vast number of possible routes, the variation between individuals, and the multitude of factors potentially affecting route choice. To account for the first challenge, many models understand route choice as a two step process, in which agents first determine a set of potentially suitable routes, from which after closer consideration the final route is chosen (Prato, 2009). This approach is justified via the assumption that travellers do not have the capacity to consider all possible alternatives (Di and Liu, 2016). The variation between individuals is often accounted for with stochastic models for the final route choice (Prato, 2009).

Typically, the heuristic that individuals apply to determine route choice candidates is not known precisely. A number of approaches exist to compute promising routes (Bovy, 2009). Many of the methods used to determine choice candidates involve considerable computational complexity and/or require detailed assumptions about the mechanisms behind route choice (Bovy, 2009). Furthermore, if route choice is assumed to be affected by multiple route characteristics (e.g. travel time, fuel consumption, or scenery along the route), corresponding data are needed. As a consequence, it is difficult to apply these approaches in large-scale models with many origins and destinations and, potentially, missing data. Since invasion models often consider many origins and destinations that are distributed over large areas, alternative methods are needed to compute route choice sets for large-scale propagule transport models.

A potential alternative to modelling the mechanisms behind route choice explicitly is to understand route choice as a multi-scale process with different factors affecting choices on different spatial scales. Consider an individual driving from origin A to destination B via some intermediate destination C. For example, this intermediate destination could be a scenic road section, a city of interest, or even the home of a friend or relative. In general, it will be difficult to know intermediate destinations of an individual. However, the route to and from an intermediate destination will likely be determined by simpler, somewhat

economic factors, such as travel time. Therefore, the chosen route may be optimal on a local scale, whereas unknown factors may affect the route on the large scale. Note that this principle also applies to routes resulting from different reasoning applied on different scales. For example, an individual may seek to minimize the overall fuel consumption by taking the shortest route. Nonetheless, they may take main roads through towns and villages rather than neighbourhood roads even if the latter may be on the shortest path.

So far, locally optimal routes have been considered in the context of route planning (Abraham et al., 2013; Delling et al., 2015; Luxen and Schieferdecker, 2015). Many mapping tools, such as Google Maps or Bing Maps, suggest to users multiple routes to a destination. To compute such routes, Abraham et al. (2013) have developed an algorithm that efficiently determines few, "good" locally optimal paths between a single origin and a single destination. This approach meets the needs of route planning software, where computational speed is valued more highly than an exhaustive search. Route choice models, however, need to consider *all* admissible routes between *many* origins and destinations. Therefore, the existing algorithms are not suited to compute route choice sets for comprehensive route choice models, and new methods are needed. This problem will be addressed in this thesis.

1.3.4 Identifiability, estimability, and credibility of parameter values and predictions

Most models for biological invasions contain parameters that are not known a priori. To estimate these parameters and ensure that conclusions drawn from the models also hold in reality, models are often fitted to empirical data. A common approach is to choose the parameter values so that the discrepancy between empirical data and model predictions is minimized. If the residuals are assumed to be caused by stochastic processes, models can be fitted by maximizing the likelihood (Casella and Berger, 2002). Roughly speaking, the "likelihood" measures how likely collected data could be observed if a model were true.

If parameters are fitted to observations that are subject to random variations, the parameter estimates are random variables as well. To ensure that conclusions drawn from models are not due to chance but rather supported by empirical evidence, it is important to assess to what extent random influences could affect parameter estimates and predictions. Though maximum likelihood estimates are typically precise and highly credible if enough data are available, it is not always clear how many data would be required. Furthermore, there are situations in which parameters cannot be estimated precisely regardless of how many data are collected (Raue et al., 2009). Consequently, it is important to determine the credibility of parameter estimates and predictions before inference is drawn.

Two situations exist in which estimates are not credible: parameters may be non-identifiable or non-estimable (Raue et al., 2009). If the best-fitting model is not unique, i.e. multiple parameter choices fit the data equally well, the corresponding parameters are called non-identifiable. Typically, non-identifiability is thought of as inherent to the model and independent of the data (Jacquez and Greif, 1985). That is, even if the data set would be increased arbitrarily, the best-fitting estimator would not be unique. In contrast, parameters are said to be non-estimable if the data set does not suffice to obtain credible parameter estimates (Raue et al., 2009). Consequently, estimability depends on the desired level of credibility and the size of the data set.

The typical cause for identifiability and estimability issues is that models account for processes without a (major) effect or for multiple processes with similar effects. For example, errors between predictions and observations could be due to environmental stochasticity as well as measurement error. Consequently, it would be impossible to determine how much each of these processes contributes to observations. Though some models permit analytical investigation of identifiability and estimability issues, the interplay of processes can be complicated and oblique (Lele et al., 2010). Therefore, computational techniques are often the only way to detect these problems.

A widely used statistical tool to assess the credibility of estimates are confidence intervals. These intervals indicate under which range of parameters the collected data would be observable with reasonable chance if the model were correct (Casella and Berger, 2002). Small confidence intervals suggest that parameters are identifiable and estimable. If models are fitted by maximizing the likelihood, confidence intervals are often computed approximately via Wald's method. This method exploits asymptotic properties of the maximum likelihood estimator and fails or yields misleading results if these properties are not achieved approximately. Furthermore, Wald's method may be inaccurate if the maximized likelihood is close to but not *at* the maximum. This may happen if the likelihood is maximized numerically and the search is stopped based on misleading termination criteria. As a consequence, Wald's method may overestimate or underestimate confidence intervals strongly.

This problem can be solved by using other, more accurate methods to construct confidence intervals. One alternative is to use sampling-based techniques (Efron, 1981; Buckland, 1984; Ponciano et al., 2009). These methods are generally reliable, and some are also suited to detect estimability issues (Ponciano et al., 2009). At the same time, however, sampling-based methods require many evaluations of the likelihood function. Therefore, these techniques may not be sufficiently efficient if the likelihood function is difficult to compute or large data sets are considered.

A second possible approach is to construct confidence intervals based on the profile likelihood (Cox and Snell, 1989). The idea is to fixate a parameter of interest θ_0 at different values and, respectively, maximize the likelihood with respect to the remaining parameters. The confidence interval for θ_0 consists of the values θ_0 that admit a sufficiently high likelihood value.

The profile likelihood approach is computational demanding, as the likelihood must be maximized several times. However, methods exist to determine the end points of profile likelihood confidence intervals within a single optimization effort, making the process much more efficient (Venzon and Moolgavkar, 1988; Neale and Miller, 1997; Wu and Neale, 2012). Un-

fortunately, these algorithms often fail if the likelihood function has unfavourable properties, such as steep "cliffs", or if parameters are not estimable (see e.g. Ren and Xia, 2019). Therefore, an approach that combines a high success rate with computational efficiency would be desirable. Such a method will be presented in this thesis.

1.4 Zebra and quagga mussels

I will apply the methods developed in this thesis to the prevention of zebra and quagga mussel introductions to British Columbia. Below, I provide some background knowledge on the biology, spread, impact, and management of these mussels.

Zebra mussels (*Dreissena polymorpha*) and quagga mussels (*Dreissena rostriformis bugensis*) are two mollusc species native to the Ponto-Caspian region and sharing several traits, such as life history characteristics and habitat requirements (Karatayev et al., 2013). The species occur in both fresh and brackish water and have relatively large temperature tolerances, ranging between a minimum of 5-15°C required for spawning and a maximum of 30°C (Mills et al., 1996; Karatayev et al., 2013). Both species require water conditions with low pH values and high calcium concentrations (Ramcharan et al., 1992; Jones and Ricciardi, 2005).

Zebra and quagga mussels have a life cycle with a veliger, juvenile, and adult stage (Karatayev et al., 2013). During its 2-4 year life span, a female zebra mussel can release up to $350,000$ eggs per productive season (Stoeckel et al., 2004); the veligers hatching from the eggs settle to some hard surface after few weeks in open water and enter the juvenile stage (Ackerman et al., 1994). Juvenile zebra mussels mature to fertile adults after reaching a size of 8-10 mm after few months (Mackie, 1991).

Zebra and quagga mussels have received particular attention due to their role as invasive species in Europe and North America (Karatayev et al., 2013). While zebra and quagga mussels have an invasion history dating back to the early 1800s in Europe, their introduction to North America, presumably via ballast water of ships, was in the mid 1980s (Karatayev

et al., 2015b). Since then, zebra and quagga mussels have invaded large parts of the United States and Canada (USGS, 2019). Zebra mussels spread and reproduce more quickly than quagga mussels but are often outcompeted by the latter in the long run (Karatayev et al., 2015b). The natural spread mechanism of dreissenid mussels is mainly through water flow but has been aided by inland vessel traffic (Karatayev et al., 2013). Intra-continental long-range spread is facilitated by the traffic of boaters transporting watercraft and gear from invaded to uninvaded waterbodies (Johnson et al., 2001).

Though a multi-layer model considering multiple spread mechanisms exists for zebra mussels (Mari et al., 2011), most models for the invasion of dreissenid mussels focus on boater traffic (Padilla et al., 1996; Bossenbroek et al., 2001; Leung et al., 2004; Bossenbroek et al., 2007), which is their main long-distance vector in North America. Models for habitat suitability focus mainly on the calcium concentration and the pH value of lakes (Ramcharan et al., 1992; Whittier et al., 2008; Karatayev et al., 2015a). Joint models for spread and establishment of zebra mussels exist for lakes in the area surrounding the Great Lakes (Bossenbroek et al., 2001; Leung et al., 2004; Leung and Mandrak, 2007). These models were fitted to historical invasion data. Bossenbroek et al. (2001) modelled the infestation probability of a lake as a linear function of the yearly number of incoming infested boaters, with one boat per year corresponding to an invasion probability of 0.000041. Leung et al. (2004) and Leung and Mandrak (2007) used models with better mechanistic justification and accounted for the Allee effect. However, because they fitted submodels for propagule transport and establishment simultaneously or used only relative travel estimates, their fitted establishment models cannot be used with different propagule transport models. Though few models consider the spread of quagga mussels explicitly, the similarities between zebra and quagga mussels (Karatayev et al., 2013) suggest that many models for zebra mussel spread can also be applied to model the spread of quagga mussels.

Zebra and quagga mussels are considered ecosystem engineers and can have a variety of significant ecological effects (Karatayev et al., 2015b). As filter feeders, dreissenid mussels

decrease the plankton level in lakes, thereby depleting resources available to native competitors and increasing water clarity, thus inducing system-wide effects benefiting littoral food webs and withering littoral food webs (Strayer, 2009; Karatayev et al., 2015b). Some studies suggest that dreissenid mussels facilitate algal toxin production through selective feeding (Knoll et al., 2008; Pick, 2016), but scientific evidence is not clear (Karatayev et al., 2015b), and quagga mussels have also been suggested as biocontrol for harmful algal blooms (Waajen et al., 2016).

Similar to the ecological effects, the economic impacts of zebra and quagga mussels are versatile and far-reaching. Impacts include clogging of water intake pipes of freshwater supply systems and power plant cooling systems, boat fouling, and loss of touristic and recreational value of lakes (Connelly et al., 2007; Rosaen et al., 2012). Quantitative cost estimates are difficult but range in the order of billions of US dollars in yearly costs to the US economy alone (Pimentel et al., 2005; Rosaen et al., 2012).

To counteract the inland spread of zebra and quagga mussels, several American states and Canadian provinces set up inspection stations at road sides, where transported watercraft are inspected for invasive mussels (Mangin, 2011; Alberta Environment and Parks Fish and Wildlife Policy, 2015; Inter-Ministry Invasive Species Working Group, 2015). If an inspected watercraft has a high potential of being infested, it is decontaminated and, if necessary, put to quarantine (BC Ministry of Environment and Climate Change Strategy, 2019). In British Columbia, 12 inspection stations were operated on a yearly budget of 3.75 million Canadian Dollars in 2019. Thereby, BC was assumed to be uninvaded. Though eradication of dreissenid mussels has been reported successful in some instances, eradication is costly and possible only if the mussel population is sufficiently contained and lake ecology and usage admit the application of control (Wimbush et al., 2009; Chakraborti et al., 2013; Lund et al., 2018).

Several studies have developed methods to optimize the management of dreissenid mussels (Leung et al., 2002; Potapov and Lewis, 2008; Potapov et al., 2008; Potapov, 2008;

Vander Zanden and Olden, 2008; Finnoff et al., 2010). For example, Leung et al. (2002) emphasize the benefits of invasion prevention. Potapov and Lewis (2008) determine the optimal configuration of inspection stations in spatially explicit lake systems, thereby highlighting the importance of controlling connections between clusters of lakes. Furthermore, contrasting the options to inspect watercraft leaving invaded waterbodies as opposed to those arriving at uninvaded lakes, Potapov et al. (2008) show that either strategy can be optimal dependent on the progression of the invasion but never both. Finally, in a qualitative study, Vander Zanden and Olden (2008) provide a conceptual framework to determine lakes that should be prioritized for early detection and rapid response measures.

1.5 Thesis overview

In this thesis, I will address major challenges involved with modelling and controlling the transport of invasive species spreading by means of human road traffic. In particular, I will develop a method to determine route choice sets for large-scale traffic models, apply the results to build a hybrid gravity and route choice model for vector traffic, and use the model output to optimize road-side control measures. Finally, I will introduce a robust and efficient method to determine profile likelihood confidence intervals. I will demonstrate the benefits of the developed techniques by applying them to the control of the potential invasion of dreissenid mussels to BC. Below I provide an overview of the chapters of the thesis.

In the second chapter, I develop a novel algorithm to determine route choice sets for large-scale traffic models. The approach focuses on locally optimal routes and adds on to existing methods by performing an efficient and exhaustive search for routes between many origins and destinations. I apply the algorithm to the road network in BC and investigate the impact of model parameters on the results and the efficiency of the algorithm.

In the third chapter, I derive a hybrid gravity and route choice model to assess vector traffic in large-scale road networks. The model involves four hierarchies accounting for agents' travel incentive, route choice, travel timing, and compliance with surveys. As a consequence,

the model can be fitted to data collected in road-side surveys. I develop methods to validate model assumptions rigorously and to overcome computational challenges involved with fitting the model. I apply the model to assess boater traffic to BC, thereby identifying the most significant origins and destinations of boaters and the most frequently used roads. The results can be used to inform control measures targeting incoming boaters and to assess the invasion risk of British Columbian lakes.

In the fourth chapter, I present a method to optimize road-side vector control. Framing the problem specific to aquatic invasive species management, I show how linear integer programming techniques can be used to optimize placement and operating times of watercraft inspection stations. I apply the method to dreissenid mussel control in BC, whereby I utilize the traffic estimates from the third chapter. I consider different management scenarios and investigate how changes in management constraints and model uncertainty affect the optimal policy.

My fifth chapter will be devoted to developing a robust and efficient method to compute profile likelihood confidence intervals. The motivation for this chapter is to compute confidence intervals for the parameters estimated in chapter 3. I build on a classic algorithm for this task (Venzon and Moolgavkar, 1988) and introduce several extensions increasing both the efficiency and the robustness of the algorithm. I evaluate the performance of the new algorithm in comparison to several existing methods by applying the algorithms to benchmark problems.

In the sixth and last chapter of this thesis, I highlight the significance of this thesis for – and beyond – invasive species management and suggest extensions for the presented methods.

Chapter 2

Locally optimal routes for route choice sets

2.1 Introduction

Route choice models have important applications in transportation network planning (Yang and Bell, 1998), traffic control (Mahmassani, 2001), and even epidemiology and ecology, as will become apparent later in this thesis. Route choice models can be classified as either perfect rationality models or bounded rationality models. In perfect rationality models (Sheffi, 1984), travellers are assumed to have complete information and choose their routes optimally according to some goodness criterion, whereas bounded rationality models (Simon, 1957) take information constraints and the complexity of the optimization process into account. Though both perfect rationality models and bounded rationality models have been used in route choice modelling, bounded rationality models have been found to fit observed data better (Di and Liu, 2016).

Many bounded rationality models consider route choice as a two-stage process: first, a so-called "choice set" of potentially good routes is generated, and second, a route from the choice set is chosen according to some goodness measure (Ben-Akiva et al., 1984). This approach is motivated through travellers' limited ability to consider all possible paths. Instead, they may heuristically identify a small set of routes from which they choose the seemingly best. Besides this conceptual reasoning, the two-step model has computational advantages, as the choice sets can be generated based on simple heuristics, while complex models may be applied to

determine travellers' preferences for the identified routes. Therefore, the two-stage process is widely used in route choice modelling (Prato, 2009).

Most of the approaches to identify route choice sets are based on a combination of the optimality assumption, the constraint assumption, and the stochasticity assumption.

- According to the optimality assumption, travellers choose routes optimally according to some criterion, which could be based on route characteristics (e.g. travel costs and travel time), or on scenarios (e.g. that the travel time on the shortest route increases). Examples include the link labelling approach (Ben-Akiva et al., 1984), link elimination (Azevedo et al., 1993), and link penalty (De La Barra et al., 1993).

- According to the constraint assumption, travellers consider all paths whose quality exceeds a certain minimal value (e.g. acyclic paths not more than 25% longer than the shortest route). This assumption motivates constrained enumeration methods (Prato and Bekhor, 2006).

- The stochasticity assumption accounts for the possibility of stochastic fluctuations of route characteristics (e.g. through traffic jams or accidents) or error-prone information. Often, stochastic route choice sets are computed based on the optimality principle applied to a randomly perturbed graph (see Bovy, 2009).

Though each of the assumptions mentioned above has a sound mechanistic justification, they require that the heuristic that travellers use to identify potentially suitable paths is known and that corresponding data are available. However, if travellers choose a route for unknown reasons, e.g. because they desire to drive via some intermediate destination, their routes would be hard to consider with the common methods. The natural solution would be to increase the set of generated routes by relaxing constraints or modelling more mechanisms explicitly. However, in comprehensive and large-scale route choice models, many origin-destination pairs may have to be considered, making it costly or even infeasible to work with large choice sets. Thus, it would be desirable to characterize choice sets based on a more

general but sufficiently restrictive criterion that does not require knowledge or data of the specific mechanism behind route choices.

A potentially suitable criterion is *local* optimality. A route is locally optimal if all its short ("local") subsections are optimal, respectively, according to a given measure. For example, if travel time is the applied goodness criterion, a locally optimal route would not contain local detours.

The rationale behind the principle of local optimality is that the factors impacting travellers' routing decision may differ dependent on the spatial scale. Tourists, for example, may want to drive along the shortest route locally but plan their trip globally to include a number of sights. Other travellers may want to drive along the quickest routes locally while minimizing the overall fuel consumption. Yet others may have a limited horizon of perfect information and act rationally within this horizon only. Independent of the specific mechanism behind travellers' route choice on the large scale, it is possible to characterize many choice candidates as locally optimal routes.

A potential problem with considering locally optimal routes is that the set of locally optimal routes between an origin and a destination can be very large and include zig-zag routes, which may seem unnatural. A possible solution is to focus on so-called *single-via paths*. A single-via path is the shortest path via a given intermediate location.

Since not all locally optimal paths are single-via paths, restricting the focus on single-via paths excludes some potentially suitable paths from the choice set. However, single-via paths have a reasonable mechanistic justification through travellers choosing intermediate destinations, and the reduced choice sets are likely to include most of the routes that travellers would reasonably choose. Since the reduced sets contain relatively few elements, sophisticated models can be used for the second decision stage, in which a route is chosen from the choice set. Therefore, constraining the search for locally optimal routes on single-via paths may lead to overall better fitting route choice models.

To date, methods identifying locally optimal single-via paths have been developed with the objective to suggest multiple routes to travellers (Abraham et al., 2013; Delling et al., 2015; Luxen and Schieferdecker, 2015; Bast et al., 2016). Such suggestions of alternative routes are a common feature in routing software, such as Google Maps or Bing Maps. However, route choice models have different demands than routing software, as travellers' decisions shall be *modelled* or *predicted* rather than *facilitated*.

Route planning software seeks to compute a small number of high-quality paths that travellers may want to choose. Thereby, computational speed is more important than rigorous application of specific criteria characterizing the returned paths. In contrast, route choice models should consider *all* routes that travellers may take, and rigorous application of modelling assumptions is key to allow mechanistic inference and to make models portable. In addition, route choice models may consider *multiple* origins and destinations. Therefore, many algorithms designed to facilitate route planning cannot be directly applied to identify route choice sets.

In this paper, we bridge this gap by extending an algorithm originally designed for route planning. The algorithm REV by Abraham et al. (2013) searches a small number of "good" locally optimal paths between a single origin-destination pair. Thereby, the algorithm uses an approximation causing some locally optimal paths being misclassified as suboptimal.

Our extended algorithm overcomes these limitations. Unlike REV, our algorithm returns (almost) *all* admissible paths between a *set* of origins and a *set* of destinations. Therefore, we call our algorithm REVC, the "C" emphasizing the attempted **c**omplete search. REVC identifies locally optimal routes with arbitrarily high precision. That is, the algorithm may falsely reject some locally optimal routes, but the error can be arbitrarily reduced by cost of computational speed. As the execution time of REVC depends mostly on the number of distinct origins and destinations rather than the number of origin-destination *pairs*, the algorithm is an effective tool to build traffic models on comprehensive scales.

This paper is structured as follows: first, we introduce helpful definitions and notation, review concepts we build on, and provide a clear definition of our goal. Then we give an overview of REVC, before we decribe each step in detail. After describing the algorithm, we present test results proving the algorithm's applicability and efficiency in real-world problems. Finally, we discuss the test results and the limitations and benefits of our approach.

2.2 Algorithm

2.2.1 Preliminaries

In this section, we specify our goal and introduce helpful notation and concepts. First, we provide definitions and notation, which we then use to characterize the routes we are seeking. Afterwards, we recapitulate Dijkstra's algorithm and briefly describe the method of reach based pruning, two basic concepts that our work builds on.

2.2.1.1 Problem statement and notation

Suppose we are given a graph $G = (V, E)$ that represents a road network. The set of vertices V models intersections of roads as well as the start and end points of interest. The directed edges $e \in E$ represent the roads of the road network and are assigned non-negative weights c_e, denoting the costs for driving along the roads. To ease notation, we will refer to the cost of an edge or path as its *length* without loss of generality. In practice, other cost metrics, such as travel time, may be used. Our goal is to find locally optimal paths between all combinations of origin locations $s \in O \subseteq V$ and destination locations $t \in D \subseteq V$.

To specify the desired paths more precisely, we introduce convenient notation and make some definitions:

$d(u, v)$ is the length of the shortest path from the vertex u to the vertex v in the considered graph.

$d_P(u, v)$ is the length of the subpath of P from vertex u to vertex v.

$l(P)$ is the length of the path P. That is, $l(P) = \sum_{e \in P} c_e$.

$P_{sv_1v_2,\ldots v_kt}$ is the shortest path from s to t via vertices $v_1, \ldots, v_k$ in the given order. That is, P_{st} is the shortest path from s to t, $P_{sv_1v_2,\ldots,v_kt} = P_{sv_1} \cup P_{v_1v_2} \cup \cdots \cup P_{v_{k-1}v_k} \cup P_{v_kt}$ and $l(P_{sv_1v_2,\ldots v_kt}) = d(s, v_1) + d(v_1, v_2) + \cdots + d(v_{k-1}, v_k) + d(v_k, t)$. For simplicity, we assume that $P_{sv_1v_2,\ldots v_kt}$ is always uniquely defined. In practise, $P_{sv_1v_2,\ldots v_kt}$ is the concatenation of shortest paths found by algorithms outlined below, which are responsible for breaking ties.

P^{uv} is the subpath of P from $u \in P$ to $v \in P$.

With this notation, we introduce the notions of *single-via path*s.

Definition 2.1. A *single-via path* (or short *v-path*) P_{svt} via a vertex v is the shortest path from a vertex s to a vertex t via v. We say, v *represents* the single-via path P_{svt} with respect to the origin-destination pair (s, t).

We proceed with a precise definition of local optimality. Generally speaking, a path is T-locally optimal if each subpath of P with a length of at most T is a shortest path. However, because paths are concatenations of discrete elements, we need a more technical definition.

Definition 2.2. Consider a subpath $P' \subseteq P$ and let $P'' \subset P'$ be P' after removal of its end points. We say P' is a T-*significant* subpath of P if $l(P'') < T$. A path P is T-*locally optimal* if all its T-significant subpaths P' are shortest paths. We say P is α-*relative locally optimal* if it is T-locally optimal with $T = \alpha \cdot l(P)$.

We want to identify locally optimal paths between many origin and destination locations. However, there may be an excessive number of such paths. Therefore, we apply slightly stronger constraints on the searched paths, which we call *admissible* below.

Definition 2.3. Let $\alpha \in (0, 1]$ and $\beta \geq 1$ be constants. A v-path P_{svt} from vertex $s \in O$ to vertex $t \in D$ via vertex $v \in V$ is called *admissible* if

1. P_{svt} is α-relative locally optimal.

2. P_{svt} is longer than the shortest path by no more than factor β, i.e. $l(P_{svt}) \leq \beta \cdot l(P_{st})$.

Objective. The objective of this paper is to identify (close to) all admissible single-via paths between each origin $s \in O$ and each destination $t \in D$.

2.2.1.2 Dijkstra's algorithm

Large parts of our algorithm are based on modifications of Dijkstra's algorithm (Dijkstra, 1959; Dantzig, 1998). Dijkstra's algorithm is a frequently used method to find the shortest paths from an origin s to all other vertices in a graph with non-negative edge weights. Though the algorithm is well-known to a large audience, we briefly recapitulate the algorithm to establish some notation that we will use later.

- In Dijkstra's algorithm, every vertex v is assigned a specific cost denoted $\text{cost}(v)$. Eventually, this cost shall be equal to the distance between the origin vertex s and vertex v. Initially, however, the cost of each vertex is ∞. An exception is the origin s, for which the initial cost is 0.

- We say that a vertex v is *scanned* if we are certain that $\text{cost}(v) = d(s, v)$. Furthermore, we say that a not yet scanned vertex v is *labelled* if $\text{cost}(v) < \infty$. All other vertices are called *unreached*. In line with our notion of scanned vertices, we call edges $e = (u, v)$ scanned if we know that $e \in P_{sv}$ for some scanned vertex v.

Dijkstra's algorithm is outlined in Algorithm 2.1. Initially, all vertices are in a container that allows us to determine the least-cost vertex efficiently. Dijkstra's algorithm consecutively removes the least-cost vertex v from the container and scans it. That is, the algorithm iterates over v's successors w and updates their costs if the distance from the origin s to w via v is smaller than the current cost of w. In this case, v is saved as the *parent* of w.

After execution of Dijkstra's algorithm, shortest paths can be reconstructed by following the trace of the computed parent vertices, starting at the destination vertex and ending at

Algorithm 2.1: Dijkstra's algorithm.

```
1 while container is not empty do
2     Take the vertex with the lowest cost from the container and remove it;
3     Scan the vertex v:
4         forall successors of v that have not been scanned yet do
5             Label w:
6                 if cost(w) < cost(v) + c_vw then
7                     Set cost(w) := cost(v) + c_vw ;      // c_vw is the length of the
                       edge from v to w
8                     Set parent(w) := v;
```

the origin. The edges $(\text{parent}(v), v)$ for all scanned vertices $v \in V$ form a *shortest path tree*. Hence, we call the procedure described above "growing a shortest path tree". The distance from the start vertex to its farthest descendant is called the *height* of the shortest path tree. As we will see below, it can be beneficial to stop the tree growth when the tree has reached a certain height.

When the shortest path between a specific pair of vertices s and t is sought, the *bidirectional Dijkstra algorithm* is more efficient than the classic algorithm (compare Figures 2.1 (a) and (b)). The bidirectional Dijkstra algorithm grows two shortest path trees: one in forward direction starting at the origin s and one in backward direction starting at the destination t. The trees are grown simultaneously; i.e., the respective tree with smaller height is grown until its height exceeds the other tree's height. The search terminates if a vertex v is included in both trees, i.e., scanned from both directions. The shortest path is the concatenation of the s-v path in the first shortest path tree and the v-t path in the second tree.

2.2.1.3 Reach-based pruning

Dijkstra's algorithm is not efficient enough to find shortest paths in large networks within reasonable time. Therefore, multiple methods have been developed to identify and prune

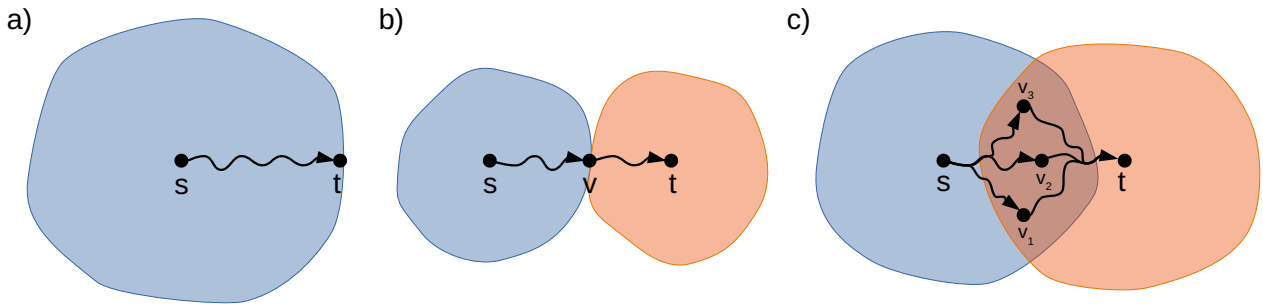


Figure 2.1: Conceptual illustration of different path search algorithms for an origin s and a destination t. The shaded areas depict shortest path trees. (a) Dijkstra's algorithm grows a single shortest path tree around the origin until the destination is reached. (b) The bidirectional Dijkstra algorithm grows a forward tree around the origin and a backward tree around the destination until the two shortest path trees meet at a vertex v. (c) Multiple v-paths can be constructed by growing overlapping shortest path trees around origin and destination.
Figures (a) and (b) are redrawn from Bast et al. (2016).

vertices that cannot be on the shortest path. One of these approaches is reach-based pruning (RE; Goldberg et al., 2006), which we introduce below.

Let us start by introducing the notion of a vertex's reach.

Definition 2.4. The *reach* of a vertex v is defined as

$$\text{reach}(v) \quad := \quad \max_{u,w \in V \,:\, v \in P_{uw}} \{\min\left(d(u,v),\, d(v,w)\right)\}\,. \tag{2.1}$$

That is, if we consider all shortest paths that include v, split each of these paths at v, and consider the shorter of the two ends, then the reach of v is the maximal length of these sections. The reach of v is high if v is at the centre of a long shortest path. Typically, vertices on highways have a high reach, since many long shortest paths include highways.

Disregarding vertices with small reaches can speed up shortest paths searches. Suppose we use the bidirectional Dijkstra algorithm to find the shortest path between the vertices s and t and have already grown shortest path trees with heights h. Let $v \in P_{st}$ be a vertex that is located on the shortest path between s and t but has not been scanned yet. Then $d(s,v) > h$ and $d(v,t) > h$, since v would have been included in one of the shortest path trees otherwise. Therefore, we know that $\text{reach}(v) \geq \min\left(d(s,v),\, d(v,t)\right) > h$. Thus, when

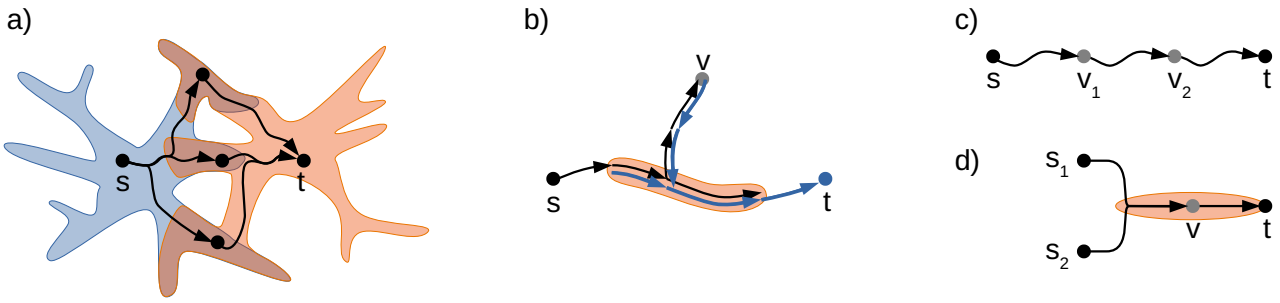


Figure 2.2: Optimizations that REVC employs to efficiently identify admissible v-paths between origin-destination pairs (s, t). (a) Shortest path trees (depicted as shaded areas) are grown to a tight bound only and exclude low-reach vertices, which cannot be on long locally optimal paths. (b) U-turn paths (e.g. $s \to v \to t$) are excluded by requiring that an edge adjacent to the via vertex is included both in the shortest path tree around the origin (black arrows) and the shortest path tree around the destination (blue arrows). Edges satisfying this constraint are highlighted with red background. Note that arrows with different directions depict distinct edges. (c) If v-paths via different vertices v_1 and v_2 are identical, only one of these vertices is chosen to represent the path. (d) If v-paths for different origin-destination pairs (here: (s_1,t) and (s_2,t)) are represented by the same via vertex v and share a subpath (highlighted red), the local optimality of this section is tested only once for all origin-destination pairs.

adding further vertices to our shortest path trees, we can neglect all vertices with a reach less or equal to h. This speeds up the shortest path search.

Computing the precise reaches of all vertices is expensive, as this would require an extremely large number of shortest path queries. However, Goldberg et al. (2006) developed an algorithm to compute upper bounds on vertices' reaches efficiently. These upper bounds can be used in the same way as exact vertex reaches.

2.2.2 Outline of the algorithm

After specifying our goal and introducing necessary notation and concepts, we can now proceed with an overview of our algorithm. The main idea of REVC is (1) to grow shortest path trees in forward direction from all origins and in backward direction from all destinations and (2) to check the admissibility of the v-paths via the vertices that have been scanned in both forward and backward direction (see Figure 2.1a). For each vertex v that is scanned both from an origin s and a destination t, the v-path P_{svt} can be reconstructed easily from the information contained in the shortest path trees. Therefore, the only remaining step is to

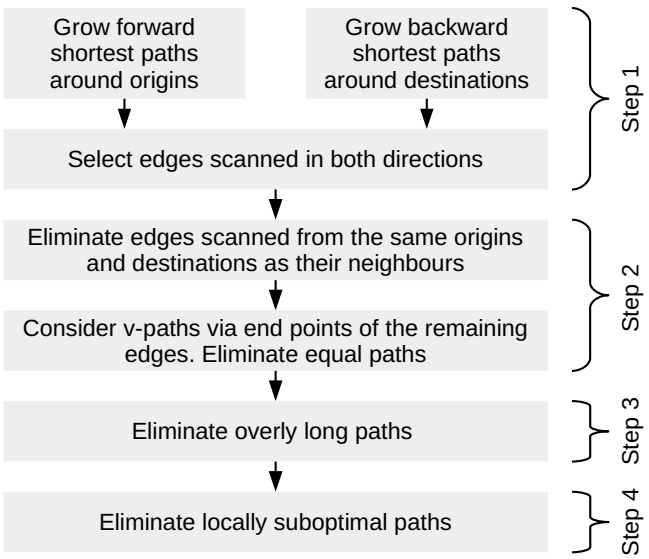


Figure 2.3: Overview of REVC.

check whether P_{svt} is admissible, i.e. locally optimal and not much longer than the shortest path P_{st}.

As each vertex $v \in V$ could serve as via vertex for many origin-destination combinations, checking the admissibility of all possible v-paths may be infeasible. Therefore, it is important to identify and exclude vertices that cannot represent admissible v-paths. The following observations can be exploited: (1) v-paths via vertices that are very far from an origin or destination cannot fulfill the length requirement. (2) Some vertices represent intersections of minor roads, which can be bypassed on close-by major roads. Thus, these vertices cannot be part of locally optimal paths. (3) Some v-paths may include a u-turn at the via vertex (see Figure 2.4). That is, travellers driving on such a path would need to drive back and forth along the same road. This is not locally optimal behaviour. (4) Some via vertices may represent the same v-paths. That is, the v-paths corresponding to distinct via vertices may be identical, and only one of these via vertices needs to be considered.

Our algorithm REVC makes use of the observations listed above. (1) When shortest path trees are grown around each origin and destination, the trees are grown up to a tightly specified height only. That way, many vertices that are too far off will not be scanned. (2) When the shortest path trees are grown, reach based pruning is applied to exclude

vertices that are not on any sufficiently locally optimal path (see Figure 2.2a). (3) Instead of considering all v-paths via vertices scanned in forward and backward direction, REVC considers only v-paths in which an *edge* adjacent to the via vertex has been scanned forward and backward. This excludes paths involving u-turns (see Figure 2.2). (4) Before checking the admissibility of the remaining v-paths, the algorithm ensures that each v-path is represented by one vertex only (see Figure 2.2c).

After these steps, REVC excludes v-paths that are exceedingly long and checks which v-paths are sufficiently locally optimal. Testing whether all v-paths P_{svt} via a specific vertex v are locally optimal would be expensive if each origin-destination pair $(s,t) \in O \times D$ were considered individually. Therefore, REVC checks the admissibility of many paths simultaneously, thereby reusing earlier results and applying approximations. That way, the algorithm becomes much more efficient than individual pair-wise searches for admissible paths (see Figure 2.2d). In Figure 2.3, we provide an overview of REVC.

Before the actual algorithm can be started, some preparational work and preprocessing are required. We will provide a detailed description of the preprocessing procedure after introducing the algorithm in detail.

2.2.3 Step 1: Growing shortest path trees

The algorithm REVC starts by growing forward shortest path trees out of each origin and backward shortest path trees into each destination. For each admissible v-path P, we need to scan at least one vertex v with $P = P_{svt}$ from both the origin s and the destination t. In addition, we want to scan one edge $e \in P$ adjacent to v from both directions if possible. These edges will be used to exclude u-turn paths. For each vertex v included in a shortest path tree, we note v's predecessor and height in the tree. Furthermore, we memorize from which origins and destinations each edge has been scanned.

2.2.3.1 Tree bound

To save the work of scanning vertices inadmissibly far away from the origins and destinations, we aim to stop the tree growth as soon as possible. We need to scan at least one vertex v for each admissible path P_{svt} with a length $l(P_{svt}) = d(s,v) + d(v,t) \leq \beta \cdot l(P_{st})$. Since either of $d(s,v)$ and $d(v,t)$ could be arbitrarily small, the algorithm REV by Abraham et al. (2013) grows the trees up to a height of $\beta \cdot l(P_{st})$. Nevertheless, we can terminate the search earlier if we take into account that we are searching for locally optimal paths.

To derive a tighter tree bound, note that for an α-relative locally optimal path P, each subsection with length $\alpha \cdot l(P)$ is a shortest path. This is in particular true for the subsection $P' \subseteq P$ starting at the origin. Since P' is a shortest path, the end point x_s of this subsection will be included in the origin's shortest path tree. Therefore, it suffices to grow the destination's shortest path tree until x_s is reached, which is closer to t than $\beta \cdot l(P_{st})$. The same applies in the reverse direction.

To specify the tree bound, define $x_s \in P$ more precisely to be the first vertex that is farther away from the origin than $\alpha \cdot l(P)$. If this vertex is located in the second half of the path, change x_s to be the last vertex in the first half of P. Choose x_t accordingly in relation to the destination. Our observations from above are formalized in the following lemma and corollary, which we prove in Appendix 2.A.

Lemma 2.1. *With s, t, x_s, x_t, and P defined as above, there is at least one vertex $v \in P$ with*

1. $d_P(s,v) = d(s,v) \leq d_P(s,x_t)$ *and*
2. $d_P(v,t) = d(v,t) \leq d_P(x_s,t)$.

Corollary 2.1. *For each admissible v-path between an origin-destination pair (s,t), a via vertex will be scanned from both directions if the shortest path trees are grown up to a height*

of

$$h_{\max} \;:=\; \max\left\{(1-\alpha)\,\beta l(P_{st}),\, \frac{1}{2}\beta l(P_{st})\right\}. \tag{2.2}$$

In Corollary 2.1, we consider a single origin-destination pair. However, we want to identify admissible paths between multiple origins and destinations and have to adjust the tree bound accordingly. The tree around each origin and destination shall be large enough to include via vertices for *all* paths starting at the respective endpoint. Hence, if we grow a tree out of origin s, we grow it to a height of $\max\left\{(1-\alpha)\,\beta M_s,\, \frac{1}{2}\beta M_s\right\}$ with $M_s = \max_{t\in D} l(P_{st})$. We proceed with destinations similarly.

Note that the tree bounds above can only be determined if the shortest distances between the origins and destinations are known. Though these distances can be determined while the shortest path trees are grown, we will see in the next section that the shortest distances can also be used to speed up the tree growth itself. Therefore, it is beneficial to determine the shortest distances in a preprocessing stage. This also makes it easy to grow the trees in parallel.

2.2.3.2 Pruning the trees

The search for admissible paths can be significantly sped up if vertices with small reach values are ignored when the shortest paths are grown. Consider a vertex v on an admissible s-t path P. Let us regard the subpath P' that is centred at v and has a length just greater than $\alpha \cdot l(P)$. Since P is α-relative locally optimal, we know that P' is a shortest path. Furthermore, P' is roughly split in half by v, unless v is close to one of the end points of P. Thus,

$$\text{reach}(v) \;\geq\; \min\left\{\frac{\alpha}{2} l(P), d(s,v), d(v,t)\right\} \tag{2.3}$$

(see Lemma 5.1 in Abraham et al., 2013).

If we are growing the tree out of origin s, we can use (2.3) to prune the successors of vertices v with $\text{reach}(v) < \min\left\{\frac{\alpha}{2} l(P), d(s,v)\right\}$. Pruning the successors but not v itself ensures that at least one vertex per admissible path is scanned from both directions, even if (2.3) is dominated by $d(v,t)$.

Since $l(P)$ is unknown when the shortest path trees are grown, the length of P must be bounded with known quantities. Abraham et al. (2013) use the triangle inequality

$$l(P) \;\geq\; d(s,v) + d(v,t) \;\geq\; \text{cost}(v). \tag{2.4}$$

However, we can also determine shortest distances before we search admissible paths and exploit that $P \geq d(s,t)$ or, if we are considering multiple origins and destinations, $l(P) \geq L_s := \min_{\tilde{t} \in D} d\big(s, \tilde{t}\big)$. Therefore, we may prune the successors of vertices v with

$$\text{reach}(v) < \min\left\{\text{cost}(v), \frac{\alpha}{2} \max\left\{\text{cost}(v), L_s\right\}\right\} \tag{2.5}$$

when we grow the shortest path tree out of origin s.

We can prune even more vertices if we grow the trees in forward and backward direction in separate steps. The idea is to use data collected in the first step to derive a sharper pruning bound for the second step. Whether we grow the forward or the backward trees in the first step depends on whether there are more destinations or more origins to process. Below we assume without loss of generality that we consider more destinations than origins, $|D| \geq |O|$.

We proceed as follows: we start by growing the forward trees out of the origins. In this phase, we prune vertices' successors according to inequality (2.5). After growing the forward trees, we determine for each scanned vertex v the distance $d_{\min}(v) := \min_{s \in O;\, v \text{ scanned from } s} d(s,v)$ to the closest origin it has been scanned from. If v has not been scanned, we set $d_{\min}(v) := \infty$. Now we grow the backward trees and use $d_{\min}(v)$ as a lower bound for $d(s,v)$ for all origins

Algorithm 2.2: Growing a forward shortest path tree out of origin s.

```
1  while container is not empty do
2      Take the vertex v with the lowest cost from the container and remove it;
3      Mark edge leading to v as visited from origin s;
4      Include v in the shortest path tree;
5      if d_min(v) > cost(v) then
6          d_min(v) := cost(v);
7      if reach(v) ≥ min(cost(v), (α/2) max(cost(v), L_s)) then
8          Scan the vertex v;                          // see Algorithm 2.1
```

Algorithm 2.3: Growing a forward shortest path into destination t.

```
1  while container is not empty do
2      Take the vertex v with the lowest cost from the container and remove it;
3      Mark edge leading to v as visited from destination t;
4      if reach(v) ≥ min(cost(v), (α/2) max(cost(v), L_t)) then
5          Include v in the shortest path tree;
6          Scan the vertex v with early pruning:
7              forall neighbors w of v that have not been scanned yet do
8                  NewCost := cost(v) + d(v, w);
9                  if reach(v) ≥ min(NewCost, (α/2) max(NewCost, L_t), d_min(v)) then
10                     Label w;                         // see Algorithm 2.1
```

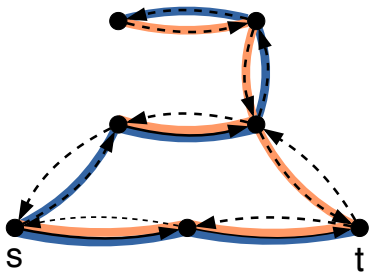


Figure 2.4: Advantages of considering via edges instead of via vertices. Arrows highlighted in dark blue depict the forward shortest path tree grown from the origin s, and arrows highlighted in light red represent the backward tree grown into the destination t. Edges that are scanned from both directions are potential via edges and drawn as solid black lines. The remaining edges are drawn as dashed black lines. All vertices are scanned both from s and t and would therefore considered potential via vertices. However, paths via the two topmost vertices would require a u-turn. Restricting the focus on v-paths via vertices adjacent to the solid lines excludes these u-turn paths.

$s \in O$. Hence, we can prune all vertices with

$$\text{reach}(v) < \min \left\{ \text{cost}(v), \frac{\alpha}{2} \max \left\{ \text{cost}(v), L_t \right\}, d_{\min}(v) \right\}. \tag{2.6}$$

In contrast to criterion (2.5), we can apply criterion (2.6) directly to each vertex v and not only to its successors. This decreases the number of considered vertices. We provide pseudo code for the tree growth procedures in Algorithms 2.2 and 2.3.

2.2.3.3 Determining potential via vertices

With the shortest path trees, we can determine which vertices may potentially represent admissible v-paths. Each vertex scanned in forward and backward direction could be such a via vertex. However, since some of the resulting paths could include u-turns, we consider the scanned *edges* rather than the vertices. This excludes paths with u-turns (see Figure 2.4).

We proceed as follows: we determine for each scanned edge e the sets O_e and D_e of origins and destinations that e has been scanned from. We discard all edges that have not been scanned from at least one origin and one destination. Let E_{via} be the resulting set of edges. The set of considered via vertices $V_{\text{via}} := \{v \in V \mid \exists w \in V : (v, w) \in E_{\text{via}}\}$ is given by the starting points of the edges in E_{via}.

Note that though the procedure above eliminates paths with u-turns, some admissible single-via paths may be rejected as well. However, this issue will rarely occur in realistic road networks, since the problem arises only at specific merging points of very long edges. We provide details in Appendix 2.B.

2.2.4 Step 2: Identifying vertices representing identical v-paths

Some of the vertices in V_{via} may represent identical v-paths. Since we want to save the effort of checking the admissibility of the same path multiple times and, similarly importantly, we do not want to return multiple identical paths, we need to ensure that each admissible path is represented by one via vertex only.

To identify vertices representing identical paths, we have to compare the v-paths corresponding to all $v \in V_{\text{via}}$ for each origin-destination pair. This requires $\mathcal{O}(|V_{\text{via}}|\,|O|\,|D|)$ steps. However, for some vertices, identical paths can be identified more quickly, as adjacent vertices typically represent similar sets of v-paths. Therefore, we proceed in two steps: first, we reduce V_{via} by eliminating vertices whose via paths are also represented by their respective neighbours, and second, we check which of the remaining vertices represent identical v-paths. Below we describe the two steps in greater detail.

2.2.4.1 Eliminating vertices that represent the same v-paths as their neighbours

The endpoints of an edge can be neglected as via vertices if the edge has been scanned from the same origins and destinations as a neighbouring edge. Consider for example an edge (v, w) that has been scanned from both an origin s and a destination t. Then $P_{sw} = P_{svw}$ and $P_{vt} = P_{vwt}$. It follows that v and w represent the same v-path with respect to (s, t): $P_{svt} = P_{swt}$. Now consider an adjacent edge (u, v) that has been scanned from s and t as well. Clearly, it is $P_{sut} = P_{svt}$ and $P_{svt} = P_{swt}$, which implies that the v-paths via u, v, and w are identical. Therefore, only one of these vertices has to be considered.

To introduce an algorithm that efficiently detects such configurations, let O_e be the set of origins and D_e the set of destinations that edge e has been scanned from. For each edge $e \in E_{\text{via}}$, we check whether one directly preceding edge $e' \in E_{\text{via}}$ has been scanned from a superset of origins and destinations, i.e. $O_e \subseteq O_{e'}$ and $D_e \subseteq D_{e'}$. If such an edge exists and one of the set inequalities holds strictly, i.e. $O_e \subset O_{e'}$ or $D_e \subset D_{e'}$, we may disregard edge e, as all v-paths via e are also v-paths via e'.

Things become more complicated if $O_e = O_{e'}$ and $D_e = D_{e'}$, as we may either reject e, e', or both edges. The latter case may occur if e' has another directly preceding edge $e'' \in E_{\text{via}}$ with $O_{e'} \subseteq O_{e''}$ and $D_{e'} \subseteq D_{e''}$. If one of these inequalities is strict, we disregard both e and e'. Otherwise, we continue traversing the edges in E_{via} until either (1) an edge is found whose origin and destination sets supersede the sets of all previous edges or (2) no further predecessor with sufficiently large origin and destination sets is found. In the second case, we may disregard all traversed edges but e. We apply the same approach to the successors of e and repeat this procedure until all edges in E_{via} have been processed.

The updated set V_{via} of via vertices consists of the starting vertices of the edges in the reduced edge set E_{via}. We provide pseudo code for the outlined algorithm in Algorithm 2.4. An efficient implementation may compare the origin and destination sets of the edges in E_{via} before the traverse is started. This makes it easy to implement the most expensive parts of the algorithm in parallel.

2.2.4.2 Eliminating identical v-paths

Using adjacency relationships to identify all vertices representing the same v-paths would involve a traverse over all edges in E_{via}. However, it is more efficient to identify similar v-paths by their lengths. To this end, we may assume that $P_{svt} = P_{swt}$ if and only if $l(P_{svt}) = l(P_{swt})$. Though it can happen that distinct paths have the same length, this case is usually not of greater concern in practical applications. The issue can be reduced by introducing a small

Algorithm 2.4: Eliminating vertices that represent the same v-paths as their neighbours.

```
1  Function has_superior_predecessor(e):
2      Remove e from E_via;
3      forall directly preceding edges e' of e do
4          if O_e ⊆ O_e' and D_e ⊆ D_e' then
5              if O_e = O_e' and D_e = D_e' then
6                  return has_superior_predecessor(e')
7              else
8                  return True;
9              return False;
10 Function has_superior_successor(e):
11     Remove e from E_via;
12     forall directly succeeding edges e' of e do
13         if O_e ⊆ O_e' and D_e ⊆ D_e' then
14             if O_e = O_e' and D_e = D_e' then
15                 return has_superior_successor(e')
16             else
17                 return True;
18             return False;
19 E'_via := ∅;
20 while E_via ≠ ∅ do
21     Set e := next entry in E'_via;
22     if not has_superior_predecessor(e) and not has_superior_successor(e)
        then
23         Add e to E'_via;
24 E_via := E'_via;
```

random perturbation for the lengths of edges. We examine this limitation further it in the discussion section.

With the above assumption, identical paths can be identified efficiently. Since for each origin-destination pair (s,t) and each potential via vertex $v \in V_{\text{via}}$ the distances $d(s,v)$ and $d(v,t)$ are known, the v-path lengths can be computed easily. For each origin-destination pair, a comparison of the lengths of the v-paths corresponding to all $v \in V_{\text{via}}$ can be conducted in linear average time with hash maps. Note that the path lengths must be compared with an appropriate tolerance for machine imprecision.

In later steps it will be of benefit if most v-paths are represented by a small set of via vertices. If there are multiple vertices representing the same v-paths, we therefore choose the via vertex v that has been scanned from the most origin-destination combinations $O_v \times D_v$. This makes it easier to reuse partial results when we check whether the v-paths are locally optimal.

2.2.5 Step 3: Excluding long paths

Before we check whether paths are sufficiently locally optimal, we exclude the paths that exceed the length allowance. That is, we disregard all paths P_{svt} with $l(P_{svt}) > \beta \cdot l(P_{st})$ with origin-destination pairs (s,t) and via vertices $v \in V_{\text{via}}$. Since this step involves a simple comparison only, it is computationally cheaper than identifying identical paths. Therefore, it is efficient to conduct this step just before identical paths are eliminated (section 2.2.4.2). This also reduces the memory required to store potentially admissible combinations (s,v,t) of origin-destination pairs and via vertices.

2.2.6 Step 4: Excluding locally suboptimal paths

The most challenging part of the search for admissible paths is to check whether paths are sufficiently locally optimal. To test whether a subpath is optimal, we need to find the shortest

alternative, which is computationally costly. Therefore, we apply an approximation to limit the number of necessary shortest path queries.

Our method generalizes the approximate local optimality test by Abraham et al. (2013). They noted that v-paths are concatenations of two optimal paths. Hence, v-paths are locally optimal everywhere except in a neighbourhood of the via vertex. More precisely, a v-path P_{svt} from s to t via v is guaranteed to be T-locally optimal everywhere except in the section that begins T distance units before v and ends T distance units after v. Therefore, Abraham et al. (2013) suggest to perform a shortest path query between the end points x and y of this section to check whether it is optimal. Abraham et al. (2013) call this procedure the T-test.

The T-test does not return false positives. That is, a path that is not T-locally optimal will never be misclassified as locally optimal. However, the T-test may return false negatives: paths that are T-locally optimal but not $2T$-locally optimal may be rejected. In modelling applications, a more precise local optimality test may be desired.

It is possible to increase the precision of the T-test. Instead of checking whether the whole potentially suboptimal subpath is optimal, we may test multiple subsections to gain a higher accuracy. While this procedure ensures that fewer admissible paths are falsely rejected, the gain in accuracy comes with an increase in computational costs. Therefore, it is desirable to use the results of earlier local optimality checks to test the admissibility of other paths.

There are two situations in which local optimality results can be reused. First, if a subsection of a path is found to be suboptimal, other paths that include this section can be rejected as well. Second, if a subpath of a path is found to be locally optimal, other paths including this subpath may be classified as locally optimal as well. That way, many paths can be processed all at once.

When reusing partial results, it is important to note that even though we require all paths to be α-relative locally optimal, the *absolute* lengths of the subsections that need to be optimal depend on how long the considered paths are. Therefore, paths must be considered in an order dependent on their lengths. We provide details below.

2.2.6.1 Preparation

Before we can start testing whether the remaining v-paths are locally optimal, a preparation step is needed to identify the subpaths that may be suboptimal and thus need to be assessed more closely. To reuse partial results efficiently, we furthermore need to determine subsections that different paths have in common. We describe the preparation procedure below.

We start by introducing helpful notation. Suppose we want to test whether the v-paths via vertex v are locally optimal. Let $\tilde{O} := \{s \in O \mid \exists t \in D : l(P_{svt}) \leq \beta \cdot l(P_{st})\}$ be the origins for which at least one destination can be reached via v without violating the length constraint. Let $\tilde{D}$ be defined accordingly for the destinations. Define $\tilde{D}_s := \left\{t \in \tilde{D} \mid l(P_{svt}) \leq \beta \cdot l(P_{st})\right\}$ as the set of destinations that can be reached from the origin s via v without violating the length constraint.

In the preparation step, we determine for each origin $s \in \tilde{O}$ the destination $t_s := \underset{t \in \tilde{D}_s}{\operatorname{argmax}} l(P_{svt})$ for which the potentially suboptimal section is longest. Furthermore, we search for the vertex $x_s := \underset{\tilde{x} \in P_{sv};\, d(\tilde{x},v) \geq \alpha l(P_{svt_s})}{\operatorname{argmin}} d(\tilde{x}, v)$, which is the last vertex on P_{sv} with $d(x_s, v) \geq \alpha \cdot l(P_{svt_s})$, and we determine x_t defined accordingly. Now we fill the arrays

$$A_{us} := \begin{cases} \text{True} & \text{if } u \in P_{sv} \\ \text{False} & \text{else,} \end{cases} \qquad A_{ut} := \begin{cases} \text{True} & \text{if } u \in P_{vt} \\ \text{False} & \text{else} \end{cases} \tag{2.7}$$

for all vertices $u \in P_{x_s v}$ and $u \in P_{v x_t}$, respectively.

The information saved in the shortest path trees are suitable to find paths from scanned vertices to the origins and destinations. However, the trees contain no information on the reverse paths starting at the end points. That is, while it is easy to find the backward shortest path from v to x_s, it is hard to follow the path in the opposite direction starting at x_s. We gather the necessary information in the preparation step: for each origin $s \in \tilde{O}$, we save the successors of each relevant vertex $u \in P_{sv}$.

Algorithm 2.5: Filling the array A for the origins and finding successors. The algorithm for the destinations is similar.

```
1  foreach destination s ∈ Õ do
2      t_s := argmax_{t∈D̃_s} (d(s, v) + d(v, t));
3      u := parent_s(v);
4      successor_s(u) := v;
5      stop := False;
6      while not stop do
7          if u ∉ A then
8              Initialize A_{us̃} := False for all s̃ ∈ Õ;
9          A_{us} := True;
10         successor_s(parent(u)) := u;
11         if d(v, u) > α (d(s, v) + d(v, t_s)) then
12             stop := True;
13         else
14             u := parent(u);
```

In Algorithm 2.5, we provide pseudo code for the described procedures. The pseudo-code considers the origins only. The algorithm for the destinations is similar. The preparation phase ends with sorting all origin-destination pairs with respect to the lengths of the respective v-paths via v.

2.2.6.2 Testing local optimality for one origin-destination pair

We use an approximation approach with flexible precision to check whether paths are locally optimal. For a parameter $\delta \in [1, 2]$, we call this procedure the T_δ-test. Thereby, δ is a measure for the test's precision.

To outline the T_δ-test, let us consider a v-path $P := P_{svt}$ from s to t via the vertex v. Let $S_s := \{u \in P_{sv} \,|\, d(u, v) < T\}$ be the set of vertices that are on the path P_{sv} and have a distance less than T to the vertex v. Furthermore, add to S_s the vertex $x := \underset{\tilde{x} \in P_{sv};\, d(\tilde{x}, v) \geq T}{\operatorname{argmin}} d(\tilde{x}, v)$ that

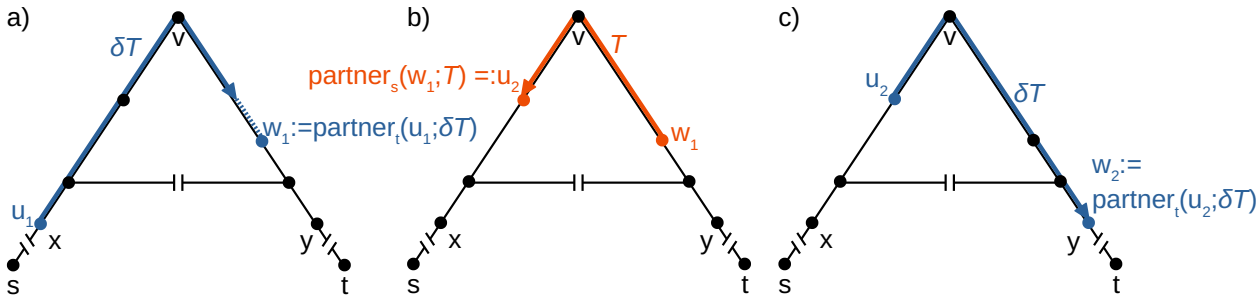


Figure 2.5: T_δ-test with $\delta = 1.4$. The three subfigures depict the steps of the T_δ-test for a path P_{svt} connecting origin-destination pair (s,t) via vertex v. The vertices x and y are the end points of the potentially locally suboptimal section. The edge lengths are given by the Euclidean distance except for the edges with an indicated gap. (a) In a first step, the test determines the vertex w_1 that is at least δT units along the path away from $u_1 := x$ (the distance is depicted as blue arrow). (b) If the shortest path query between u_1 and w_1 indicates that the subsection $P_{svt}^{u_1 w_1}$ is optimal, the test continues by determining the first vertex u_2 that is at least T units away from w_1 in backwards direction. (c) From u_2, the algorithm searches the vertex w_2 that is at least δT units along the path beyond u_2 and conducts a shortest path query between u_2 and w_2. If all the shortest path queries yield subpaths of P_{svt}, the path is deemed approximately T-locally optimal. Note that a T_2-test would have misclassified the path as not locally optimal, provided the shortest path from x to y includes the horizontal edge.

is closest to v but has $d(x,v) \geq T$ if such a vertex exists. Choose S_t accordingly with respect to the destination vertex t. Let $\mathrm{partner}_t(u;\tau) := \underset{\tilde{w}\in S_t;\, d_P(u,\tilde{w})\geq\tau}{\mathrm{argmin}} d_P(u,\tilde{w})$ for $u \in S_s$ be the vertex $w \in S_t$ that is closest to u but has $d_P(u,w) \geq \tau$. If no such vertex exists in S_t, set $\mathrm{partner}_t(u;\tau) = y := \underset{\tilde{w}\in S_t}{\mathrm{argmax}}\, d_P(u,\tilde{w})$. Define accordingly $\mathrm{partner}_s(w;\tau)$ for $w \in S_t$ as the vertex $u \in S_s$ that is closest to w but has $d_P(u,w) \geq \tau$.

The T_δ-test proceeds as follows: the algorithm starts at the vertex $u_1 := x$ and checks whether the subpath $P^{u_1 w_1}$ between u_1 and $w_1 := \mathrm{partner}_t(u_1;\delta T)$ is a shortest path. If so, the algorithm progresses searching $u_2 := \mathrm{partner}_s(u_1;T)$ in backward direction and repeats the steps formerly applied to u_1 now with u_2. This procedure repeats until $u_n = v$ for some $n \in \mathbb{N}$. If all the shortest path queries yield subpaths of P, the path is deemed approximately T-locally optimal. Otherwise, it is classified as not locally optimal. We depict the algorithm in Figure 2.5. We provide pseudo-code in Algorithm 2.6.

Similar to the T-test, the T_δ test does not return false positives. However, paths that are T-locally optimal but not δT-locally optimal might be rejected. Hence, the the T_1-test is exact, whereas the "classical" T-test by Abraham et al. (2013) is the T_2-test. An increase in precision comes with a computational cost. The T_δ-test requires at most $2\left\lceil \frac{1}{\delta-1} \right\rceil$ shortest

Algorithm 2.6: T_δ-test.

```
1  Search for the vertex x ∈ S_s with maximal distance to v;
2  Set u := x;
3  Set w := v;
4  while u ≠ v and w ≠ y do
5      Set w' := partner_t(u; δT);
6      if w = w' then
7          Set w := next farthest vertex to v in S_t;
8      else
9          Set w := w';
10     Check whether the u-w subpath is optimal
11         if d(u, w) < d(u, v) + d(v, w) then
12             return "Not locally optimal"
13     Set u' := partner_s(w; T);
14     if u = u' then
15         Set u := next closest vertex to v in S_s;
16     else
17         Set u := u';
18 return "Locally optimal"
```

path queries if $\delta > 1$. However, query numbers around $\frac{1}{\delta-1}$ are more common. Either way, the number of required queries is bounded by a constant independent of the graph, unless $\delta = 1$.

2.2.6.3 Using test results to check local optimality for multiple origin-destination pairs

The T_δ-test is a suitable procedure to check whether a single v-path is locally optimal. However, if many v-paths shall be tested, the required number of shortest path queries may exceed a feasible limit. Therefore, we show below how negative test results can be used to

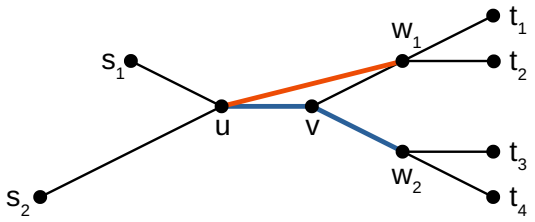


Figure 2.6: Accepting and rejecting multiple paths at once. Suppose we want to check the admissibility of the paths from the origins s_i to the destinations t_j via the vertex v. Suppose that we start with the path $P_{s_1vt_2}$ from s_1 to t_2 via v and find that the subsection P_{uvw_1} is not optimal, because there is a shorter path (orange) from u to w_1. Then we know that the paths $P_{s_1vt_1}$, $P_{s_2vt_1}$, and $P_{s_2vt_2}$ are not sufficiently locally optimal, either. Now suppose we continue with the pair (s_1, t_3) and find that $P_{s_1vt_3}$ is locally optimal because the section P_{uvw_2} (blue) is optimal. Since $P_{s_1vt_4}$ includes this subsection, too, and is not much longer than $P_{s_1vt_3}$, we can deduce that $P_{s_1vt_4}$ is approximately locally optimal as well.

reject multiple paths at once. Afterwards we describe a method to use positive test results to classify many paths as locally optimal.

2.2.6.3.1 Rejecting paths

Suppose that in order to test whether P_{svt} is admissible, we checked whether the subpath P_{svt}^{uw} between some vertices u and w is a shortest path, and suppose we obtained a negative result, i.e. found that $d(u,w) < d(u,v) + d(v,w)$. We can not only conclude that the path P_{svt} is not locally optimal but also reject other v-paths that include the subpath P_{svt}^{uw} (see Figure 2.6).

To see which paths can be rejected, let $\Omega_u := \{\tilde{s} \in O \,|\, d(\tilde{s},v) = l(P_{\tilde{s}uv})\}$ be the set of origins for which u is on the shortest path to v and define $\Delta_w := \{\tilde{t} \in D \,|\, d(v,\tilde{t}) = l(P_{vw\tilde{t}})\}$ accordingly for the destinations. Let furthermore $\mathcal{P} := \left\{(s,t) \in \tilde{O} \times \tilde{D} \,|\, l(P_{svt}) \leq \beta \cdot l(P_{st})\right\}$ be the set of all origin-destination pairs with a potentially admissible v-path via v, and let $\mathcal{P}_{uw} := \mathcal{P} \cap (\Omega_u \times \Delta_w)$ denote the respective set of origin-destination pairs for which the v-path via v also includes u and w. The following lemma shows which paths can be rejected as approximately inadmissible.

Lemma 2.2. *Suppose the* T_δ*-test is applied to check whether a path* P_{svt} *is* α*-relative locally optimal and that the test fails, because* $d(u,w) < d(u,v) + d(v,w)$ *for some vertices* u *and*

w. Then, for each pair $(\tilde{s}, \tilde{t}) \in \mathcal{P}_{uw}$ *with* $P_{\tilde{s}v\tilde{t}} \geq l(P_{svt})$*, the v-path* $P_{\tilde{s}v\tilde{t}}$ *is not relative locally optimal with a factor higher than* $\alpha_{\tilde{s}v\tilde{t}} < \frac{l(P_{xvy})}{l(P_{svt})} \leq \alpha\delta$*, whereby x and y are the neighbours of u and w in direction of v, respectively.*

Proof. By construction of $\mathcal{P}_{uw}$, it is $P_{xvy} \subseteq P_{\tilde{s}v\tilde{t}}$ for any origin-destination pair $(\tilde{s}, \tilde{t}) \in \mathcal{P}_{uw}$. Therefore, $P_{\tilde{s}v\tilde{t}}$ is at most T-locally optimal with $T < l(P_{xvy})$. Hence, the local optimality factor $\alpha_{\tilde{s}v\tilde{t}}$ for $P_{\tilde{s}v\tilde{t}}$ satisfies

$$\alpha_{\tilde{s}v\tilde{t}} = \frac{T}{l(P_{\tilde{s}v\tilde{t}})} < \frac{l(P_{xvy})}{l(P_{\tilde{s}v\tilde{t}})} \leq \frac{l(P_{xvy})}{l(P_{svt})} \leq \frac{\alpha\delta l(P_{svt})}{l(P_{svt})} = \alpha\delta. \tag{2.8}$$

□

Following Lemma 2.2, we can reject all pairs $(\tilde{s}, \tilde{t}) \in \mathcal{P}_{uw}$ with $P_{\tilde{s}v\tilde{t}} \geq l(P_{svt})$. The origin-destination pairs in question can be determined by considering the array A constructed in the preparation phase (equation (2.7)). Let $\tilde{A}_u := \left\{ s \in \tilde{O} \,|\, A_{us} = \text{True} \right\}$ and $\tilde{A}_w := \left\{ t \in \tilde{D} \,|\, A_{wt} = \text{True} \right\}$. Then, $\mathcal{A}_{uw} := \tilde{A}_u \times \tilde{A}_w \subseteq \mathcal{P}_{uw}$, and $\mathcal{P}_{uw} \backslash \mathcal{A}_{uw}$ contains only pairs $(\tilde{s}, \tilde{t})$ with $l(P_{\tilde{s}v\tilde{t}}) < l(P_{svt})$. It follows that all pairs $(\tilde{s}, \tilde{t}) \in \mathcal{P}_{uw}$ with $P_{\tilde{s}v\tilde{t}} \geq l(P_{svt})$ are also in $\mathcal{A}_{uw}$.

As $\mathcal{A}_{uw}$ may also contain pairs $(\tilde{s}, \tilde{t})$ with $l(P_{\tilde{s}v\tilde{t}}) < l(P_{svt})$, we process the origin-destination pairs in the order of increasing via-path length. Then the pairs $(\tilde{s}, \tilde{t}) \in \mathcal{A}_{uw}$ with $l(P_{\tilde{s}v\tilde{t}}) < l(P_{svt})$ will be processed before (s, t). If we label these pairs as "processed" and exclude them from $\mathcal{A}_{uw}$, then we can reject all remaining pairs in $\mathcal{A}_{uw}$.

2.2.6.3.2 Accepting paths

The procedure outlined in the previous section allows us to reject many inadmissible paths with a single shortest distance query. However, the procedure may yield limited performance gain if many of the considered paths are admissible. Therefore, we introduce a second relaxation of our local optimality condition: we classify paths as (approximately) admissible if they are $(\alpha\gamma)$-relative locally optimal with some constant $\gamma \in (0, 1]$.

To see how this relaxation can be exploited, suppose that we are considering an origin-destination pair (s,t) and that we have already confirmed that the path P_{svt} is α-relative locally optimal. Let $x := \underset{\tilde{x}\in P_{sv};\, d(\tilde{x},v)\geq \alpha l(P_{svt})}{\operatorname{argmin}} d(\tilde{x},v)$ be the last vertex on P_{sv} with a distance to v of at least $\alpha \cdot l(P_{svt})$. Let $y := \underset{\tilde{y}\in P_{vt};\, d(v,\tilde{y})\geq \alpha l(P_{svt})}{\operatorname{argmin}} d(v,\tilde{y})$ be defined accordingly for the destination branch. During the T_δ-test we have ensured that the section P_{xvy} is approximately T-locally optimal with $T = \alpha \cdot l(P_{svt})$.

In the lemma below, we will identify the paths that can be classified as approximately admissible after a successful T_δ-test. In line with the notation in the previous section, let $\Omega_x := \{\hat{s} \in O \,|\, d(\hat{s},v) = l\,(P_{\hat{s}xv})\}$, $\Delta_y := \big\{\hat{t} \in D \,|\, d\big(v,\hat{t}\big) = l\big(P_{vy\hat{t}}\big)\big\}$, and $\mathcal{P}_{xy} := \mathcal{P} \cap (\Omega_x \times \Delta_y)$.

Lemma 2.3. *Let $(s,t) \in \mathcal{P}$ be an origin-destination pair. If the T_δ-test applied to P_{svt} considered the vertices on $P_{xvy} \subseteq P_{svt}$ and confirmed that the path is α-relative locally optimal, then all paths $P_{\tilde{s}v\tilde{t}}$ with $\big(\tilde{s},\tilde{t}\big) \in \mathcal{P}_{xy}$ and $l(P_{\tilde{s}v\tilde{t}}) \leq \frac{1}{\gamma} l(P_{svt})$ are at least $(\alpha\gamma)$-relative locally optimal.*

Proof. The T_δ-test for P_{svt} assured that P_{svt} is T-locally optimal with $T = \alpha \cdot l(P_{svt})$. Therefore, all paths $P_{\tilde{s}v\tilde{t}}$ with $\big(\tilde{s},\tilde{t}\big) \in \mathcal{P}_{xy}$ are also T-locally optimal with $T = \alpha \cdot l(P_{svt})$. The local optimality factor $\alpha_{\tilde{s}v\tilde{t}}$ of paths $P_{\tilde{s}v\tilde{t}}$ with $\big(\tilde{s},\tilde{t}\big) \in \mathcal{P}_{xy}$ and $l(P_{\tilde{s}v\tilde{t}}) \leq \frac{1}{\gamma} l(P_{svt})$ is therefore at least

$$\alpha_{\tilde{s}v\tilde{t}} = \frac{T}{l(P_{\tilde{s}v\tilde{t}})} \geq \frac{T}{\frac{1}{\gamma} l(P_{svt})} = \frac{\gamma\alpha l(P_{svt})}{l(P_{svt})} = \alpha\gamma. \tag{2.9}$$

That is, the paths $P_{\tilde{s}v\tilde{t}}$ are at least $(\alpha\gamma)$-relative locally optimal. □

Following Lemma 2.3, we can accept all pairs $\big(\tilde{s},\tilde{t}\big) \in \mathcal{P}_{uw}$ with $l(P_{\tilde{s}v\tilde{t}}) \leq \frac{1}{\gamma} l(P_{svt})$. We do this in the same manner as we rejected paths. Let $\mathcal{A}_{xy} \subseteq \mathcal{P}_{xy}$ be defined as in the previous section. Since $\mathcal{P}_{xy} \backslash \mathcal{A}_{xy}$ contains only pairs $\big(\tilde{s},\tilde{t}\big)$ with $l(P_{\tilde{s}v\tilde{t}}) < l(P_{svt})$, which have been processed before P_{svt}, we only need to consider the pairs in $\mathcal{A}_{xy}$ and classify all not yet processed v-paths $P_{\tilde{s}v\tilde{t}}$ with $\big(\tilde{s},\tilde{t}\big) \in \mathcal{A}_{xy}$ and $l(P_{\tilde{s}v\tilde{t}}) \leq \frac{1}{\gamma} l(P_{svt})$ as admissible. The described procedure to reject and accept multiple paths at once is outlined in Algorithm 2.7.

Algorithm 2.7: Testing whether the potentially admissible paths are approximately α-relative locally optimal.

```
1  R := ∅;                                   // set of approximately admissible paths
2  foreach vertex v ∈ V_via do
3      Let P be the set of all origin-destination combinations for which v is a potential
        via vertex;
4      Sort the pairs in P in increasing order of the lengths of their v-paths;
5      while ≠ ∅ do
6          (s, t) := next origin-destination pair in P;
7          Do a T_δ-test for the path P_svt via v;
8          if the test fails and finds a suboptimal section P_uvw ⊆ P_svt then
9              foreach pair (s', t') ∈ do
10                 if P_uvw ⊆ P_s'vt' then
11                     Remove (s', t') from P;
12         else
13             Add P_svt to R;
14             Let P_xvy ⊆ P_svt be the subsection of P_svt that has been checked for local
                optimality;
15             foreach pair (s', t') ∈ do
16                 if P_xvy ⊆ P_s'et' and γ · l(P_s'vt') ≤ l(P_svt) then
17                     Add P_s'vt' to R;
18                     Remove (s', t') from P;
19 return R;
```

2.2.6.4 Optimization: using previous shortest path queries to determine locally optimal subsections

The outlined speedups become even more effective if the results of individual shortest path queries are reused. Therefore, we save all vertex pairs (u, w) for which we know that $P_{uvw} = P_{uw}$. Note that we do not have to save unsuccessful shortest path tests, because all v-paths $P_{\tilde{s}v\tilde{t}}$ with $P_{uvw} \subseteq P_{\tilde{s}v\tilde{t}}$ will be rejected right after P_{uvw} has been found to be suboptimal (see section 2.2.6.3).

The gain obtained from reusing shortest path results decreases as the considered paths become longer. Since we are considering paths in increasing order of lengths, the lengths of the subsections that are required to be optimal increase as well. Therefore, the results of earlier shortest path queries are of limited value if they are only used as a lookup Table.

However, we can exploit that due to the δ-approximation, the shortest path queries in the T_δ-test typically consider sections longer than required. The T_δ-test conducts shortest path queries between vertices u and their partners $w := \text{partner}_t(u; \delta T)$. Choosing $\delta > 1$ reduces the number of necessary shortest path queries but also makes the algorithm reject admissible paths. Therefore, a test that sets $w := \text{partner}_t(u; \tau)$ for some $\tau \in [T, \delta T]$ will do at least as good as the original algorithm.

With this observation, we can reuse previous shortest path results as follows: when we search for the partner $w := \text{partner}_t(u; \delta T)$ of a vertex u, we test for all intermediate visited vertices $\tilde{w} := \text{partner}_t(u; \tau)$ with $\tau \leq \delta T$ whether the subpath $P_{uv\tilde{w}}$ is known to be optimal. If such a vertex $\tilde{w}$ is found and $\tau \geq T$, we accept $\tilde{w}$ as the partner of u and progress as usual.

2.2.7 Preprocessing

Before REVC can be applied, a preprocessing step is required. If the set of origins and destinations of interest is known a priori, we may start by reducing the graph by deleting dead ends that do not lead to any of the considered origins and destinations. In a second

step, we may add a random perturbation to the edge lengths to make it easier to identify identical paths based on their length. As the road costs (length, travel time, or other) are usually known with limited precision, small perturbations will typically not change the results significantly.

After these preparation steps, we can follow the preprocessing algorithm by Goldberg et al. (2006). The algorithm determines upper bounds on the reaches of vertices. Thereby, the algorithm introduces shortcut edges, which may bias the results so that admissible paths are falsely rejected. However, it is easy to impose a length constraint on the shortcut edges to reduce the introduced error. If REVC is applied to a set of origins and destinations known in the preprocessing phase, vertices bypassed by shortcut edges can be removed completely from the graph. This increases the efficiency further.

The preprocessing step concludes with computing the shortest distances between all origins and destinations. This can either be done with individual shortest path queries for all origin-destination combinations or in a single effort involving only one shortest path tree per origin-destination pair. Either way, this step usually does not add significantly to the algorithm's overall runtime. If the origins and destinations are not known at the reprocessing time, this step can be postponed to the execution of REVC.

2.3 Tests

To test the performance of REVC and to assess how the parameters affect results and computational efficiency, we applied REVC to random route finding scenarios. Below we first provide details on the test procedure and implementation and present the results afterwards.

2.3.1 Test procedure

We tested REVC by applying it to a road network modelling the Canadian province British Columbia (BC). The graph had 1.36 million vertices and 3.16 million edges weighted by travel time. When we preprocessed the graph, we limited the length of shortcut edges to 20 min,

which was less than 3% of the mean shortest travel time between the considered origins and destinations.

We used a Monte Carlo approach to assess the effect of different parameters on the performance and the results of REVC. Specifically, we considered the local optimality constant α, the length constant β, the approximation parameters γ and δ, and the numbers of origins and destinations. We randomly generated 10 route finding scenarios (20 for tests on γ and δ) and computed the mean and standard deviation of the results.

For each random scenario, we selected the origin and destination locations randomly from the graph's vertices. We generated 10 (+10 for tests on γ and δ) sets of origins and destinations, which we reused for each assessed parameter combination to reduce random influences on the results. When we varied the number of origins and destinations, we increased the origin and destination sets as necessary.

To measure the performance of the algorithm, we noted the execution time of the algorithm and the execution time per resulting path. Furthermore, we determined the slowdown factor (see Abraham et al., 2013), denoting the ratio between the execution time of REVC and the corresponding pair-wise shortest path search. In contrast to the execution time, the slowdown factor is not strongly affected by the implementation and hardware, since both REVC and the shortest path queries are run with the same software on the same machine. Therefore, the slowdown factor may be a more meaningful performance measure than the execution time.

Note that it is possible to execute shortest path queries between many origin-destination pairs in linear time of the origins and destinations (Bast et al., 2016). However, the pair-wise approach used to compute the slowdown factor provides a better comparison to pair-based algorithms used in route choice modelling. Therefore, we applied the pair-wise approach.

To assess the resulting paths, we determined the average number and distribution of identified approximately admissible paths and the mean length of these paths. These metrics may provide hints on which parameter combinations are suitable in modelling applications.

2.3.2 Implementation

We implemented REVC in the high-level programming language Python (version 3.7) in combination with the numerical computing library Numpy (version 1.16) and the software Cython (version 0.29), which we used in particular to build a C extension for the shortest path search. Despite our efforts to reduce bottle necks with C extensions, a low-level implementation of REVC can be expected to be faster by orders of magnitude. We computed shortest paths with the algorithm RE (Goldberg et al., 2006). We executed our code in parallel on a Linux server with an Intel Xeon E5-2689 CPU (20 cores with 3.1 GHz) and with 512 GB RAM.

2.3.3 Results

Below we describe our test results. The results are also displayed in Figure 2.7.

The constant α, controlling the local optimality requirement, had a strong influence both on the algorithm's running time and the number of resulting paths. The effect of α on the execution time levelled off at high values of α. Decreasing α from 0.3 to 0.05 increased the execution time by more than 60% and reduced the execution time per identified path by about factor 20. In contrast, increasing α from 0.3 to 0.5 had little effect. The mean number of paths followed a power law in α (exponent -1.75). The length of the resulting paths decreased gradually as α increased. An increase from 0.05 to 0.5 decreased the mean length of admissible paths by about a quarter.

The parameter β, limiting the length of admissible paths, affected the number and length of identified admissible paths but not the execution time. The number of admissible paths increased almost linearly with β, whereby an increase of 0.1 resulted in about 0.7 additional paths being found per origin-destination pair. Consequently, the execution time per resulting path decreased with β. The mean lengths of the identified paths increased with their number. Raising β from 0.1 to 2 increased the mean path length by about 30%.

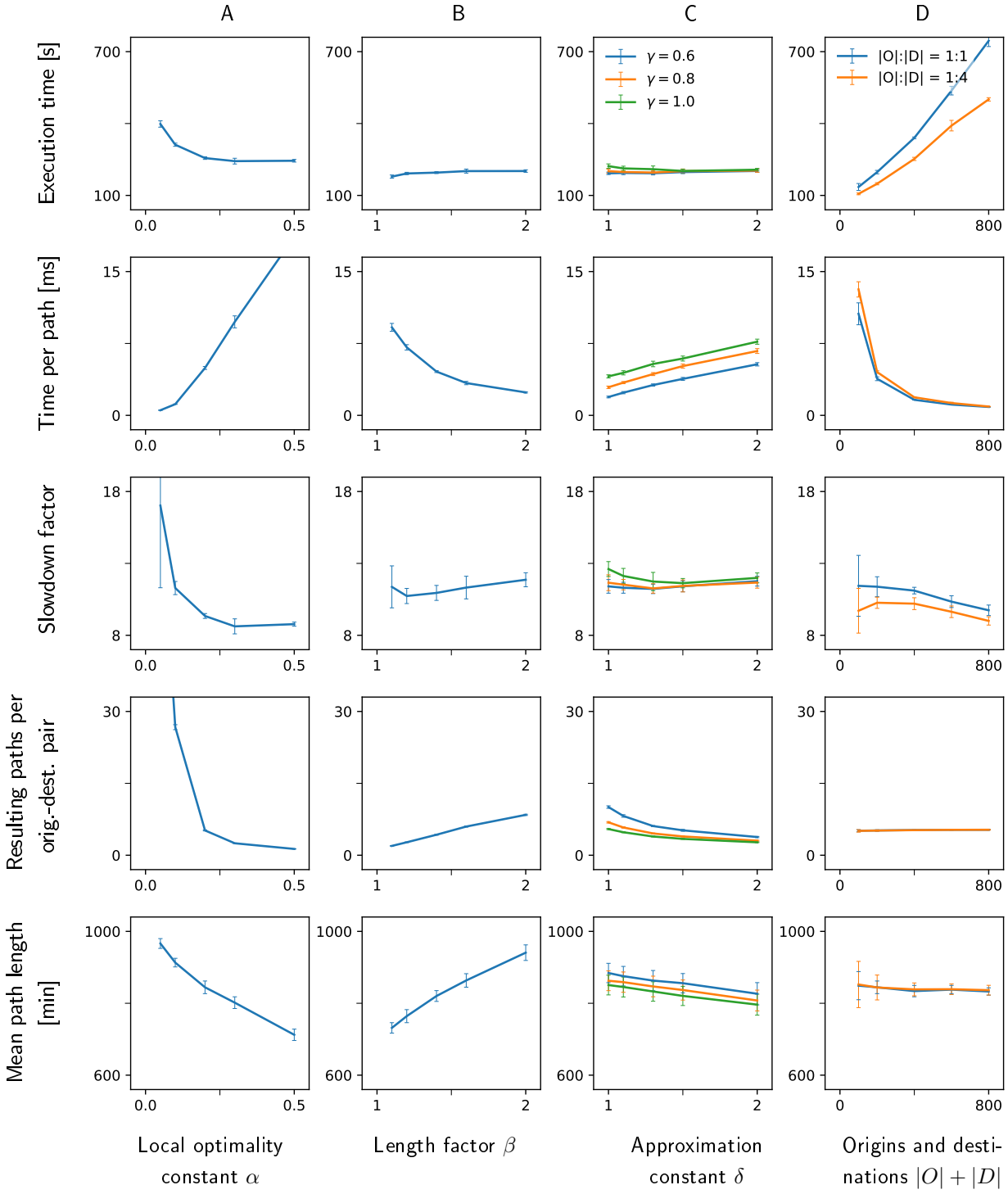


Figure 2.7: Test results. Different performance measures and result characteristics are plotted against parameters. The whiskers depict the estimated standard deviation. The line colours in column C correspond to different values of the approximation constant γ. The line colours in column D correspond to different ratios of origin number and destination number.
(Parameters unless specified otherwise: $\alpha = 0.2$, $\beta = 1.5$, $\gamma = 0.9$, $\delta = 1.1$, $|O| = |D| = 100$)

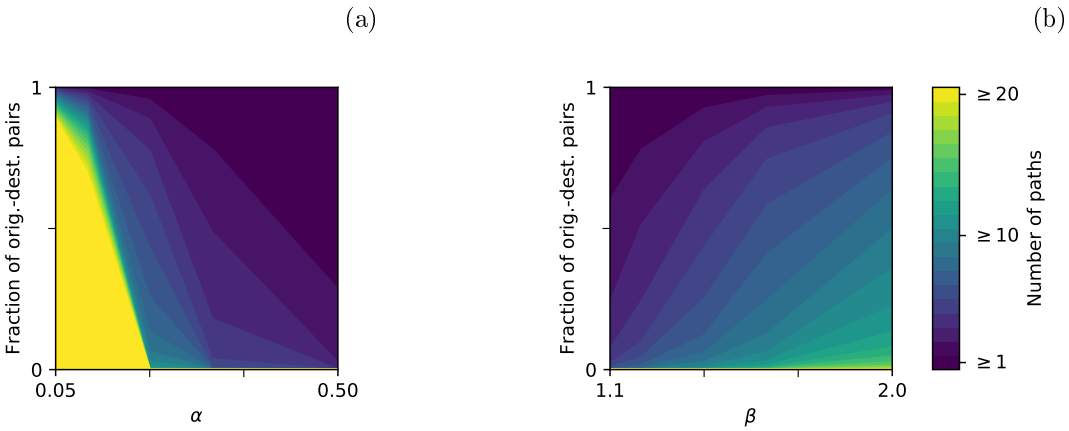


Figure 2.8: Distribution of paths dependent on (a) the local optimality constant α and (b) the length constant β. The y-axis shows which fraction of origin-destination pairs were connected by at least the number of paths given by the colour. The parameters are the same as in Figure 2.7 column A and B.

The approximation parameters γ and δ had little effect on the execution time but a notable impact on the results. An increase of γ (increase in precision) consistently lengthened execution times slightly. However, a decrease of δ (again, increase in precision) *reduced* the execution time per resulting path and led to an optimal execution time at intermediate values of δ.

The number of identified paths varied more strongly than the execution time. Dependent on the value of δ, decreasing γ from 1 to 0.6 increased the number of identified routes by 40%-80%. Conversely, an increase of δ from 1 to 2 decreased the number of identified paths by more than 50%. The lengths of the resulting paths decreased gradually both in γ and δ.

Changing the number of origins and destinations affected the execution time but not the characteristics of the admissible paths. The execution time increased mostly linearly with the origin and destination number, whereby the slope depended on the origin to destination ratio. With a ratio of $1:1$, the execution time increased by 87 s per 100 origins and destinations. With a ratio of $1:4$, the average increase was 56 s per 100 origins and destinations. The time per identified path and the slowdown factor decreased as more origin and destination locations were added.

Figure 2.8 displays the distribution of paths per origin-destination pair dependent on the local optimality constant α and the length constant β. Many origin-destination pairs are connected by numerous admissible paths if α is smaller than 0.2. For example, with $\alpha = 0.1$ and $\beta = 1.5$, about three quarters of the origin-destination pairs were connected by more than 20 routes. In contrast, with $\alpha = 0.3$, less than 0.7% of the pairs were connected by more than 5 paths, and 22% of the pairs were connected by the shortest path only. The latter fraction increased to 72% for $\alpha = 0.5$.

The distribution of paths per origin-destination pair changed more gradually with β. With $\alpha = 0.2$, a large value of $\beta = 2$ resulted in 99% of the pairs being connected by multiple admissible paths, whereby 22% were connected by more than 10 paths. On the other end of the spectrum, with $\beta = 0.1$, 40% of the origin-destination pairs were connected by one admissible path only and 0.6% were connected by more than 5 admissible paths.

2.4 Discussion

We have introduced an algorithm that efficiently identifies locally optimal paths between many origin-destination pairs and tested the algorithm's performance in a realistic road network. Our algorithm REVC identifies all approximately admissible routes between the origins and destinations, and its execution time is driven by the number of distinct origins and destinations rather than the number of origin-destination *pairs*. Therefore, REVC is applicable in large-scale traffic models.

Our test results show that REVC's performance depends mostly on the local optimality constant α and the number of origins and destinations. While the total execution time increases with the number of considered origins and destinations and with decreasing α, the time per identified path gets reduced. That is, REVC becomes more efficient compared to repeated path queries the more paths are generated.

The length bound β had only a minor effect on the execution time. This may be surprising, as an increase in β allows more vertices to be included in the shortest path trees. However,

the impact of β is reduced by our pruning technique, which is most effective for long paths. Furthermore, large parts of the graph had been scanned for small values of β already, since we considered origins and destinations spread over the whole graph. Therefore, few additional vertices were considered with increased β.

The effect of β may be larger if all origin and destination locations are located within a small subsection of the graph. Nonetheless, in many modelling applications, the origin and destination locations will be distributed over the whole considered road network. For example, when the traffic from the outskirts of a city to downtown is modelled, it is unlikely that travellers leave the greater metropolitan area. Therefore, it is reasonable to consider an accordingly constrained graph.

REVC applies approximations to gain efficiency. However, the approximation constants had relatively small effects on the performance in our tests. This suggests that approximations may not always be necessary. However, the benefit of the approximations will become larger if the origin and/or destination vertices are not randomly spread over the whole graph but located in constrained areas. Then, partial results can then be reused more effectively. As the admissibility checks were responsible for a small portion of the overall execution time only, the gain of the approximations will also become more significant if more paths have to be checked for local optimality.

An interesting observation is that intermediate values of the approximation constant δ led to lower execution times than large values. This is surprising, because smaller values of δ increase the number of shortest path queries required in the T_δ-test. However, small values of δ have the advantage that the subsections checked for local optimality get shorter. This makes it more likely that test results can be reused to reject many inadmissible paths at once. In point to point queries, the T_2-test (used by Abraham et al., 2013) may still be superior.

2.4.1 Significance

Determining multiple paths between an origin and a destination based on a local optimality criterion is a well established approach in route planning research (Abraham et al., 2013; Delling et al., 2015; Luxen and Schieferdecker, 2015; Bast et al., 2016). An obstacle hindering the application of these algorithms in route choice models was that these algorithms return only few heuristically chosen paths rather than the complete set of admissible paths. Furthermore, these algorithms are based on an inflexible approximation whose impact on the result was not exactly known. Our algorithm REVC solves these issues. Though REVC may not be competitive in point to point queries, the algorithm efficiently exploits redundancies occurring when many origin-destination pairs are considered.

Generating route choice sets based on local optimality has multiple advantages. The underlying principle is simple and has a sound mechanistic justification. The optimality principle is applied on a local scale, whereas the mechanisms governing travellers' overall route choices do not need to be known. Therefore, no extensive data sets are needed to generate choice sets.

Fitting the choice set parameters to data is a discrete optimization problem and can therefore be challenging. REVC permits two free variables: the local optimality parameter α and the length parameter β. As the latter does not have a strong impact on the execution time, β can be chosen liberally, leaving α as the only remaining free parameter. Optimizing α, in turn, is comparatively easy, as this is a one-dimensional problem.

Choice sets formed by locally optimal v-paths are typically relatively small while still covering a broad spectrum of different routes (see Abraham et al., 2013). This allows for sophisticated models for the second decision step, in which travellers choose routes from the choice sets. The option to use sophisticated metrics to measure the quality of the route candidates may improve the overall model fit.

The favourable quality to quantity ratio of locally optimal v-paths and the practically linear relationship between execution time and origin and destination numbers make REVC par-

ticularly useful in comprehensive traffic models. In such applications, many origin-destination pairs have to be considered, and the computed choice sets need to be kept in memory for further processing. This makes it difficult to apply methods based on point to point queries, such as link elimination (Azevedo et al., 1993), link penalty (De La Barra et al., 1993), or constrained enumeration methods (Prato and Bekhor, 2006). Similar challenges face algorithms that need to generate many paths, such as stochastic approaches or methods that include a filtering step to select admissible paths from a large number of candidates (see Bovy, 2009). Therefore, REVC may be of specific use in comprehensive models.

The results of REVC provide insights into the distribution and properties of locally optimal routes in real road networks. In our tests, the number of admissible paths decreased with α in a power law relationship, whereas it increased linearly in β. Such experimental results could be the starting point for a more in-depth theoretical analysis of the distribution of locally optimal routes in road networks. The resulting insights may facilitate the development of new algorithms.

The experimental results are also valuable as benchmarks for existing algorithms searching locally optimal v-paths for route planning purposes (Abraham et al., 2013; Kobitzsch, 2013; Luxen and Schieferdecker, 2015). Some of these algorithms apply approximations to gain efficiency. The presented results can help to assess the impact of these approximations. Our results suggest that the applied T_2-approximation falsely rejects half of the admissible paths.

In addition to assessing the accuracy of faster algorithms, the complete sets of admissible paths generated with REVC can also be used to evaluate the success rate and the quality of the paths generated with these algorithms. Note, however, that our definition of admissible paths deviates slightly from the definition applied in earlier papers. Refer to Appendix 2.C for details.

REVC contains several optimizations that can be directly applied to make the family of algorithms based on REV more efficient. These optimizations include the improved bounds for tree growth and pruning as well as the idea to exclude u-turn paths by considering via

edges. Similarly, the T_δ-test can be directly applied to increase the accuracy of all algorithms using the T-test. Hence, this paper may also contribute to make route planning software more efficient. We provide a more in-depth discussion in Appendix 2.C.

2.4.2 Limitations

REVC focuses on single-via paths. A complete search for locally optimal routes should not limit the set of considered paths. However, considering v-paths can be justified by assuming that travellers may drive via an intermediate destination. Furthermore, the focus on v-paths excludes zig-zag routes, which may be deemed unrealistic. Therefore, a criterion limiting the set of admissible paths may not only be a computational necessity but also beneficial in route choice models.

Nonetheless, REVC may be extendable to include paths via two intermediate destinations. Road networks usually have a small set W of vertices so that every sufficiently long shortest path includes at least one of these vertices (Abraham et al., 2010). If W could be identified efficiently, REVC could be applied to compute v-paths from the origins to the vertices in W and from the vertices in W to the destinations. Concatenating these v-paths to admissible "double-via" paths would be comparable to the admissibility checks described in this paper.

REVC seeks to identify all admissible paths between the given origins and destinations. However, even if we do not apply approximations (i.e. choose $\gamma = \delta = 1$), some admissible paths may be falsely rejected. This limitation is due to the preprocessing step, in which shortcut edges are added to the graph, and the requirement that an edge adjacent to the via vertex must be scanned in forward and backward direction. However, we have already noted that the effect of the shortcut edges can be arbitrarily reduced by imposing length constraints on shortcut edges. Furthermore, most admissible paths will satisfy the mentioned edge requirement (see Appendix 2.B). Therefore, these limitations generally have minor effects on the results.

REVC, as introduced in this paper, identifies identical paths based on their lengths. Alternative approaches exist but might be less efficient. In practice, distinct paths may have identical lengths, and REVC may therefore falsely reject some admissible paths. Paths with equal lengths occur most frequently in cities whose roads form a grid structure. Nevertheless, since the roads may have distinct speed limits and traffic volumes, and because turns take additional time, paths with identical lengths may not occur frequently in practice. Since ties are even less likely in long paths, we argue that it is reasonable to distinguish paths based on their lengths.

Misclassifications of distinct paths with equal lengths can be reduced by adding small random perturbations to the lengths of all edges. Though this procedure makes it unlikely that admissible paths with similar lengths are considered identical, the perturbation term randomly defines an optimal path in grid networks. Therefore, the random perturbation is of limited help in these networks. Note, however, that regardless of how we identify identical paths, REVC and similar shortest path based methods are not well suited in grid networks, as ties must be broken when the shortest path trees are grown.

In this paper, we presented performance measurements to assess the efficiency of REVC. When evaluating these results, it is important to note the limitations of our implementation. For example, our parallel implementation comes with scheduling overheads. Some parts of the algorithm were not parallelized at all, leaving room for further speedups. Furthermore, the slowdown factors we measured can be considered as upper bounds, since we compared a highly optimized shortest path search with a high-level implementation of REVC. Despite these limitations, the most important timing result remains visible: the performance of REVC scales well with the numbers of routes and end points.

We provided several conceptual arguments suggesting that sets of locally optimal v-paths are likely to cover most paths considered by real travellers. Nonetheless, we did not present empirical evidence in this paper. In chapter 3 REVC will be applied to model the traffic of recreational boaters across North America. However, an in-depth empirical validation of the

hypothesis that travellers generally choose locally optimal paths remains a task for future research.

2.5 Conclusion

Generating route choice sets with locally optimal single-via paths has a sound mechanistic justification, leads to small choice sets with reasonable alternatives, and requires minimal data. We presented an algorithm that efficiently generates such choice sets for large numbers of origin-destination pairs. The algorithm is able to identify (almost) all locally optimal single-via paths up to a specified length between the origins and destinations. Therefore, the algorithm extends earlier methods based on local optimality and makes the approach a valuable method to generate route choice sets.

We tested our results on a real road network and assessed the algorithm's performance dependent on the input parameters. The results provide insights into the effect of approximation parameters and the distribution of locally optimal paths in real road networks. Therefore, our study provides the necessary prerequisites to construct route choice sets based on local optimality in large-scale traffic simulation applications.

Appendices

2.A Proofs

In this Appendix, we prove Lemma 2.1 and Corollary 2.1 (main text). We adjust the statement of Lemma 2.1 to recall notation from the main text.

Lemma 2.1. *Consider an arbitrary admissible single-via path P from s to t. With $x'_s = \operatorname{argmin}_{x \in P;\, d_P(s,x) \geq \alpha l(P)} d_P(s,x)$, let*

$$x_s := \begin{cases} x'_s & \text{if } d_P(s,x'_s) \leq \frac{1}{2} l(P) \\ \operatorname{argmax}_{x \in P;\, d_P(s,x) \leq \frac{1}{2} l(P)} d_P(s,x) & \text{else.} \end{cases} \tag{A2.1}$$

Choose x_t accordingly. Then there is at least one vertex $v \in P$ with

1. $d_P(s,v) = d(s,v) \leq d_P(s,x_t)$ *and*
2. $d_P(v,t) = d(v,t) \leq d_P(x_s,t)$.

Proof. Since P is a single-via path, P contains at least one vertex v' such that $d_P(s,v') = d(s,v')$ and $d_P(v',t) = d(v',t)$. That is, v' splits P into two shortest paths. Now choose a vertex v as follows:

$$v := \begin{cases} v' & \text{if } d_P(s,v') \leq d_P(s,x_t) \text{ and } d_P(v',t) \leq d_P(x_s,t), \\ x_t & \text{if } d_P(s,v') > d_P(s,x_t), \\ x_s & \text{if } d_P(v',t) > d_P(x_s,t). \end{cases} \tag{A2.2}$$

We show that v satisfies the lemma's requirements by regarding the different possible choices of v:

1. If $d_P(s, v') \leq d_P(s, x_t)$ and $d_P(v', t) \leq d_P(x_s, t)$, then the conditions 1 and 2 are clearly satisfied for $v := v'$.

2. If $d_P(s, v') > d_P(s, x_t)$, then inserting $v := x_t$ yields $d_P(s, v') > d_P(s, v)$. Therefore, the subpath P^{sv} from s to v is a subpath of the subpath $P^{sv'}$ from s to v'. Since v' splits P into two shortest paths, $P^{sv'}$ is a shortest path. Therefore, P^{sv} must be a shortest path, too. Thus, $d_P(s, v) = d(s, v) = d_P(s, x_t)$, and condition 1 is satisfied.
   To show that condition 2 holds as well, observe that $d_P(v, t) = d_P(x_t, t) \leq \frac{1}{2} l(P) \leq l(P) - d_P(s, x_s) = d_P(x_s, t)$. It remains to be shown that $d_P(v, t) = d(v, t)$. Since P is α-relative locally optimal, each subpath whose length after removal of one end point would be smaller than $\alpha l(P)$ is a shortest path. By construction, this applies to the subpath from x_t to t. Hence, it is $d_P(v, t) = d(v, t)$ and condition 2 is satisfied.

3. The proof for the case $d_P(v', t) > d_P(x_s, t)$ is analogous to the argument presented under point 2.

□

Corollary 2.1. *For each admissible v-path between an origin-destination pair (s, t), a via vertex will be scanned from both directions if the shortest path trees are grown up to a height of*

$$h_{max} := \max \left\{ (1 - \alpha)\, \beta l(P_{st}),\ \frac{1}{2} \beta l(P_{st}) \right\}. \tag{A2.3}$$

Proof. Let P be an admissible path, which implies that $l(P) \leq \beta l(P_{st})$. Recall that

$$\begin{aligned} x'_t &= \operatorname*{argmin}_{x \in P;\, d_P(x,t) \geq \alpha l(P)} d_P(x, t) \\ &= \operatorname*{argmin}_{x \in P;\, l(P) - d_P(s,x) \geq \alpha l(P)} (l(P) - d_P(s, x)) \\ &= \operatorname*{argmax}_{x \in P;\, d_P(s,x) \leq (1-\alpha) l(P)} d_P(s, x). \end{aligned} \tag{A2.4}$$

Therefore, x_t is either the last vertex in P with $d_P(s,x) \leq (1-\alpha)\, l(P) \leq (1-\alpha)\, \beta l(P_{st})$ or the last vertex with $d_P(s,x) \leq \frac{1}{2} l(P) \leq \frac{1}{2}\beta l(P_{st})$ (see equation (A2.1)). Either way, x_t will be included in the shortest path tree if we grow the tree to a height of just above $\max\left\{(1-\alpha)\, \beta l(P_{st}),\, \frac{1}{2}\beta l(P_{st})\right\}$. The same argument holds in backward direction for x_s. From Lemma 2.1 we know that P is a v-path via a vertex $v \in P^{x_s x_t}$ located between x_s and x_t. Since both x_s and x_t are scanned from both sides, the vertex v will be scanned from both sides as well. □

2.B Admissible paths excluded by requiring that a neighbouring edge of the via vertex has been scanned from both directions

Requiring that a neighbouring edge of the via vertex has been scanned in both directions excludes u-turns without reducing the number of found admissible paths significantly. However, there is exactly one scenario in which an admissible v-path is not found if we impose this constraint. The situation is depicted in figure A2.1.

Suppose the v-path P from s to t via the vertex v is admissible but falsely rejected by the exact version of REVC ($\gamma = \delta = 1$). Suppose furthermore that $u \in P$ is the predecessor of v and $w \in P$ the successor. Then there must be a vertex $x \in P^{su}$ and a vertex $y \in P^{wt}$ such that the following conditions hold:

1. The shortest path from x to w does not include v: $d(x,v) + d(v,w) > d(x,w)$.
2. The shortest path from u to y does not include v: $d(u,v) + d(v,y) > d(u,y)$.
3. Let x' be the direct successor of x in P. It must be $d(x',v) > \alpha \cdot l(P)$.
4. Let y' be the direct predecessor of y in P. It must be $d(v,y') > \alpha \cdot l(P)$.
5. The shortest path from u to w must include v: $d(u,w) = d(u,v) + d(v,w)$.

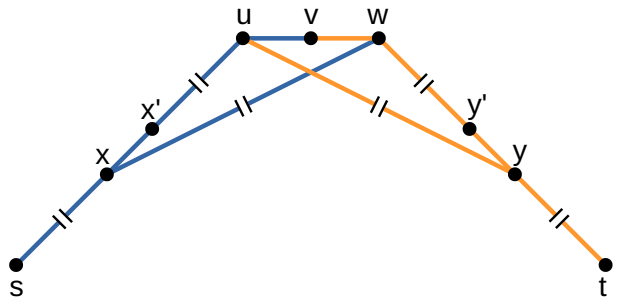


Figure A2.1: Scenario in which an admissible path is excluded due to the requirement that an edge adjacent to the via vertex is scanned in both directions. Blue lines depict the edges included in the forward shortest path tree grown from the origin s and orange lines the edges of the backward tree grown into the destination t. Lines that may represent multiple edges are indicated with a gap. As the edges adjacent to v are included in one shortest path tree only, the path P_{svt} would be rejected by REVC.

If the first two conditions were not satisfied, at least one edge on P adjacent to v would be scanned from both directions and P would be found. If the last three conditions were not satisfied, P would not be admissible.

Though it is possible that all of these conditions are satisfied, we believe that such a scenario is unlikely in real road networks.

Remark 2.1. It can be shown that pruning does not weaken these conditions.

2.C Comparison of REV and REVC

In this Appendix, we compare our algorithm REVC to the algorithm REV (Abraham et al., 2013) that it is based on. To a large extent, REVC uses the same ideas as REV: shortest path trees are grown around the origin and destination, and v-paths via vertices scanned from both directions are checked for admissibility using an approximate test for local optimality. However, REV and REVC differ in (1) the admissibility definition (2) the choice of the returned paths, and (3) technical optimizations that REVC introduces. Below we discuss each of these points.

2.C.1 Admissibility definition

The admissibility definition by Abraham et al. (2013) includes three requirements. They say a v-path P_{svt} is admissible if

1. P_{svt} has limited overlap with previously identified admissible paths P_{swt} between s and t. That is, $l\Big(P_{svt} \cap \Big(\bigcup_w P_{swt}\Big)\Big) \leq \eta \cdot l(P_{st})$.

2. P_{svt} is T-locally optimal with $T = \alpha \cdot l(P_{st})$.

3. P_{svt} has β-uniformly bounded stretch. That is, for all $u, w \in P_{svt}$, it is $l(P_{svt}^{uw}) \leq \beta \cdot l(P_{uw})$.

None of these requirements coincides exactly with the constraints we imposed in our paper.

Requirement 1 does not appear in our admissibility definition. The constraint requires that the admissible paths have a clearly specified order. However, though Abraham et al. (2013) suggest a reasonable ordering, this introduces another degree of freedom whose impact on the results may be oblique. Furthermore, we were interested in identifying *all* routes that satisfy certain criteria and leave it to the second modelling stage, in which a route is chosen from the choice set, to take route overlaps into account (see e.g. Cascetta et al., 1996). Lastly, the local optimality criterion naturally limits the pair-wise overlap of paths. Therefore, we dropped this constraint.

Requirement 2 differs from our local optimality constraint, because the length T of the subsections required to be optimal depends on the shortest distance between s and t rather than the length of the via path. This allows for more admissible paths. We changed this requirement for two reasons: (1) the spatial scale at which travellers' decision routines change is likely dependent on the path they *actually* choose rather than the shortest alternative, which may – dependent on the global quality metric – not even be a favourable option. Travellers on a long trip may have a higher incentive to choose a route with long optimal subsections. (2) The adjusted local optimality criterion allows for more effective pruning with

simpler bounds when considering many origin-destination pairs. Using a pair-wise static local optimality criterion as Abraham et al. (2013) would require us to choose the pruning bound dependent on the origin-destination pair closest together. For these reasons, we introduced the notion of relative local optimality. Note that REVC can also be used to identify all paths satisfying requirement 2 if the constant α is adjusted accordingly and the resulting paths are filtered so that suboptimal paths are excluded.

Requirement 3 is relaxed in our admissibility definition. Abraham et al. (2013) do not introduce an efficient algorithm to identify paths satisfying requirement 3. Instead of bounding the lengths of all subpaths, they consider the complete path only, as we do in this paper. Nonetheless, uniformly bounded stretch is a valuable characteristic for choice set elements. However, since REVC will return a moderate number of paths in many applications, paths could be checked for uniformly bounded stretch after execution of REVC. Consequently, we have used the relaxed constraint directly.

2.C.2 Returned paths

Abraham et al. (2013) aim to compute a small number of high-quality paths between an origin and a destination efficiently. To save computation time, they do not assess the admissibility of all path candidates. Instead, REV processes the potentially admissible paths in an order dependent on some objective function, estimating the quality of the paths. REV returns the first n processed approximately admissible paths.

Since we are interested in an exhaustive search for admissible paths, we do not process the paths in a specific order. We return all approximately admissible paths and leave the assessment of their quality, if desired, to a second, independent algorithm.

2.C.3 Optimizations

REVC introduces multiple optimization to REV. First, REVC uses a tighter bound for the tree growth and the pruning stage. Though our pruning bound would have to be adjusted

to comply with the admissibility definition applied by Abraham et al. (2013) (see section 2.C.1), the ideas introduced in this paper are still applicable.

Second, REVC excludes u-turns by considering via edges rather than via vertices. Furthermore, REVC identifies vertices representing identical paths before assessing their admissibility. Both optimizations could be directly applied to speed up REV. However, REV processes the paths in an order given by some objective function (see section 2.C.2). It is possible to construct this objective function so that u-turn paths are not processed before any admissible path.

Third, to control the accuracy of the results, REVC uses the T_δ-test instead of the T-test to check whether a path is locally optimal. This optimization could also be applied in REV, though it may effect the performance of REV more strongly than the performance of REVC.

Lastly, REVC is optimized to process many origin-destination pairs at once. Though the idea to grow each shortest path three only once per origin and destination is straightforward, the main innovation of REVC is in the efficient local optimality checks of many v-paths via one via vertex.

Chapter 3

A hybrid gravity and route choice model to assess vector traffic in large-scale road networks

3.1 Introduction

Assessing road traffic and the transportation of goods through road networks is key to understanding the impacts of human movement in the context of epidemiology and invasion biology. For example, animal transport and trade are major vectors for animal and human diseases (Karesh et al., 2005). Similarly, many invasive species spread by means of human traffic along roads. Examples include plant seeds contained in dirt on cars (Von der Lippe and Kowarik, 2007), insects carried in firewood of campers (Koch et al., 2012), baitfish carried by anglers (Drake and Mandrak, 2014), and aquatic invasive species "hitchhiking" on trailered watercraft (Johnson et al., 2001).

To understand and control these processes, scientists and managers need estimates of the traffic flows in road networks. There are two perspectives on modelling traffic flows: the supply/demand perspective (Friedrich et al., 2014) and the route choice perspective (Prato, 2009). While models for supply and demand (or travel incentive and destination choice) measure the motivation for travel or transport, route choice models determine the pathways along which the travel or transport occurs. Individually, supply/demand models and route choice models provide powerful tools for estimating traffic flows. However, as we will show below, there are situations where a hybrid approach is desirable.

The distribution of trips between origins and destinations is often modelled with gravity models (Anderson, 2011), which have two main sources of data: on-site surveys of individual agents taken at source/destination locations, or mail-out surveys collecting details of planned or past trips from potential travellers. In general, on-site surveys yield precise estimates of absolute traffic flows but are more expensive, unless the data are readily available e.g. through booking records. In contrast, mail-out surveys may be more subject to sampling error but less expensive. While both survey types are used for parameterizing gravity models, field surveys are typically necessary if absolute measures of traffic flows are needed.

A potential alternative approach is to sample the traffic flow at given locations on roads. This contrasts with the on-site survey approach described above, where agents are sampled at source or destination locations. In many realistic situations, surveys conducted at intermediate roads can provide much more data than origin/destination sampling. For example, consider a region with 100 possible sources and a region with 100 possible destination locations, with 2 main routes connecting them. The number of agents travelling along any of these main roads will, on average, be 50 times higher than the number leaving from or arriving at any individual location. Therefore, when there are many possible source and destination locations but few major routes linking them, the number of agents sampled at intermediate roads will far exceed the number sampled leaving sources or arriving at destinations.

Because of the large amounts of data potentially available along roads, it would be advantageous to use such data to parameterize gravity models. However, to the best of our knowledge, this has not yet been done. As the traffic flow through roads depends on travellers' route preferences, a hybrid approach, which links gravity models to route choice models, would be required. This is the approach taken in this paper.

Gravity models and large-scale systems

The main idea of gravity models is to estimate the number of trips between an origin and a destination location based on agents' tendency to start a trip at the origin (repulsiveness),

their tendency to travel to the destination (attractiveness), and the distance between origin and destination. Based on this basic idea, variations on gravity models have been derived to increase their predictive accuracy and mechanistic validity, such as constrained gravity models (Wilson, 1970) and stochastic gravity models (Flowerdew and Aitkin, 1982). In "classical" gravity models, traffic flows are assumed to be deterministic, and variations in observed traffic are viewed as measurement error. In contrast, stochastic gravity models suppose that the traffic flow itself is a stochastic process. That is, properties of donor and recipient determine the *mean* traffic flow, whereas the *actual* traffic flow varies over time, following some stochastic distribution.

Though stochastic gravity models were originally developed in the context of economics (Flowerdew and Aitkin, 1982), they have also been successfully applied in invasion ecology and epidemiology to model the traffic of potential invasive species or disease vectors (Drake and Mandrak, 2010; Potapov et al., 2010; Muirhead and MacIsaac, 2011; Muirhead et al., 2011; Potapov et al., 2011; Barrios et al., 2012; Chivers and Leung, 2012; Drake and Mandrak, 2014). The systems modelled in these studies had small or medium spatial scale. However, long-distance trips can occur sufficiently often to pose a considerable risk of introducing invasive species or diseases to regions far away from the infested area. Hence, long-distance trips can be a major factor for shifting invasion or disease fronts (Kot et al., 1996). Therefore, models for long-distance traffic are needed.

In large-scale systems, it is hard to collect the data required to fit a gravity model. Often, origins and destinations span over large areas, or regions of origin and destination may be considered instead of individual locations. In both cases, the considered origins and destinations have many access points, which are expensive to monitor all at once. Conducting mail-out surveys is usually not an option, too, as only few of the surveyed individuals who could potentially start a trip *will* actually start a long-distance trip and thus provide useful data. Consequently, an alternative approach is required to fit gravity models in large-scale systems.

The shortcomings of gravity models in large-scale systems concern not only the model fit but also how the models can be used to facilitate management of diseases or invasive species. A common management goal is to reduce the number of vectors leaving an infested area or entering a susceptible area. As the number of origins and destinations is large and they may have many access points in large-scale systems, it may be infeasible to apply control directly at the infested and susceptible locations. Instead, managers may want to control the traffic on intermediate roads that are shared by agents travelling from different origins to different destinations. To find the best roads for such control measures, a route choice model is necessary, which determines how the traffic between an origin and a destination is distributed over the road network.

Route choice models

Travellers are usually not able to consider all possible routes to their destination due to the vast number of options. Therefore, many route choice models assume that travellers make route choices in two steps: first, they apply some heuristic to determine a set of potentially good ("admissible") routes, and second, they choose one of these routes based on their characteristics (Di and Liu, 2016).

A variety of approaches have been developed to model the two decision steps. Models for route admissibility may determine all routes that satisfy certain criteria or focus on routes that are optimal with respect to different goodness measures (Bovy, 2009). Alternatively, locally optimal routes may be considered (see chapter 2), which assume that travellers act rationally on local scales while unknown factors may affect the routes on large scales. This method has been found to yield realistic routes while maintaining high computational efficiency (chapter 2; Abraham et al., 2013).

To model the second stage of the decision process, the admissible routes are typically assigned probabilities for being chosen. The corresponding models may include economic aspects, such as the length of a route and the expected travel time, but also other factors,

such as potential intermediate destinations and the scenery and sights along a route (Prato, 2009). However, since multiple admissible routes between all combinations of origins and destinations must be considered, large-scale systems require a model balancing accuracy and computational efficiency.

Outline

Both gravity models and route choice models are widely used in their respective fields. In this paper, we present a hybrid model combining the two to assess traffic in large-scale systems. Since traffic varies over time, we use an additional model to account for time-driven variations in survey data. Furthermore, we introduce another model for the compliance of travellers, because not every traveller may participate in the survey and provide complete information. This hybrid approach allows us to fit a gravity model to data collected in road-side surveys. As a result, the hybrid method is applicable regardless of the system's spatial scale and yields not only estimates of the traffic outflow and inflow of origins and destinations but also estimates the traffic volume on roads.

We demonstrate our approach by applying it to the potential invasion of zebra and quagga mussels *Dreissena spp.* to the Canadian province British Columbia (BC). Dreissenid mussels are invasive in North America and cause severe economic and ecological damages (Pimentel et al., 2005; Rosaen et al., 2012). A major spread mechanism of zebra and quagga mussels is boaters transporting mussel-infested watercraft and gear to uninvaded lakes (Johnson et al., 2001). Therefore, knowledge of destinations and travel routes for these boaters is key for mussel prevention and early detection.

This paper is structured as follows: in section 3.2, we give an overview of the hybrid approach and the submodels for the the distribution of trips between origins and destinations, the route choice, temporal traffic patterns, and the compliance of travellers. In section 3.3, we describe how survey data collected at roads can be used to fit the submodels. In section 3.4, we apply the hybrid model to the potential invasion of dreissenid mussels to BC and

present the resulting estimates of vector pressure and pathways in BC. Finally, in section 3.5, we discuss shortcomings, applicability, and potential extensions of our approach.

3.2 Model

Before introducing our hybrid traffic model, we need to clarify which travellers we want to consider. Not every person travelling from an infested region to a susceptible destination has the potential to carry a disease or invasive species. Similarly, not every potential carrier of propagules or pathogens will actually be infested and thus be a vector. In this paper, we assess the traffic of all *potential* vectors, regardless of whether they carry pathogens or propagules. Below, we call these potential vectors "agents".

We propose a hierarchical approach to model how many agents can be observed in a survey shift conducted at a road side. An agent will be observed in a road-side survey if, and only if, they (1) start a trip, (2) choose a route via the survey location, (3) time their journey so that they pass the survey location during the survey shift, and (4) participate in the survey. Since these decisions are difficult to know precisely, we assume that the number of surveyed agents results from a hierarchical stochastic process (see Figure 3.1): (1) every time unit, a random number N_{ij} of agents travel from origin i to destination j; (2) out of these agents, a random number N_{ijk} choose a route via the survey location k; (3) out of these agents, a random number N_{ijkt} time their journey so that they pass the survey location during the time interval t when the survey is conducted; (4) out of these agents, a random number N^{+}_{ijkt} agents decide to participate in the survey and provide complete information. This approach allows us to fit the model to data collected in road-side surveys.

The distributions of N_{ij}, N_{ijk}, N_{ijkt}, and N^{+}_{ijkt} depend on submodels. Though some applications may require more specific submodels, we now propose a set of models applicable in many real-world systems. A detailed list of our assumptions can be found in Appendix 3.A.

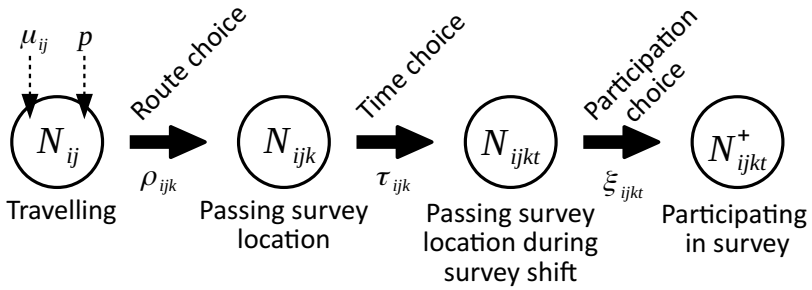


Figure 3.1: Hierarchical stochastic model for the number of agents passing a survey location during a survey shift. The total number N_{ij} of agents travelling from i to j depends on the parameters μ_{ij} and p. With a probability ρ_{ijk}, the travelling agents will choose a route via the survey location k. With probability τ_{ijkt}, the N_{ijk} agents who choose such a route will also time their journey so that they pass the location in the time interval t when the survey is conducted. These N_{ijkt} choose with probability ξ_{ijkt} to participate in the survey and to provide complete information. The resulting N^{+}_{ijkt} agents are the ones included in the survey.

3.2.1 Gravity model

We model the daily numbers N_{ij} of agents travelling from origin i to destination j with a stochastic gravity model. The mean value μ_{ij} of the random variable N_{ij} is proportional to the repulsiveness m_i of the origin i, the attractiveness a_j of the destination j, and a negative power of the distance between i and j:

$$\mu_{ij} \quad = \quad c\frac{m_i a_j}{d_{ij}^{\alpha_\mathrm{d}}}. \tag{3.1}$$

Usually, m_i and a_j are estimated as functions of covariates that correlate with the number of agents leaving donor region i and the number of agents arriving at recipient j, respectively.

The functions used to estimate m_i and a_j consist of "building blocks" corresponding to one covariate x_r, $r \in \{1, \ldots, n\}$, each. Convenient functional forms for the building blocks are the power function $f_0(x_r) := x_r^{\alpha_1}$ and the saturating function $f_1(x_r) := \left(\frac{x_r}{x_r+\alpha_0}\right)^{\alpha_1}$. The functional form f_1 is appropriate if the covariate has a particularly high impact after some threshold value or if differences in large covariate values are insignificant (see e.g. Potapov et al., 2010). Otherwise, f_0 is typically sufficient.

Many such building blocks can be connected to account for spatial heterogeneity. If two covariates are effective only in combination with each other, their respective building blocks should be multiplied together. For example, if *both* recreational opportunities and accommodations are necessary to attract agents, attractiveness is given by the product of the corresponding building blocks. In turn, if covariates have an effect independent of each other, the respective building blocks should be added together. For example, if *either* a boat launch or mountain biking opportunities can attract agents, the corresponding building blocks should be added together. In that sense, multiplication models an "and" relationship, whereas addition models an "or" relationship.

Though the mean number μ_{ij} of travelling agents is given by a deterministic function, the number N_{ij} of agents travelling in a time unit follows a stochastic distribution. Most stochastic gravity models build on the Poisson distribution, the negative binomial distribution, or the zero-inflated negative binomial distribution (Burger et al., 2009). The Poisson distribution is applicable if agents decide independently of each other in each time unit whether they start a trip. If agents' decisions are correlated, for example because weather conditions, holidays, and other factors affect many agents at once, the density of the Poisson random variable can be chosen to vary with time. If the sources of correlations are not known precisely, a negative binomial distribution can be used to approximately account for the overdispersion resulting from such correlations (Gardner et al., 1995). Lastly, zero-inflated distributions suppose that there is a stochastic mechanism that stops all agents from travelling between an origin and a destination in some time units. In the remaining time units, N_{ij} is assumed to follow a common stochastic distribution, such as the negative binomial distribution. We build our gravity model based on the negative binomial distribution, as this distribution is appropriate in many use cases and generalizes the Poisson distribution.

We parameterize the count distribution so that the ratio between mean and variance of the agent counts is constant for all origin-destination pairs. With this parameterization, the sum of two independent negative binomial random variables is still negative binomially

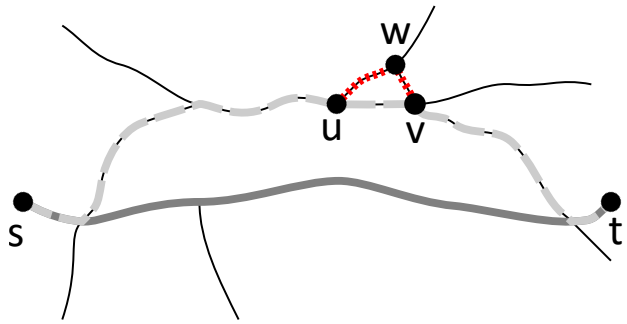


Figure 3.2: Admissible paths from origin s to destination t. The shortest path (solid grey) and the path via u and v (dashed grey) are admissible. The path via point w (dotted red from u to v, dashed grey from s to u and from v to t) is inadmissible, because it is not locally optimal: the short subsection $u \to w \to v$ (dotted red) is not a shortest path.

distributed. This is particularly important when the model is built to assess traffic between regions of multiple individual origin or destination locations. In this scenario, the flow between the regions is the sum of the flows between the individual locations. Choosing a constant mean to variance ratio makes the model invariant to how the individual locations are pooled together. Refer to Appendix 3.B for further details.

3.2.2 Route choice model

We assume that agents choose their routes randomly and independently from one another. This is reasonable, because agents of concern usually constitute only a fraction of the full traffic on a road. Therefore, traffic jams and other traffic-dependent factors that affect the attractiveness of routes are mostly independent of the modelled agents' routing decisions.

Many route choice models assume that agents choose their routes from a small set of "admissible" routes (Prato, 2009). We define route admissibility as in chapter 2 in this thesis, where we claim that admissible paths should not contain local detours. The rationale behind this claim is that major route decisions may be affected by factors unknown to us, while minor route decisions follow strict rational rules. Consequently, an admissible path P can only contain a detour if the detour is longer than $\delta \cdot \text{length}(P)$. The constant δ defines which detours are deemed "local". We illustrate this concept of local optimality in Figure 3.2.

The resulting set of admissible paths may still be very large. To limit the number of admissible paths further, we require that they are not more than a factor γ longer than the shortest alternative. Furthermore, we focus on "single-via paths". These are shortest paths via one arbitrary intermediate destination, respectively. We compute the corresponding set of admissible paths with the algorithm presented in chapter 2.

After computing the set of paths that agents may choose from, we need to assign the individual paths with probabilities. We assume that the probability that an agent chooses a route P is inverse proportional to a power of its length l_P. That is, if $\mathcal{P}_{ij}$ is the set of admissible routes from origin i to destination j and $\lambda \geq 0$ a constant, the probability to choose route P is given by

$$\mathbb{P}(\text{choose route } P \,|\, \text{travelling on admissible route}) \;=\; \frac{l_P^{-\lambda}}{\sum\limits_{\tilde{P} \in \mathcal{P}_{ij}} l_{\tilde{P}}^{-\lambda}}. \tag{3.2}$$

Though we expect most agents to drive on admissible paths, some agents may choose routes deemed inadmissible. We account for that possibility by assuming that agents choose inadmissible routes with a small probability η_{c}. As these agents could choose any path through the road network, it is difficult to estimate the probability to observe such agents at a specific survey location. In the absence of a "good" model and considering that only few agents choose inadmissible routes, we assume that any survey location could be on any inadmissible route with probability η_{o}, respectively. In summary, the probability that an agent travelling from i to j passes a survey location k is

$$\rho_{ijk} \;=\; \underbrace{(1-\eta_{\mathrm{c}})}_{\substack{\text{prob. to choose} \\ \text{an adm. route}}} \; \underbrace{\sum_{P \in \mathcal{P}_{ij} : k \in P}}_{\substack{\text{sum over all} \\ \text{adm. routes via } k}} \; \underbrace{\frac{l_P^{-\lambda}}{\sum\limits_{\tilde{P} \in \mathcal{P}_{ij}} l_{\tilde{P}}^{-\lambda}}}_{\substack{\text{prob. to} \\ \text{choose route } P}} \;+\; \underbrace{\eta_{\mathrm{c}}\eta_{\mathrm{o}}}_{\substack{\text{prob. to be observed} \\ \text{on inadm. route}}}. \tag{3.3}$$

3.2.3 Temporal pattern model

The numbers of agents observed in road-side surveys vary in temporal patterns. Traffic may fluctuate in daily, weekly, and seasonal cycles and depend on the survey location, because agents will reach locations far away from their starting points later than locations close to their origins. In this study, we focus on daily patterns to keep the model simple. Furthermore, we assume that the temporal traffic pattern is independent of the survey location, because starting time, travel speed, and overnight breaks vary among agents. The complex interplay of these factors makes it difficult to model traffic patterns mechanistically. Therefore, it is appropriate to use a simple phenomenological traffic pattern model.

Unimodal cyclic distributions constitute a good first approximation to daily traffic patterns, since traffic is denser during the day than during the night, in general. A commonly used unimodal cyclic distribution is the von Mises distribution (Lee, 2010). This distribution resembles a normal distribution and takes a location parameter, determining the traffic peak time, and a scale parameter, controlling how "spiky" the peak is. Other distributions can be used if traffic is expected to follow a more complex pattern, but we will proceed with the von Mises distribution due to its simplicity and intuitive shape.

3.2.4 Compliance model

The number of agents stopping to be surveyed may depend on their origin and destination, the time of day of the survey, and the setup of the survey location. For example, more agents may stop if the survey location is clearly visible or if compliance can be enforced. If required, the compliance rate could be measured for each survey location individually. However, to keep the model simple, we assume that the probability that an agents chooses to participate in the survey – and provide complete and correct data – is constant across agents, survey time, and survey locations.

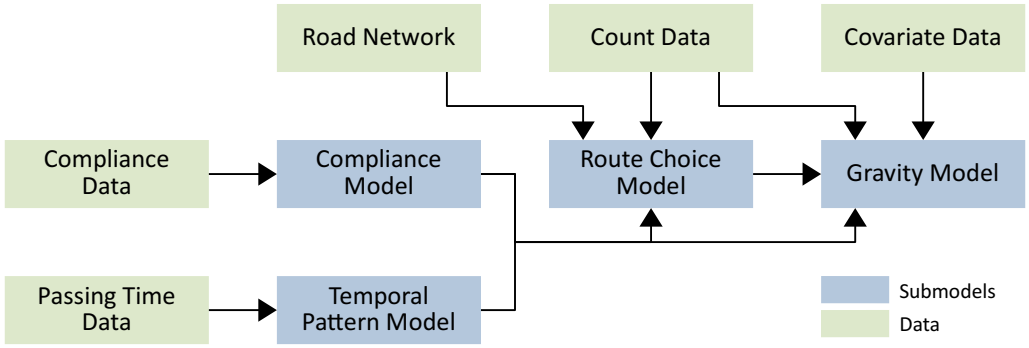


Figure 3.3: Overview of the model fitting procedure. The green rectangles depict data; the blue rectangles depict submodels. The arrows show which components are needed to fit the three submodels, respectively.

3.3 Model fit

In the previous section, we described a hierarchical model for the number of agents observed in a road-side survey shift. In this section, we show how such survey data can be used to fit the model.

We fit the four submodels in the order inverse to the hierarchy. That is, we start with the compliance model and the temporal pattern model, proceed with the route choice model, and end with the gravity model (see Figure 3.3). Before we describe the fitting procedures in detail, we give an overview of the data required to fit the model.

3.3.1 Required data

We need five data sets to fit our hybrid model: (1) a count data set, (2) a compliance data set, (3) a survey time data set, (4) a covariate data set, and (5) a graph representation of the road network with edges weighted by length or travel time. The count data set contains the start and end time of each survey shift, the respective survey location, and how many agents were surveyed driving from each origin to each destination. Most of these count values will be zero, especially if many origin-destination pairs are considered. The compliance data set contains the total number of agents who passed the survey locations and the number of agents who participated in the survey and provided complete data. The survey time data

set encompasses the times of day when agents were surveyed and the start and end times of the respective survey shifts. The covariate data set contains information related to the outbound and inbound traffic volume at origins and destinations. For example, this could be the population counts for the source locations or the number of close-by tourist attractions for the destination locations. Lastly, we require a graph representation of the road network we consider. Roads translate to edges, weighted by the roads' respective lengths or the time required to drive along the roads. The set of vertices consists of all junctions of the road network as well as the origins and destinations of the agents. All survey locations, origins, and destinations must correspond to specific vertices or edges in the graph. Collectively, the five data sets are shown by the green rectangles in Figure 3.3.

3.3.2 Fitting the compliance model

The compliance model measures which proportion of agents is expected to stop at a survey location and to provide complete data. The model can be fitted in a single step by dividing the number of agents who provided useful data by the total number of agents who passed the survey locations. However, in some applications, the origin of agents, and thus their potential of being a vector, can be determined easily once they have stopped for the survey. This could be done, for example, by using license plate information. In this case, only the data provided by agents from infested jurisdictions need to be checked for integrity, and the overall compliance rate ξ may be obtained by estimating the participation rate ξ_{p} and the complete data rate ξ_{c} separately.

We compute the rate ξ_{p} using count data of how many agents stopped at survey locations and how many agents passed these locations without stopping. The estimated participation rate ξ_{p} is given by the number of agents who stopped divided by the total number of passing agents:

$$\xi_{\mathrm{p}} \quad = \quad \frac{\#\text{agents stopped}}{\#\text{agents stopped} + \#\text{agents bypassed}}. \tag{3.4}$$

Similarly, we compute the complete data rate ξ_c as

$$\xi_c = \frac{\#\text{high-risk agents providing complete data}}{\#\text{high-risk agents stopped}}. \tag{3.5}$$

The overall compliance rate ξ is the product of the two rates:

$$\xi = \xi_p \xi_c. \tag{3.6}$$

3.3.3 Fitting the temporal pattern model

The temporal pattern model accounts for the temporal variations in the traffic density. When we fit this model, we have to take into account that the survey shifts in which the data were collected do not cover all times of day equally well, in general. For example, if most surveys were conducted in the morning, our data set would contain a disproportionate number of agents observed in the morning, even if the true traffic peak were during the afternoon. To avoid the resulting bias, we fit our model with a maximum likelihood approach based on the conditional likelihood, which takes into account when the surveys were conducted. We provide details in Appendix 3.D.

3.3.4 Fitting the route choice model

The route choice model specifies the probabilities that agents take specific routes. As with the temporal pattern model, we fit the route choice model based on the conditional likelihood. Usually, it is infeasible to monitor all potential routes of agents at once, and surveyors have to focus on a small set of routes. To ensure that our choice of survey locations does not bias our results, we fit the route choice model by maximizing the likelihood conditional on which routes we monitored for how long.

There are several practical challenges associated with fitting the route choice model. These challenges are not only due to the computational complexity of the task but also due

to identifiability problems, which could lead to non-informative results. In Appendix 3.D we provide more details of these challenges and show how the issues can be resolved.

3.3.5 Fitting the gravity model

The gravity model estimates how many agents are driving from each origin to each destination per time unit. We fit the model by maximizing the composite likelihood (Besag, 1975). The difference with classical likelihood estimation is that we make an approximation via independence assumptions so as to facilitate straightforward computation.

When we fit the gravity model, we exploit that the number N^+_{ijkt} of surveyed agents is negative binomially distributed (Villa and Escobar, 2006). This simplifies the model fit, as the likelihood function can be written down easily. Nonetheless, computing the likelihood is computationally costly, because each survey shift yields a count value for each origin-destination pair. In Appendix 3.D we present an algorithm to speed up the computations by orders of magnitude.

3.4 Application

In the previous sections, we outlined the hybrid gravity, route choice, temporal pattern, and compliance model and described how it can be fitted to data. Now we demonstrate our approach by applying it to the potential invasion of zebra and quagga mussels *Dreissena spp.* to the Canadian province British Columbia (BC).

3.4.1 Methods

We fit the hybrid model with survey data collected by the BC Invasive Mussel Defence Program. The survey data were obtained during 1571 inspection shifts at 31 locations in BC over the course of the years 2015 and 2016. All shifts were conducted during day time. As small boats present a lower risk of being fouled by dreissenid mussels, we counted only

medium to large motorized watercraft (e.g. cabin cruiser, wakeboard boats, speed boats, car toppers) as potential mussel vectors.

By provincial law, it was mandatory for boaters to stop at the survey locations. Nonetheless, not all boaters complied with this provision. We counted the number of bypassing boaters in 293 of our survey shifts. As it is difficult to determine the type of bypassing towed boats precisely, we did not distinguish between boat types when estimating the participation rate. However, the proportion of boaters providing complete data was determined with respect to high-risk boaters only.

We identified 5981 potentially boater accessible lakes in British Columbia and considered them as potential destination points for the boaters. As origins we included the Canadian provinces and territories and the American states of the North American mainland. We treated a state or province as potential zebra and quagga mussel donor if either (1) there was a confirmed dreissenid mussel detection in a waterbody within the jurisdiction or (2) if the jurisdiction (2a) had a connected waterway with a dreissenid mussel infested lake in a neighbouring state or province and (2b) did not have an established dreissenid mussel monitoring program at the time at the time the data were collected. All remaining source jurisdictions were used to fit the model but ignored when we assessed potential propagule transport.

We fitted a gravity model with the population number and number of registered anglers as proxies for the repulsiveness of donor jurisdictions. To estimate lake attractiveness, we considered the lake area, the lake perimeter, the presence of marinas, campgrounds, and other facilities (including public toilets, tourist information, viewpoints, parks, attractions, and picnic sites) in a 500 m range of the lakes, and the population living in 5 km ranges around the lakes. To measure distances and compute potential routes, we used a road network with edges weighed based on travel time. We provide further details of the data, including a list of the data sources, in Appendix 3.C.

We used a model selection criterion to determine which covariates our model should include to fit the data well without overfitting. Contrasting the criterion by Akaike (AIC) and the Bayesian information criterion (BIC), Ghosh and Samanta (2001) point out that AIC is to be preferred if the goal is to provide precise predictions. Therefore, we chose our model based on AIC. See Appendix 3.E for a more in-depth discussion of model selection.

Our model candidates incorporated a large number of covariates. Therefore, it was not feasible to check all possible combinations of covariates, parameters, and functional forms of the building blocks. Thus, we ignored models with few covariates after noting that they led to much larger AIC values in general.

To get a sense of the credibility of our parameter estimates and check for identifiability issues, we determined confidence intervals for the model parameters with the method that will be introduced in chapter 5 (see also our notes on composite likelihood based confidence intervals in Appendix 3.E). Furthermore, we tested our base hypotheses on boater counts and the temporal traffic pattern and assessed the accuracy of our model. Details can be found in Appendix 3.G.

3.4.2 Results

In this section, we provide information on the fitted submodels and show results on the compliance rate, the temporal traffic distribution, the sources of high-risk boaters, the boater inflow to threatened lakes, and the boater traffic through the road network.

3.4.2.1 Resulting models

The participation rate was estimated to be 80%. That is, only a fifth of the boaters passed the survey locations without participating in the survey. 93% of the boaters driving to BC from other jurisdictions provided complete data. The estimated overall compliance rate is thus 74.4%.

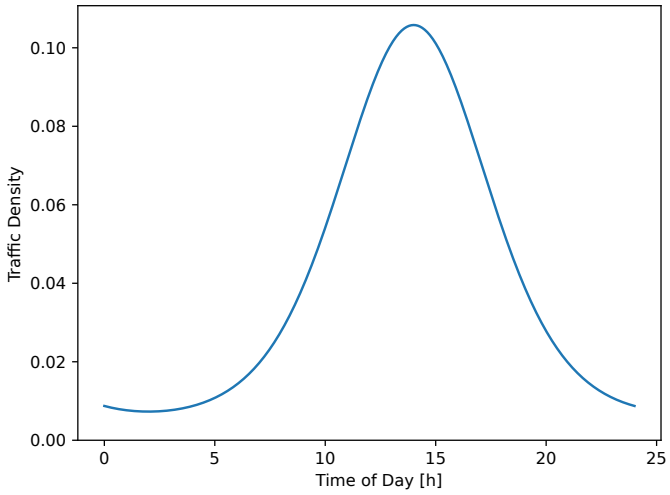


Figure 3.4: Traffic profile. The line depicts the probability density function modelling the time when boaters pass survey locations.

Our fitted traffic pattern model has the traffic peak at 2 : 00 PM. Thereby, the estimated boater traffic is about 15 times higher during the peak time than at night. The probability density function of the temporal pattern model is plotted in Figure 3.4.

The fitted route choice model suggests that boaters have a strong preference for the shortest route. According to the model, an alternative route only 10% longer than the shortest route attracts only half as many agents.

The gravity model with minimal AIC value estimates the repulsiveness m_i of source jurisdictions based on their population count and nation. Canadian provinces were weighed about 15 times as high as American states. The submodel for the lake attractiveness a_j included all available covariates except for the lake perimeter, whereby the presence of a marina and a large population close to a lake had the highest weight. The travel times between jurisdictions and recipient lakes had a huge effect on the expected numbers of travelling boaters. Numbers decreased in cubic order of the travel time.

In Appendix 3.F, we provide further details of the fitted model and present parameter estimates and confidence intervals.

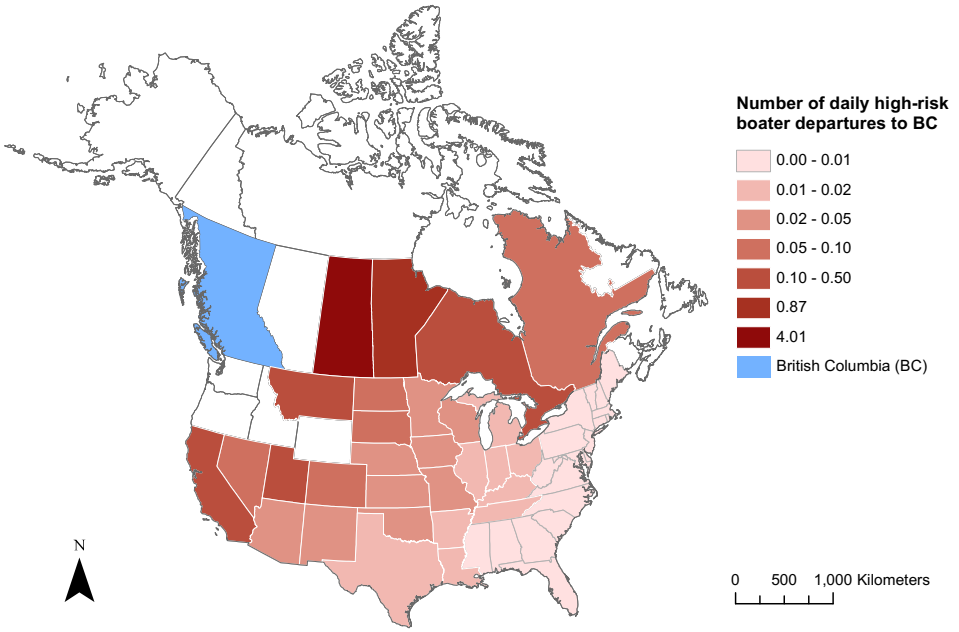


Figure 3.5: Potential donor regions of dreissenid mussels. The red shading depicts how many boaters are estimated to drive from the jurisdictions to BC each day.

3.4.2.2 Propagule transport

Donor regions

According to our model, most of the external boaters driving to BC come from Alberta (71%) and Washington (19%). However, we did not consider these jurisdictions as potential propagule donors. The most significant sources of high-risk boaters were Saskatchewan (4.3% of the total inflow) and Manitoba (1%). Note that we treated Saskatchewan as a *potential* donor of dreissenid mussels even though no dreissenid mussels have been found in the province to date (see section 3.4.1). In total, the Canadian provinces were contributing more than three times as many high-risk boaters as the American states. In Figure 3.5, we depict the respective contributions of the potential donor regions.

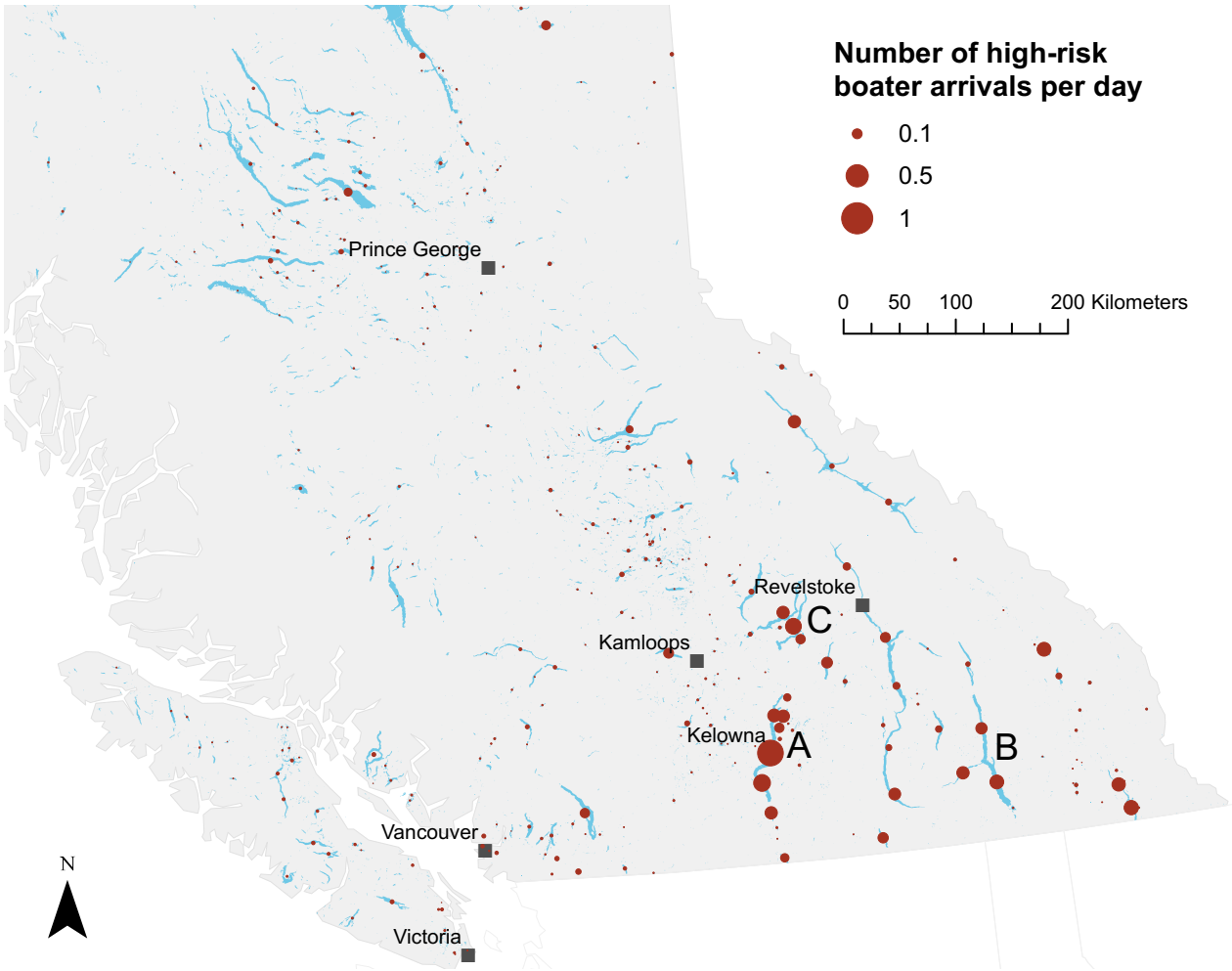


Figure 3.6: Daily arrivals of potentially infested boats at British Columbian lakes. The sizes of the red circles correspond to the respective arrival counts. Subsections of large lakes are treated as separate lakes to allow for a higher spatial resolution. The letter labels correspond to the three lakes with the highest boater inflow (summed over all subsections): (A) Okanagan Lake, (B) Kootenay Lake, (C) Shuswap Lake.

Boater pressure to lakes

The inflow of high-risk boaters concentrates on few lakes in BC. The 9 most-frequented lakes receive 50% of the total high-risk boater pressure; the top 157 lakes receive 90% of the total high-risk boater pressure. The lakes attracting most high-risk boaters were Okanagan Lake (received 17% of all high-risk boaters), Kootenay Lake (7.5%), and Shuswap Lake (6%). These lakes are large and located in the populated southern part of BC. See Figure 3.6 for a map showing the high-risk boater arrivals for the British Columbian lakes.

Most frequented roads

In Figure 3.7, the high-risk boater traffic is mapped onto the highway network of BC. The traffic concentrates on a small set of major roads accommodating traffic to clusters of many or highly attractive lakes. Thereby, the roads crossing the eastern border of BC, in particular the Trans-Canada Highway, have the highest boater counts.

3.5 Discussion

We presented a hybrid gravity, route choice, temporal pattern, and compliance model to assess traffic flows in realistic continent-sized road networks. The hybrid model can be used to estimate the agent outflow of donor regions, the agent traffic volume on roads, and the arrival counts of agents at recipients. We provided both a general framework for building traffic models based on field traffic survey data as well as a set of directly applicable submodels. We demonstrated the applicability of our approach by studying the inflow of potentially mussel-infested boats to the Canadian province British Columbia.

Combining a gravity, route choice, temporal pattern, and compliance model has two major advantages: data can be collected and used more efficiently, and the combined models yield more information than the submodels individually. First, data collected at few locations in the road network can be used to draw inference on the traffic between many origin-destination

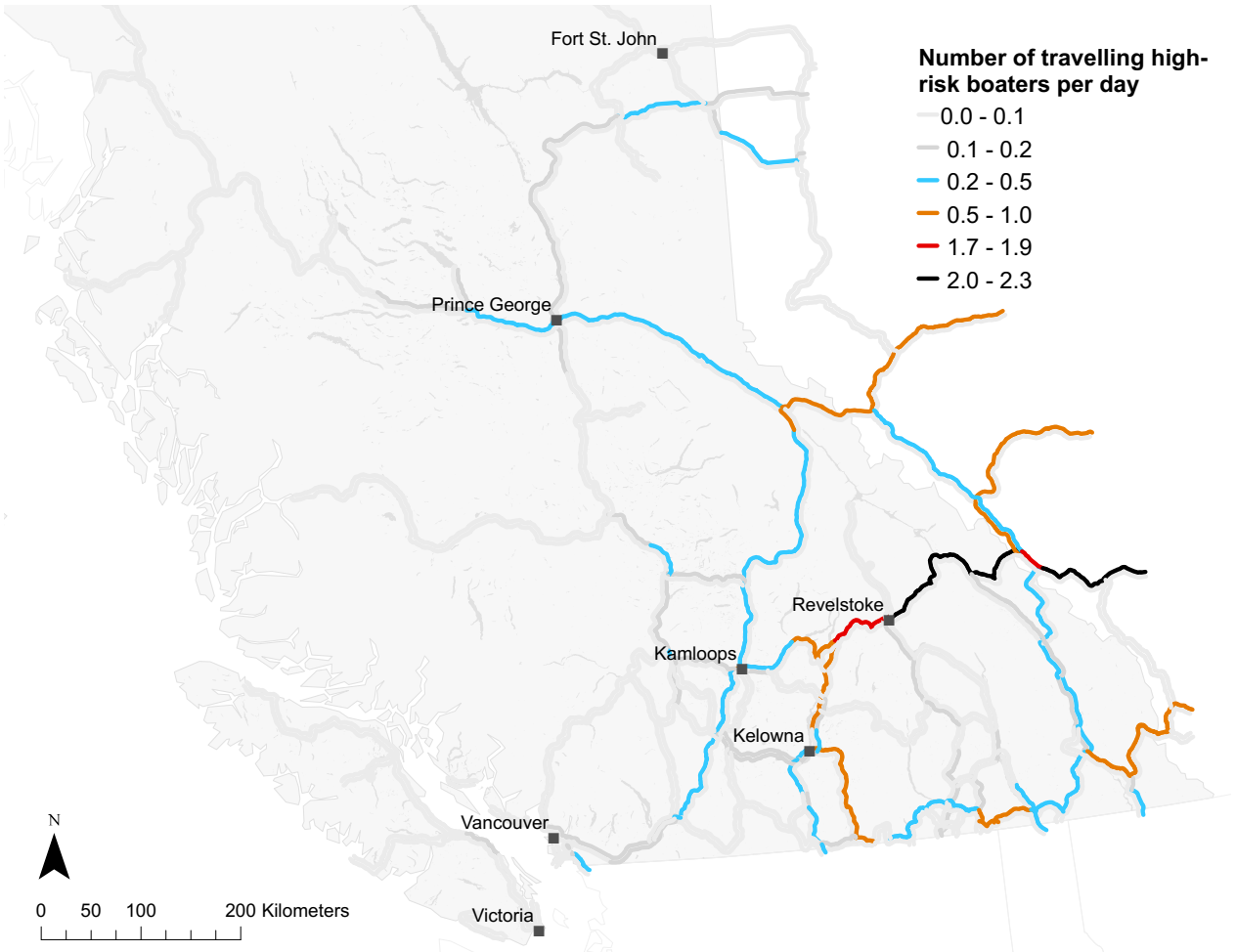


Figure 3.7: Traffic of potentially infested boats along major British Columbian roads. The colours correspond to the expected daily numbers of travelling boaters. The roads' lanes are coloured separately to depict the traffic in different driving directions.

pairs at once. This makes it possible to assess traffic even in continent-scale road networks. Second, neither a gravity model nor a route choice model alone could provide estimates of how many agents travel along a specific road. A model predicting *how many* agents drive is required as much as a model predicting *where* these agents drive. Thus, our combined approach is more powerful than sequential individual modelling efforts.

Various data sources have been used to fit gravity models in ecology. However, as will become apparent below, these data sources have considerable limitations in many scenarios.

Most studies in ecology are based on data gathered in mail-out surveys (e.g. Drake and Mandrak, 2010; Potapov et al., 2010; Chivers and Leung, 2012). Though this is often the

easiest method to gather data to parameterize gravity models, mail-out surveys are subject to significant sampling error, in particular if only few of the surveyed potential travellers actually start a trip. Furthermore, mail-out surveys can only yield relative traffic estimates, unless further data are available to calibrate the model.

In other studies, gravity models are fitted with survey data collected at a small sample of origin or destination locations (Bossenbroek et al., 2007). Similar to mail-out surveys, these data are prone to sampling error. In addition, special care has to be taken to ensure that the sample of origins or destinations is representative. Otherwise, the data will lead to biased estimates.

In some rare cases, traffic data can be obtained from booking systems at the destination locations (Prasad et al., 2010). This data source is among the best possible foundations for fitting gravity models. However, data from booking systems are often not available, especially in large-scale systems, in which each destination may cover a large area.

Lastly, some studies in invasion ecology combine a gravity model with an establishment model, which maps the output of the gravity model to invasion probabilities. Then, the joint model is fitted to data of the temporal progression of the considered invasion (Bossenbroek et al., 2001; Leung et al., 2004; Mari et al., 2011). This approach can be taken only if the invasion has already progressed sufficiently far and the temporal progression of the invasion is known. Furthermore, this method may not yield concrete estimates of the traffic flows, because some traffic-related parameters may remain unidentifiable if gravity and establishment model are fitted simultaneously (Leung et al., 2004). Consequently, a combined gravity and establishment model is useful only in specific cases.

In conclusion, road-side surveys are often better suited for fitting gravity models than the data sources commonly used to date. The hybrid gravity and route choice model makes these road-side survey data available for fitting gravity models.

Though the presented model for the transport of propagules or pathogens in large-scale systems is new, other studies have considered large-scale invasions before (Bossenbroek et al.,

2007; Mari et al., 2011). These studies reduce the need for survey data by making strong assumptions on the drivers of repulsiveness and attractiveness. However, the models may suffer from inaccuracy, since large parts of the models are fitted without survey data. In fact, errors resulting from the additional assumptions cannot even be measured, because no data are available to validate the models rigorously. Furthermore, the added assumptions also decrease model portability (Potapov et al., 2010). Thus, the hybrid model, fitted with actual survey data, has strong advantages over earlier large-scale models, which were largely based on strong assumptions without data.

3.5.1 Applications

The primary purpose of the hybrid model presented in this paper is to study the traffic of agents potentially carrying propagules or pathogens. If the travel behaviour of these agents is known, early detection and control actions can be implemented more effectively. Thus, the hybrid model can help managers to control invasions and infectious diseases.

First, the hybrid model can facilitate early detection of invasions and infections by providing estimates of the number of potentially infested agents arriving at susceptible locations. These estimates are a valuable proxy for propagule or pathogen pressure and have been used to estimate invasion or infection risk (Bossenbroek et al., 2001; Prasad et al., 2010; Barrios et al., 2012). These risk estimates, in turn, could be applied to allocate early detection effort and rapidly deploy resources to the locations that are threatened most.

Second, the hybrid model's estimates of agent traffic along roads can be used to decrease invasion or infection risk before infestations occur. For example, invasive species managers in BC set up watercraft inspection stations on roads to detect and treat mussel-infested trailed watercraft. Since most long-distance traffic concentrates on a small number of roads, it is much more efficient to apply such control measures on intermediate roads rather than at the access points of susceptible locations. Our hybrid model could be used to facilitate the choice of optimal control locations.

When using the hybrid model to find optimal control locations, it is helpful that the model does not only estimate the agent traffic at all considered roads but also predicts how control applied at one road affects the remaining propagule or pathogen flow at other roads. As a consequence, the hybrid model has the potential to aid management much better than simple traffic measurements on roads, the momentarily common method to identify good control locations.

Besides facilitating management of invasions and infectious diseases, the hybrid model could also lead to a more comprehensive general understanding of human-aided dispersal of species. As the hybrid model focuses on agents that have the potential to carry several invasive species, it would be possible to investigate the dispersal of multiple species with a single modelling effort. The option to incorporate many origin-destination pairs with relatively low survey effort would allow comprehensive studies. This could help ecologists to gain a deeper understanding of the dispersal of both native and invasive species and to assess the impact of road traffic on ecosystems.

3.5.2 Limitations

Since the hybrid model involves four submodels for specific agent decisions, it has a considerable level of complexity, which we aimed to reduce by using simple submodels. As a consequence, some of the proposed submodels may seem unrealistic. Nonetheless, we argue that the proposed models provide valuable insights despite their limitations.

First, we assumed that the compliance of agents is independent of when and where the survey is conducted and who is surveyed. However, in particular the survey location can play a major role for the compliance of agents. For example, more agents may participate in the survey at a boarder crossing, where they all travellers have to stop. However, we chose our survey locations carefully with proper signage, and compliance was mandatory. This decreases the variations of the compliance rates.

Second, we accounted for temporal traffic variations with a simple two-parameter model. Thereby, we ignored weekly and seasonal traffic patterns and assumed that the temporal traffic distribution is independent of the sampling location. In reality, traffic is likely to follow more complex patterns. However, even if the fitted temporal traffic distribution does not match the data perfectly, the introduced error will be small, unless the model is very far from the real traffic pattern. Furthermore, the overdispersion resulting from not properly modelled weekly and seasonal traffic patterns is phenomenologically accounted for with the negative binomial distribution. Therefore, our simple temporal traffic pattern will yield generally accurate estimates, even though estimates resulting from a more sophisticated model could be more precise.

Third, we assumed that agents base their route choices solely on expected travel time, and we ignored potential issues arising from overlapping admissible routes (Cascetta et al., 1996). In addition, our noise traffic model, accounting for agents travelling along inadmissible paths, allows unrealistic disconnected routes. All these issues could be resolved by using more sophisticated submodels. However, modelling routing decisions more realistically could make further data necessary, and the model fit would become computationally harder. We believe that our route choice model constitutes a good first approximation of routing decisions.

Fourth, we made several approximations via independence assumptions. These assumptions decrease the meaningfulness of confidence intervals and model selection criteria (see Appendix 3.E). Nonetheless, parameter estimates remain unbiased (Lindsay, 1988), while the gain of computational efficiency resulting from the independence assumptions is considerable. In fact, accounting for all potential dependencies could make the model fit computationally infeasible. Therefore, the independence assumptions may be a necessary concession to computational efficiency.

The precision of the hybrid model is strongly dependent on how well the available covariates describe attractiveness and repulsiveness of origins and destinations. Due to this limitation, the differences between predictions and observations were larger than expected

for our boater traffic model (see Appendix 3.G). However, model accuracy is always dependent on the explanatory power of the used data. Therefore, it is unlikely that a different model based on the same data would yield significantly more precise estimates.

Note that though a more precise model would be desirable, the rigorous model validation that revealed our model's inaccuracies would have been hardly possible without the comprehensive survey data made available through the hybrid approach. For example, mail-out surveys are typically designed as cross-sectional studies. Solely based on these data, it is difficult to determine whether differences between model predictions and observations are due to random processes or due to a poorly fitting model. A longitudinal study, such as repeated collection of count data at road sides, is required to discern between prediction error and stochasticity inherent to the modelled system.

Given that existing models could not be validated as rigorously as ours, we do not have evidence that our hybrid model of boater traffic is less accurate than similar models presented earlier. Quite the contrary, the hybrid model could make a contribution to reveal hidden shortcomings of commonly used models.

3.5.3 Future Directions

A strength of our approach is in its flexibility. The model fitting techniques that we presented in this paper remain applicable if submodels are exchanged or added. Therefore, we hope that future research will build on this study and develop adjusted and refined submodels to tackle different problems in invasion ecology and epidemiology.

The increased amount of survey data made usable by our approach can also lead to new methodological results. The newly available survey data may allow modellers to incorporate more covariates in gravity models and use more effective methods to draw inference from the covariates. For example, machine learning techniques could be used to compute repulsiveness and attractiveness of origin and destination locations more accurately. This could lead to traffic models with a new level of predictive quality.

Additional data could be used to fit more sophisticated models for compliance, temporal traffic patterns, and route choice. Compliance rates could be estimated for each survey location independently. Furthermore, the conditional likelihood method presented in this paper could easily be extended to fit a temporal traffic pattern model accounting for weekly and seasonal cycles. Alternatively, a gravity model with a temporally variable mean could be used. Route choice probabilities could be computed based on a variety of route characteristics, such as the scenery or the number of sights along a route (see e.g. Alivand et al., 2015). With such improvements, the model could become more accurate.

New and more precise ways of fitting the gravity model could be developed if cell phone tracking data of agents are available. Such data could not only yield precise measures of relative count data but also be used to fit a more realistic route choice model, potentially even without computing admissible routes first (Ton et al., 2018). With such improvements, agent traffic could be predicted and understood more precisely.

The results on agent flows computed with the techniques presented in this paper open new possibilities for optimizing invasion and disease control measures. If agent traffic flows are known, methods from optimal control theory could be used to improve control strategies and determine locations where control measures are most effective. Consequently, this study provides the prerequisites for a number of highly relevant management problems.

Appendices

3.A Modelling assumptions

Below we provide a comprehensive list of our modelling assumptions.

1. For each time unit, the number of travelling agents N_{ij} is given by a stochastic gravity model.

2. Each time unit, the number N_{ij} is drawn from a negative binomial distribution. Thereby, the numbers N_{ij} and N_{kl} for origin-destination pairs $(k,l) \neq (i,j)$ are independent of each other and of the past.

3. The distribution of N_{ij} is independent of the spatial scale at which we consider the system.

4. Agents choose their routes randomly and independently of each other.

5. Most agents drive along a set of "admissible" routes. This route set should encompass all "major alternatives" that agents choose from.

6. A route is admissible if it does not contain local detours and is not much longer than the shortest route from the origin to the destination.

7. The probability to choose an admissible route is inverse proportional to a power of its length.

8. All agents who are not driving along an admissible route can be observed everywhere in the route network with the same probability.

9. Agents choose randomly the time of day when they pass a location on their route.

10. The distribution of the time of day when an agent passes a certain location is independent of the location, the origin and destination of the agent, earlier time choices of the agent, and other agents' timing.

11. The temporal pattern determining when agents pass a survey location is a von Mises distribution.

12. Agents choose randomly and independently of each other whether or not they participate in the survey.

13. The compliance rate is independent of the respective agents, the time when they pass the survey location, and the location of the survey.

3.B Scale-invariant count distributions

A desirable property of spatial models is scale invariance. In the case of gravity models this means that the distribution of the number of trips starting or arriving at a region i should not change if we increase the spatial resolution and considered subregions i_1 and i_2 instead of i. That is, we require $N_{i_1 j} + N_{i_2 j} \stackrel{d}{=} N_{ij}$. See figure A3.1 for a depiction of the considered scenario.

If the agent counts in the subregions are independent of each other, Poisson random variables satisfy this condition always. However, independent negative binomial random variables satisfy this property only if they have the same mean to variance ratio, $p = \frac{\mu}{\sigma^2}$. This ratio measures the level of overdispersion of the distribution.

Following the claim that the gravity model is scale invariant and assuming that the agent counts in subregions are independent, we have to choose the mean to variance ratio p independently of origins and destinations. Hence, it makes sense to parameterize the negative binomial distribution in terms of the mean μ and the parameter p. Then, the probability mass function of N_{ij} reads

$$\mathbb{P}(N_{ij} = n) \quad = \quad \binom{n + r_{ij} - 1}{n} p^{r_{ij}} (1-p)^n \tag{A3.1}$$

with $r_{ij} := \frac{p}{1-p}\mu_{ij}$.

Note that scale invariant distributions are also invariant against how locations are pooled together. Since large regions can be split in smaller regions without changing the cumulative distribution of the count data, the same applies also when smaller regions are connected to larger regions. Consequently, origin and destination regions can be chosen based on practical considerations without the risk of introducing a bias.

The negative binomial distribution can be understood as a Poisson distribution with a gamma distributed rate. That is, we could write

$$N_{ij} \quad \sim \quad Poisson\left(\mu_{ij}\lambda\right),$$

whereby λ is a gamma distributed random variable with mean 1. The random rate λ models that all agents' travel decisions may depend on common unknown factors, such as weather. If we hold the mean to variance ratio p constant, this implies that the variance of the rate $\mathbb{V}(\lambda) = c\mu_{ij}^{-1}$ is inverse proportional to the mean number of travelling agents μ_{ij} with some proportionality constant c. This can be interpreted as a phenomenological model for mechanisms that reduce the variance of travelling agents at highly frequented destinations, where limitations of accommodations and other facilities may play a major role in reducing the variance of the agent inflow. If the model did not account for these factors, the model might predict an exceeding variance for count data from highly frequented locations.

3.C Details of the data

This appendix contains details of the data used in the application section of this study.

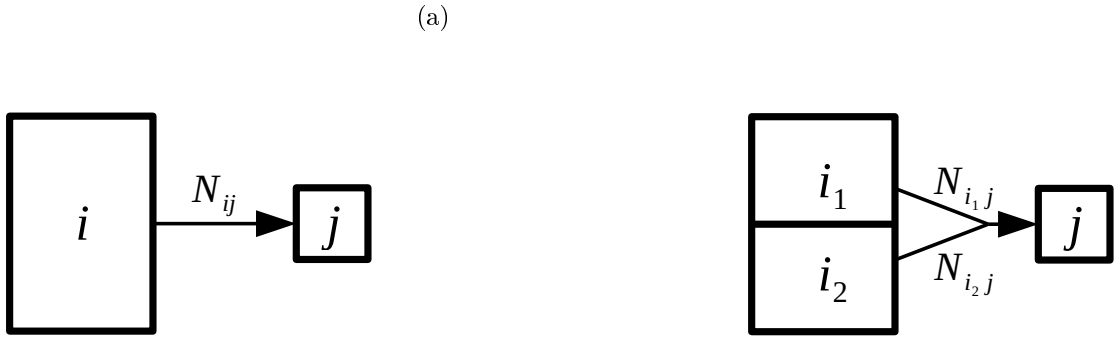


Figure A3.1: Scale invariance property. The total flow N_{ij} from a source i to a sink j (Panel a) shall not change if we split the source region into two subregions i_1 to i_2 and consider the two flows $N_{i_1 j}$ and $N_{i_2 j}$ (Panel b). Thus, $N_{i_1 j} + N_{i_2 j}$ has the same distribution as N_{ij}.

3.C.1 Data sources

The sources for the data used in this study are displayed in Table A3.1.

3.C.2 Variable spatial resolution

We used data with variable spatial resolution. Often it is hard to find or collect data with high spatial resolution. Similarly, incorporating high-resolution data in models can come with considerable computational challenges. However, typically, a high spatial resolution is only required in certain areas of interest. Consequently, it is advisable to use data with a resolution that is high in the area of interest and low elsewhere.

Following this principle, we used a detailed road data set for BC and a sparse data set for the rest of Canada and the USA, because all survey locations and all roads of management interest were located in BC. The sparse road network contained highways only. In total, our road network consisted of 1.4 million vertices and 1.6 million edges.

To get a better spatial resolution of the boater origins close to BC, we split the province Alberta into three parts (north, middle, south) and the state Washington into an eastern and a western part. Some lakes in BC span hundreds of kilometres. This can make it difficult to determine the best access routes if the lakes have far-apart access points. Therefore, we

Data	Source	URL
Boater Survey Data	BC Ministry of Environment	https://www2.gov.bc.ca/gov/content/invasive-mussels
Base GIS Data (e.g. road network, lake data, borders)	BC Ministry of Environment	https://catalogue.data.gov.bc.ca/dataset
Angler Count Data Canada	Department of Fisheries and Oceans Canada	www.dfo-mpo.gc.ca/stats/rec/can/2010/section4-eng.htm
Angler Count Data USA	American Sportfishing Association	asafishing.org/wp-content/uploads/Sportfishing_in_America_January_2013.pdf
Population Data Canada	Statistics Canada	www.statcan.gc.ca/eng/start
Population Data USA	U.S. Census Bureau	www.census.gov
Locations of Cities	Open Street Map	www.openstreetmap.org
Facilities (public toilets, tourist information, viewpoints, parks, attractions, and picnic sites)		
Campgrounds	USCAmpgrounds	www.uscampgrounds.info
	British Columbia Lodging and Campgrounds Association	www.campingrvbc.com/camping/
Marinas	Manual web search for marinas in BC	–

Table A3.1: Data sources.

checked the access routes to all lakes with a perimeter larger than 100 km and split the lakes that were accessible via multiple substantially different routes.

3.C.3 Data accuracy

In this section, we discuss the accuracy of the data we used.

3.C.3.1 Survey data

The destinations of some surveyed boaters were not perfectly clear. The surveyed boaters were asked for their destination waterbodies and close-by cities. As not all lakes in BC have unique names and cities are rare in some regions of BC, we had to use common sense to deduce which lakes boaters went to, when the destinations were ambiguous. Thereby, we took into account the properties (size and available facilities) of the potential destination lakes and considered where the boaters were surveyed. As the data were unambiguous for highly frequented lakes, only a small fraction of the data were affected by the cleaning step. Nonetheless, for the lakes that we split due to their large size (see section 3.C.2), some boater destinations may have been misclassified. Though these errors may result in skewed estimates of how many boaters use which section of a large lake, the errors should not have a major affect on the arrival estimates for the complete lakes. Clearly inconsistent or incomplete data were used only to determine the rate at which boaters provide trustworthy and complete data.

While erroneous and ambiguous data could be reduced by providing agents with a comprehensive list of possible destination locations, a second problem arises if surveys are conducted close to destination regions with multiple access points. A considerable number of agents accessing these recipients may not pass the survey location if they are using other access points. Furthermore, the route choice model will be imprecise close to destination points unless they are not known exactly, which is often not the case. Consequently, data collection

in direct proximity of destinations with multiple access routes can yield unreliable results. This issue may also have affected our study.

3.C.3.2 Covariate data

As we collected the covariate data from external sources, we do not have specific insights on their accuracy. Note, however, that the angler data we used were collected by different agencies in Canada and the USA. This can lead to a bias if the classification of anglers is different in the two countries. We sought to reduce the potential resulting error by including the nation of source jurisdictions in the model.

3.D Details of the model fit

In this Appendix, we provide details of how to fit the four submodels of the hybrid model and compute the likelihood functions efficiently. Furthermore, we outline the likelihood maximization procedure. Though we describe all important conceptual steps, we do not provide implementation details.

To make our explanations more understandable, we choose a specific time unit for this appendix. This contrasts with the main text, where we have have formulated our model in terms of a general time unit and left it up to the modeller to decide whether it is appropriate to model traffic as a repetitive process running in daily, weekly, or other cycles. Though we choose "days" as our time unit for this appendix, all the described methods apply without further limitations if a different time unit is used instead of days.

Slight adjustments to the presented equations may be necessary if multiple survey shifts are conducted during one time interval. In this appendix, we assume that at most one survey shift is conducted at a location per day. If it is possible that multiple, disjoint survey shifts are conducted in a time interval, the notion of "survey time interval" has to be replaced by "survey time set", and the computation of probabilities has to be adjusted accordingly.

However, these adjustments concern only simple probability calculations and should be clear from the context.

3.D.1 Fitting the compliance model

No sophisticated techniques are required to determine the compliance rate. To determine the participation rate, we simply determine the number of surveyed agents and divide it by the total number of agents passing our survey location. However, as it is typically impossible to know origin and destination or other properties of bypassing agents, the number of surveyed boaters should not be filtered by origins and destination or any other characteristics. Hence, it is important to record the compliance of agents that may not be of interest, unless these agents can be clearly distinguished from agents of interest without surveying them.

We proceed similarly to determine the proportion of agents that provide consistent and complete data. However, the origins of agents can often be determined easily based on license plate information once the agents have stopped for the survey. Hence, the complete data rate can be determined focusing on the agents of interest only.

3.D.2 Fitting the temporal pattern model

The temporal pattern model describes the distribution of traffic over the day. When we fit this model, we have to recall that our sampling effort is not uniformly distributed over all day times. Therefore, we have to fit the model using the conditional likelihood.

Let T_i be the random variable describing when the i-th agent passes a survey location, and let $\left[t_i^{\text{start}}, t_i^{\text{end}}\right]$ be the time interval of the survey shift in which agent i was observed. As we can only observe agents who pass our location while we conduct the survey, $T_i \in \left[t_i^{\text{start}}, t_i^{\text{end}}\right]$ must hold for all agents i in our data set. Consequently, if f_{Time} is the probability density function of the temporal pattern model and F_{Time} the respective cumulative density function,

the likelihood function for our temporal pattern model reads

$$\begin{aligned} L_{\text{Time}}\left(\theta_{\text{Time}}\right) &= \prod_i f_{\text{Time}}\left(t_i^{\text{obs}} | \theta_{\text{Time}},\, t_i^{\text{obs}} \in \left[t_i^{\text{start}}, t_i^{\text{end}}\right]\right) \\ &= \prod_i \frac{f_{\text{Time}}\left(t_i^{\text{obs}} | \theta_{\text{Time}}\right)}{F_{\text{Time}}\left(t_i^{\text{end}} | \theta_{\text{Time}}\right) - F_{\text{Time}}\left(t_i^{\text{start}} | \theta_{\text{Time}}\right)}. \end{aligned} \tag{A3.2}$$

Here, t_i^{obs} is the time when the i-th agent has been observed, and θ_{Time} is the parameter vector for the temporal pattern.

Since both f_{Time} and F_{Time} are usually easy to evaluate and the computational complexity is independent of the system size and linear in the number of surveyed agents, no sophisticated algorithms are required to evaluate and maximize the likelihood.

3.D.3 Fitting the route choice model

In this section, we provide instructions on how to fit the route choice model. We start by discussing conceptual details before we show how to evaluate the likelihood function efficiently by computing and reusing partial results.

We maximize the likelihood of the route choice model in a repeated two step procedure: first, we compute the set of admissible routes that most agents choose from. Then we fit the submodel that assigns the admissible routes with probabilities. We repeat these steps with different route admissibility parameters until a model is identified that maximizes the likelihood approximately.

The need for the two step procedure comes from our distinction between admissible and inadmissible routes. Whether or not a route is classified as admissible is a yes/no question. Therefore, the likelihood function is not continuous in the parameters that define admissibility. As a consequence, classic gradient descent methods cannot be applied to find the best fitting parameters to define route admissibility. In fact, computing admissible routes is so computationally costly that an exhaustive search for the optimal route admissibility parameters is often impracticable.

Below, we will focus on the second step of the two step procedure outlined above. We will not provide details of how to compute admissible paths, as this is beyond the scope of this paper. Instead, we refer the interested reader to chapter 2 of this thesis. Throughout this appendix, we will assume that a suitable set of admissible routes has already been computed and focus on fitting the submodel that assigns probabilities to routes.

Recall that our survey effort is not uniformly distributed over all potential routes in general. Therefore, we have to consider where, when, and for how long we conducted surveys on the survey date. To measure the effect of survey timing, the temporal pattern model must be fitted before the route choice model. Similar to the survey timing, the compliance rate affects the route choice model, too, as we will see below. Therefore, the compliance model must be fitted before the route choice model as well.

If an agent appears in our data set, they must have been surveyed somewhere on the survey date. Let k_a^{obs} be the location where we observed agent a. With the compliance model, the temporal pattern model, the route choice model, and parameters θ_{Route} to be fitted, we can determine the probability $p_a^{\text{obs}}(\theta_{\text{Route}})$ to observe agent a in the survey shift conducted at location k_a^{obs} on the observation day. Furthermore, we can compute the probability $p_a^{\text{all}}(\theta_{\text{Route}})$ to observe agent a at *some* survey location operated on that day. This probability reflects the survey effort on the day of the observation and takes the lengths of the survey shifts into account. The quotient $p_a^{\text{obs}}/p_a^{\text{all}}$ is the probability that we observed the agent at location k_a^{obs} given that the agent was surveyed at some survey location operated that day. Consequently, the conditional likelihood function reads

$$\begin{aligned} L_{\text{Route}}(\theta_{\text{Route}}) &= \prod_a \mathbb{P}\big(\text{survey agent } a \text{ at location } k_a^{\text{obs}} | \theta_{\text{Route}}, \text{ agent } a \text{ surveyed on day } d_a\big) \\ &= \prod_a \frac{p_a^{\text{obs}}(\theta_{\text{Route}})}{p_a^{\text{all}}(\theta_{\text{Route}})}. \end{aligned} \tag{A3.3}$$

To compute p_a^{obs} and p_a^{all}, we have to recall the structure of our route choice model. If agent a is travelling from origin i to destination j, then the probability that agent a passes

survey location k on their journey is

$$\rho_{ijk} \quad = \quad (1-\eta_{\mathrm{c}}) \sum_{P \in \mathcal{P}_{ij}: k \in P} \frac{l_P^{-\lambda}}{\sum\limits_{\tilde{P} \in \mathcal{P}_{ij}} l_{\tilde{P}}^{-\lambda}} + \eta_{\mathrm{c}}\eta_{\mathrm{o}}. \tag{A3.4}$$

Here, $\mathcal{P}_{ij}$ is the set of admissible routes for the source-sink pair (i, j), l_P is the length of path P, and η_{c}, η_{o}, and λ are the parameters to be fitted. Recall that η_{c} is the probability that an agent chooses an inadmissible path, and η_{o} is the probability that agents driving on inadmissible paths are driving via any given location in the road network.

In subsection 3.D.3.1, we will provide details of how equation (A3.4) is related to p_a^{obs} and p_a^{all}. Furthermore, in subsection 3.D.3.2, we will show how the likelihood function can be computed efficiently. Beforehand, however, we have to discuss issues that could result in non-informative models.

Avoiding dominant noise

Equation (A3.4) includes a noise term accounting for agents not driving on admissible routes. We assume that these agents choose the locations they pass randomly. If all traffic were random ($\eta_{\mathrm{c}} = 1$) and all randomly driving agents were driving by all survey locations ($\eta_{\mathrm{o}} = 1$), the probabilities to observe these agents would be maximized. However, with $\eta_{\mathrm{c}} = \eta_{\mathrm{o}} = 1$, our route choice model would be non-informative and misleading.

To avoid that maximizing the likelihood results in a non-informative model, we need to integrate additional information. We therefore assume that agents driving on an inadmissible route have not been surveyed more than once once. This makes models unfit in which agents drive on zig-zag routes via many survey locations.

We apply this assumption to our survey data only and not for potential model predictions. That is, the additional assumption does not affect our model but how we view our data set. Since survey locations are often far apart from each other, the additional assumption is typically true in practice. However, if we surveyed traffic at locations close to each other,

the additional assumption could lead to wrong results. Nonetheless, tests with simulated observation data suggest that the error introduced by this additional assumption is small as long as only few agents travel on inadmissible routes.

Non-estimability of noise

Even if noise does not dominate the model, our noise model leads to identifiability issues. Our route choice model allows us to determine the correct ratio between traffic along admissible paths and the random traffic *observed* at survey locations. However, our data contain no information on how many agents are driving on inadmissible routes *without passing* any survey location. Therefore, the total share of agents driving on inadmissible routes remains unknown. Consequently, we can neither estimate the probability η_c that agents choose an inadmissible path nor the probability η_o that these agents are observed at a survey location.

This issue can be resolved by assuming that *most* of the traffic (e.g. 95%) follows admissible paths. We can obtain specific estimates of η_c and η_o only if we know the total daily number of driving agents for some donor-recipient pairs. This number, however, is usually unknown in large-scale systems.

We argue that it is reasonable to assume that most agents drive on admissible routes, unless models with a larger noise term fit the data significantly better. We fit our model constraining $\eta_c \leq 0.05$.

3.D.3.1 Deriving the likelihood function of the route choice model

After providing an overview of the model fitting procedure and potential issues, we proceed to derive the concrete structure of the likelihood function to be maximized.

Consider an agent a travelling from i to j via survey location k on day d. Let ξ be the compliance rate, and let τ_{kd} be the probability that this agent passes the survey location while it is operated. The value of τ_{kd} depends on the length and the starting time of the survey shift conducted at location k on day d. As route choice and travel timing are assumed

to be independent random choices, the probability to survey agent a at k on day d is given by $p^{\text{obs}}_{ijkd} = \rho_{ijk}\tau_{kd}\xi$. Recall that ρ_{ijk} is the probability that agent a drives via location k.

As discussed in the previous section, we make an additional assumption on agents travelling on inadmissible routes. Therefore, we cannot apply equation (A3.4) to determine ρ_{ijk} when we fit the model. To derive an expression for p^{obs}_{ijkd} based on the adjusted ρ_{ijk}, we start by considering agents travelling on inadmissible routes. As proposed above, we assume that such agents in our data set were not observed at more than one survey location. Hence, the probability that we observed such an agent on day d at location k (and not at any other operated survey location) is

$$\tilde{\eta}^{\text{o}}_{kd} \;=\; \xi\tau_{kd}\eta_{\text{o}} \prod_{\bar{k}\in L_d:\bar{k}\neq k} \left(1-\xi\tau_{\bar{k}d}\eta_{\text{o}}\right), \tag{A3.5}$$

whereby L_d is the set of all survey locations operated on day d. Here,

$$\xi\tau_{kd}\eta_{\text{o}} \;=\; \mathbb{P}(\text{survey } \tilde{a} \text{ at location } k) \tag{A3.6}$$

$$1-\xi\tau_{\bar{k}d}\eta_{\text{o}} \;=\; \mathbb{P}\big(\text{do not survey } \tilde{a} \text{ at location } \bar{k}\big) \tag{A3.7}$$

for any agent $\tilde{a}$ driving on an inadmissible route.

Now let us consider an agent a travelling from i to j on day d on an admissible *or* an inadmissible route. Recall that agents choose inadmissible routes with probability η_{c}. Consequently, the probability that agent a chooses an admissible route via location k is

$$(1-\eta_{\text{c}}) \sum_{P\in\mathcal{P}_{ij}:k\in P} \frac{l_P^{-\lambda}}{\sum\limits_{\tilde{P}\in\mathcal{P}_{ij}} l_{\tilde{P}}^{-\lambda}}$$

and the probability that we surveyed a at k on day d is given by

$$p^{\text{obs}}_{ijkd} \;=\; \xi\tau_{kd}\,(1-\eta_{\text{c}}) \sum_{P\in\mathcal{P}_{ij}:k\in P} \frac{l_P^{-\lambda}}{\sum\limits_{\tilde{P}\in\mathcal{P}_{ij}} l_{\tilde{P}}^{-\lambda}} + \eta_{\text{c}}\tilde{\eta}^{\text{o}}_{kd}. \tag{A3.8}$$

After computing p^{obs}_{ijkd}, we must determine the probability p^{all}_{ijd} to observe an agent travelling from i to j on day d at *some* location. Note that the distribution of travelling agents is independent of the day according to our model. The only reason why p^{all}_{ijd} depends on d is that surveys are conducted at different locations and times on different days.

We can split p^{all}_{ijd} into

$$p^{\text{all}}_{ijd} \;=\; (1-\eta_{\text{c}})\, p^{\text{adm}}_{ijd} + \eta_{\text{c}} p^{\text{inadm}}_{d}, \tag{A3.9}$$

whereby p^{adm}_{ijd} is the probability to observe an agent driving on an admissible route from i to j on day d, and p^{inadm}_{d} the respective probability for an agent driving along an inadmissible route. The value of p^{inadm}_{d} is independent of origin and destination of the considered agent.

We find p^{adm}_{ijd} by summing over all admissible paths $P \in \mathcal{P}_{ij}$ from i to j that go via a survey location $\tilde{k} \in L_d$ that is operated on day d. The probability to choose an admissible path $P \in \mathcal{P}_{ij}$ is given by

$$\mathbb{P}(\text{choose path } P \,|\, \text{driving on an admisible path}) \;=\; \frac{l_P^{-\lambda}}{\sum\limits_{\tilde{P}\in\mathcal{P}_{ij}} l_{\tilde{P}}^{-\lambda}}. \tag{A3.10}$$

The probability to observe an agent driving along this path at at least one operated survey location is

$$\mathbb{P}(\text{survey agent} \mid \text{driving on path } P \text{ on day } d) \;=\; 1 - \prod_{\bar{k}\in L_d : \bar{k}\in P} (1-\xi\tau_{\bar{k}d})\,. \tag{A3.11}$$

Consequently,

$$p^{\text{adm}}_{ijd} \;=\; \sum_{P\in\mathcal{P}_{ij}:\tilde{k}\in P, \tilde{k}\in L_d} \frac{l_P^{-\lambda}}{\sum\limits_{\tilde{P}\in\mathcal{P}_{ij}} l_{\tilde{P}}^{-\lambda}} \left(1 - \prod_{\bar{k}\in L_d:\bar{k}\in P} (1-\xi\tau_{\bar{k}d})\right). \tag{A3.12}$$

After finding p_{ijd}^{adm}, we need to find an expression for p_d^{inadm}. This is the probability to observe an agent driving along an inadmissible path at exactly one of the survey locations operated on day d (compare with equation (A3.5)):

$$p_d^{\text{inadm}} = \eta_{\text{o}} \sum_{\bar{k} \in L_d} \xi \tau_{\bar{k}d} \prod_{\hat{k} \in L_d : \hat{k} \neq \bar{k}} \left(1 - \xi \tau_{\hat{k}d} \eta_{\text{o}}\right). \tag{A3.13}$$

Putting these pieces together we obtain the log-likelihood function

$$L(\theta) = \prod_{(ijkd)} \frac{p_{ijkd}^{\text{obs}}(\theta)}{p_{ijd}^{\text{all}}(\theta)} \tag{A3.14}$$

with p_{ijkd}^{obs} as defined in equation (A3.8) and p_{ijd}^{all} as defined in equation (A3.9) with the terms given in equations (A3.12) and (A3.13). Our goal is to find $\hat{\theta} = \left(\hat{\lambda}, \hat{\eta}_c, \hat{\eta}_o\right)$ that maximizes $L(\theta)$.

3.D.3.2 Computing the likelihood of the route choice model

During the likelihood maximization, we have to evaluate $L(\theta)$ and its derivatives many times. Computing $L(\theta)$ as derived above is expensive, because we have to compute nested products and sums. In this section, we show how the function can be split to speed up computations.

When we maximize the likelihood $L(\theta)$, we consider the log-likelihood $\ln L(\theta)$ to avoid numerical instabilities. However, we will work with the original likelihood function here for notational convenience.

To evaluate the likelihood function faster, we compute the following expressions first:

$$\begin{aligned}
\Xi_{dP} &= 1 - \prod_{\bar{k} \in L_d : \bar{k} \in P} \left(1 - \xi \tau_{\bar{k}d}\right), & \Lambda_{ij}^{\text{norm}}(\lambda) &= \frac{1}{\sum_{\tilde{P} \in \mathcal{P}_{ij}} l_{\tilde{P}}^{-\lambda}}, \\
\Phi_d(\eta_\text{o}) &= \prod_{\bar{k} \in L_d} \left(1 - \xi \tau_{\bar{k}d} \eta_\text{o}\right), & \Lambda_{ijk}^{\text{spec}}(\lambda) &= \sum_{P \in \mathcal{P}_{ij} : k \in P} l_P^{-\lambda}, \\
\Upsilon_d(\eta_\text{o}) &= \sum_{\bar{k} \in L_d} \frac{\xi \tau_{\bar{k}d}}{1 - \eta_\text{o} \xi \tau_{\bar{k}d}}, & \Lambda_{ijd}^{\text{all}}(\lambda) &= \sum_{P \in \mathcal{P}_{ij} : \tilde{k} \in P, \tilde{k} \in L_d} l_P^{-\lambda} \Xi_{dP}.
\end{aligned} \tag{A3.15}$$

With these expressions, we can write

$$L(\lambda, \eta_\text{c}, \eta_\text{o}) = \prod_{(ijkd)} \frac{\xi \tau_{kd} \left(\left(1 - \eta_\text{c}\right) \Lambda_{ij}^{\text{norm}}(\lambda) \Lambda_{ijk}^{\text{spec}}(\lambda) + \frac{\eta_\text{c} \eta_\text{o}}{1 - \eta_\text{o} \xi \tau_{kd}} \Phi_d(\eta_\text{o}) \right)}{\left(1 - \eta_\text{c}\right) \Lambda_{ij}^{\text{norm}}(\lambda) \Lambda_{ijd}^{\text{all}}(\lambda) + \eta_\text{c} \eta_\text{o} \Upsilon_d(\eta_\text{o}) \Phi_d(\eta_\text{o})}. \tag{A3.16}$$

Computing all required values of Ξ_{dP}, $\Phi_d(\eta_\text{o})$, $\Upsilon_d(\eta_\text{o})$, $\Lambda_{ij}^{\text{norm}}(\lambda)$, $\Lambda_{ijk}^{\text{spec}}(\lambda)$, and $\Lambda_{ijd}^{\text{all}}(\lambda)$ before the likelihood increases the computational efficiency.

Runtime analysis

To demonstrate how the function split speeds up the likelihood computation, we will now conduct a runtime analysis. To this end, we define count variables as follows:

n_{obs}: total number of surveyed agents

n_{pairs}: number of origin-destination pairs for which we have surveyed at least one agent

n_{days}: number of survey days

n_{loc}: number of survey locations

$n_{\text{pairs/day}}$: average daily number of origin-destination pairs for which we have surveyed at least one agent

$n_{\text{loc/day}}$: average number of survey locations operated on a survey day

$n_{\text{paths/pair}}$: average number of admissible routes between an origin and a destination

Let us start the runtime analysis by noting that Ξ_{dP} is independent of all parameters that we are optimizing. Therefore, we can compute Ξ_{dP} for all indices d and P before the optimization. For each survey day, we have to compute Ξ_{dP} for all admissible paths connecting origin-destination pairs for which we have observed an agent. Computing a single value of Ξ_{dP} requires $\mathcal{O}\big(n_{\text{loc/day}}\big)$ operations. Hence, we can compute Ξ_{dP} in $\mathcal{O}\big(n_{\text{days}} n_{\text{pairs/day}} n_{\text{paths/pair}} n_{\text{loc/day}}\big)$. Later, we can access the pre-computed values in effectively constant time.

To determine the values of Φ_d and Υ_d, we compute a product or sum over all survey locations operated on each day, respectively. This are $\mathcal{O}\big(n_{\text{days}} n_{\text{loc/day}}\big)$ operations.

The normalization constants $\Lambda_{ij}^{\text{norm}}$ for route choice probabilities must be computed for each origin-destination pair for which we have surveyed at least one agent. Computing a single $\Lambda_{ij}^{\text{norm}}$ value requires us to sum over all paths from the considered origin to the respective destination. Hence, we require $\mathcal{O}\big(n_{\text{pairs}} n_{\text{paths/pair}}\big)$ operations in total. The same applies to the computation of $\Lambda_{ijk}^{\text{spec}}$ with the exception that we have to consider a different set of paths for each observed combination of origin, destination, and survey location. Hence, computing all the $\Lambda_{ijk}^{\text{spec}}$ values requires $\mathcal{O}\big(n_{\text{pairs}} n_{\text{paths/pair}} n_{\text{loc}}\big)$ operations.

To compute the values of $\Lambda_{ijd}^{\text{all}}$, we conduct operations similar to those for $\Lambda_{ijk}^{\text{spec}}$. However, each survey day we may consider a different set of survey locations. Therefore, we need $\mathcal{O}\big(n_{\text{days}} n_{\text{pairs/day}} n_{\text{paths/pair}} n_{\text{loc/day}}\big)$ operations.

With all partial results determined, we can compute the likelihood in $\mathcal{O}(n_{\text{obs}})$ operations. We arrive at a final runtime of $\mathcal{O}\big(n_{\text{pairs}} n_{\text{paths/pair}} n_{\text{loc}} + n_{\text{days}} n_{\text{pairs/day}} n_{\text{paths/pair}} n_{\text{loc/day}} + n_{\text{obs}}\big)$, whereby the second summand is usually dominating. Note that $\mathcal{O}\big(n_{\text{days}} n_{\text{pairs/day}}\big)$ is bounded by $\mathcal{O}(n_{\text{obs}})$. Furthermore, $n_{\text{paths/pair}}$ and $n_{\text{loc/day}}$ are usually moderate numbers that are independent of the scale of the considered system and do not increase with the sample size. Thus, it is appropriate to classify the runtime of our algorithm as $\mathcal{O}(n_{\text{obs}})$, which is the sample size.

3.D.4 Likelihood of the stochastic gravity model

In this section, we first state the optimization problem that must be solved to fit the gravity model to survey data. Then, we describe why this problem is computationally hard. In the second part of this section, we show how the structure of the likelihood function and the excess of observed zero counts can be exploited to compute the log-likelihood more efficiently.

3.D.4.1 Deriving the likelihood function of the stochastic gravity model

We parameterize the negative binomial distribution for a random variable N by

$$\mathbb{P}(N = n) \;=\; f_{\mathrm{NB}}(n \,|\, \mu, p) \;=\; \binom{n + r(\mu, p) - 1}{n} p^{r(\mu,p)} (1-p)^n \tag{A3.17}$$

with $r(\mu, p) := \frac{p}{1-p}\mu$. Here, $\mu = \mathbb{E}(N)$ is the mean of the random variable N, and $p = \frac{\mu}{\sigma^2}$ is the quotient of mean and variance of N. For convenience, we write $f_{\mathrm{NB}}(n \,|\, r, p)$ below. The fitted value of μ can be obtained using the equation $\hat{\mu} = \frac{1-\hat{p}}{\hat{p}}\hat{r}$.

Let Ψ be the set of all considered origin-destination pairs, denoted by $(i, j) \in \Psi$. We assume that on each day, the number of travelling agents for each origin-destination pair $(i, j) \in \Psi$ is negative binomially distributed with parameters r_{ij} and p. The parameter r_{ij} depends on the origin-destination pair (i, j), because we estimate the mean number of travelling agents with a gravity model that depends on the properties of origins and destinations. The parameter p, however, is assumed to be similar for all origin-destination pairs.

Let us index survey shifts with $s \in S$, whereby S is the set of all survey shifts. Each survey shift $s \in S$ is conducted at a location k_s and in a time interval $t_s = \left[t_s^{\mathrm{start}}, t_s^{\mathrm{end}}\right]$. Let ρ_{ijk_s} be the probability that an agent travelling from origin i to destination j chooses a path via location k_s. Furthermore, let τ_s be the probability that an agent travelling via k_s passes the survey location k_s during the time interval t_s the survey was conducted, and let ξ be the compliance rate. Lastly, let n_{ijs} be the number of agents travelling from i to j who were surveyed in shift s.

In our hierarchical stochastic model, the number N_{ijs} of agents travelling between the origin-destination pair $(i,j) \in \Psi$ and observed in shift $s \in S$ is distributed as

$$N_{ijs} \sim Binomial\left(Binomial\left(Binomial\left(NegativeBinomial\left(r_{ij}, q\right), \rho_{ijk_s}\right), \tau_s\right), \xi\right). \quad \text{(A3.18)}$$

Define $\tilde{p}_{ijs} := \frac{p}{p+\rho_{ijk_s}\tau_s\xi(1-p)}$. It can be shown (Villa and Escobar, 2006) that

$$\mathbb{P}(N_{ijs} = n_{ijs}) \quad = \quad f_{\text{NB}}\left(n_{ijs} \,|\, r_{ij}, \tilde{p}_{ijs}\right). \quad \text{(A3.19)}$$

We desire to maximize the likelihood

$$L\left(\theta\right) \quad = \quad \prod_{(i,j,s)\in\Psi\times S} f_{\text{NB}}\left(n_{ijs} \,|\, r_{ij}\left(\theta\right), \tilde{p}_{ijs}\left(\theta\right)\right), \quad \text{(A3.20)}$$

whereby θ is a vector of parameters.

3.D.4.2 Computing the likelihood of the stochastic gravity model

To compute the likelihood given in equation (A3.20), we have to consider $|\Psi|\,|S|$ combinations of origin-destination pairs and survey shifts. This is a very large number in general. For example, in the application section of this paper, we considered about $|\Psi| \approx 3.7 \cdot 10^5$ origin-destination pairs and $|S| \approx 1600$ survey shifts. Though computing $f_{\text{NB}}\left(n_{ijs} \,|\, r_{ij}\left(\theta\right), \tilde{p}_{ijs}\left(\theta\right)\right)$ for all combinations of i, j, and s might be feasible, it takes too much time for a multidimensional optimization. To maximize the likelihood with reasonable effort, we would need to evaluate L within fractions of a second. Below, we present a way to speed up the likelihood computation.

Given agent counts n_{ijs}, the log-likelihood function reads

$$\ell\left(\theta\right) \quad = \quad \sum_{(i,j,s)\in\Psi\times S} \ln f_{\text{NB}}\left(n_{ijs} \,|\, r_{ij}\left(\theta\right), \tilde{p}_{ijs}\left(\theta\right)\right). \quad \text{(A3.21)}$$

The probability mass function $f_{\mathrm{NB}}\left(n_{ijs} \,|\, r_{ij}\left(\theta\right), \tilde{p}_{ijs}\left(\theta\right)\right)$ is particularly simple to compute in two cases: (1) if there is no admissible path from i to j via the survey location k_s, and (2) if $n_{ijs} = 0$. Typically, most of the observations fall in one of these categories. We exploit that in the following way:

1. We assume that *all* observations satisfy the criterion (1) or (2), respectively, and compute the log-likelihood under this assumption.

2. We consider all the data for which the assumption above was incorrect and compute the actual likelihood for these count values.

3. We determine the portion of the likelihood that we computed in step 1 under a wrong assumption. Then we replace this part of the likelihood with the correct likelihood computed in step 2.

Before we provide further details, we introduce some helpful notation.

3.D.4.2.1 Some notes on notation

Let $\Omega = \Psi \times S$ be the set of the indices of all observations. For convenience, we label the following logical statements as given below:

- Statement "0": $n_{ijs} = 0$
- Statement "e": $\exists P \in \mathcal{P}_{ij} : k_s \in P$.

Recall that $\mathcal{P}_{ij}$ is the set of all admissible paths from i to j.

To denote that all elements in an index set satisfy a certain logical statement, we attach a corresponding subscript to the set. For example, $\Omega_0 \subseteq \Omega$ is the subset of Ω for which all elements satisfy statement "0":

$$\Omega_0 \quad = \quad \left\{(i,j,s) \in \Omega \,|\, n_{ijs} = 0\right\}. \tag{A3.22}$$

That is, Ω_0 contains the indices of zero counts. Similarly,

$$\Omega_e \;=\; \{(i,j,s) \in \Omega \,|\, \exists P \in \mathcal{P}_{ij} : k_s \in P\} \tag{A3.23}$$

contains the indices of all counts of agents surveyed at one of their admissible routes. That is, observations in Ω_e do not have to be considered traffic noise. Instead, these agents were observed where we expected them. Hence, we labelled the corresponding logical statement "e" for "expected".

We can use the same subscript notation to denote intersections, unions, and complements of sets. Recall that for any logical statements A and B, "$\neg A$" means "*not* A", $A \vee B$ means "A *or* B", and "$A \wedge B$" means "A *and* B". Thus, for example, $\Omega_{\neg 0} = \Omega \backslash \Omega_0$, $\Omega_{\neg e} = \Omega \backslash \Omega_e$, $\Omega_{0 \wedge e} = \Omega_0 \cap \Omega_e$, and $\Omega_{0 \vee e} = \Omega_0 \cup \Omega_e$.

Below, we are going to compute the log-likelihood ℓ under specific assumptions about our data. To show which data we are considering, respectively, we use a *subscript*. For example,

$$\ell_{\Omega_e}(\theta) \;=\; \sum_{(i,j,s) \in \Omega_e} \ln f_{\mathrm{NB}}\left(n_{ijs} \,|\, r_{ij}(\theta), \tilde{p}_{ijs}(\theta)\right) \tag{A3.24}$$

denotes the log-likelihood of data with indices in Ω_e.

Furthermore, we use a *superscript* to denote under which assumption we compute the log-likelihood. For example, if we compute the log-likelihood of all data under the assumption that we only observed zeros, we write

$$\ell^0_{\Omega}(\theta) \;=\; \sum_{(i,j,s) \in \Omega} \ln f_{\mathrm{NB}}\left(0 \,|\, r_{ij}(\theta), \tilde{p}_{ijs}(\theta)\right). \tag{A3.25}$$

Note that we iterated over *all* data here. That is, we included non-zero counts and assumed (falsely) that they *were* zero.

3.D.4.2.2 Splitting the log-likelihood

After introducing the required notation, we now proceed explaining how the log-likelihood can be computed more efficiently. Observe that for any logical statement A,

$$\ell_{\Omega}(\theta) = \ell_{\Omega}^{A}(\theta) - \ell_{\Omega_{\neg A}}^{A}(\theta) + \ell_{\Omega_{\neg A}}(\theta). \tag{A3.26}$$

That is, if we compute the log-likelihood under some assumption A, subtract the portion of this quantity for which the assumption was wrong, and add the correct log-likelihood value for these data instead, then we obtain the correct log-likelihood value.

Applying this observation and basic set operations, we obtain

$$\ell_{\Omega}(\theta) = \ell_{\Omega}^{0\wedge\neg e}(\theta) - \ell_{\Omega_{\neg 0\vee e}}^{0\wedge\neg e}(\theta) + \ell_{\Omega_{\neg 0\vee e}}(\theta) \tag{A3.27}$$

$$\ell_{\Omega_{\neg 0\vee e}}^{0\wedge\neg e}(\theta) = \ell_{\Omega_e}^{0\wedge\neg e}(\theta) + \ell_{\Omega_{\neg 0\wedge\neg e}}^{0\wedge\neg e}(\theta) \tag{A3.28}$$

$$\ell_{\Omega_{\neg 0\vee e}}(\theta) = \ell_{\Omega_e}(\theta) + \ell_{\Omega_{\neg 0\wedge\neg e}}(\theta) \tag{A3.29}$$

$$\ell_{\Omega_e}(\theta) = \ell_{\Omega_e}^{0}(\theta) - \ell_{\Omega_{\neg 0\wedge e}}^{0}(\theta) + \ell_{\Omega_{\neg 0\wedge e}}(\theta) \tag{A3.30}$$

Inserting these equations into each other yields

$$\ell_{\Omega}(\theta) = \ell_{\Omega}^{0\wedge\neg e}(\theta) - \ell_{\Omega_e}^{0\wedge\neg e}(\theta) - \ell_{\Omega_{\neg 0\wedge\neg e}}^{0\wedge\neg e}(\theta) + \ell_{\Omega_e}^{0}(\theta) - \ell_{\Omega_{\neg 0\wedge e}}^{0}(\theta) + \ell_{\Omega_{\neg 0\wedge e}}(\theta) + \ell_{\Omega_{\neg 0\wedge\neg e}}(\theta). \tag{A3.31}$$

The likelihood components on the right hand side of equation (A3.31) are easy to compute, because they have either a simple functional form or consider only a small fraction of our data. Most of our observations are in Ω_0 and/or Ω_e.

3.D.4.2.3 Computing the log-likelihood

To compute the log-likelihood we determine all the individual components of equation (A3.31) and insert them into the equation. Below, we describe how to compute each of the components efficiently.

$\ell_{\Omega}^{0\wedge\neg e}(\theta)$: If none of our survey locations were on any admissible route (statement "$\neg e$"), then the probability that an agent travels from i to j via a survey location k_s is

$$\rho_{ijk_s} \quad = \quad \eta_c \eta_o, \tag{A3.32}$$

which is independent of origin, destination, and survey location. It follows that $\tilde{p}_{ijs} = \frac{p}{p+\rho_{ijk}\tau_s(1-p)} = \frac{p}{p+\eta_c\eta_o\tau_s(1-p)} = \tilde{p}_s$ is independent of the considered source-sink pair (i,j). If we furthermore assume that no agent has been observed (statement "0"), then the likelihood function is given by

$$L_{\Omega}^{0\wedge\neg e}(\theta) \quad = \quad \prod_{s\in S}\prod_{(i,j)\in\Psi}\binom{0+r_{ij}(\theta)-1}{0}\left(\tilde{p}_s(\theta)\right)^{r_{ij}}\left(1-\tilde{p}_s(\theta)\right)^{0}, \tag{A3.33}$$

and the log-likelihood is

$$\ell_{\Omega}^{0\wedge\neg e}(\theta) \quad = \quad \left(\sum_{s\in S}\ln\left(\tilde{p}_s(\theta)\right)\right)\left(\sum_{(i,j)\in\Psi} r_{ij}(\theta)\right). \tag{A3.34}$$

We can compute this value in $\mathcal{O}(|S|+|\Psi|)$ steps.

$\ell_{\Omega_e}^{0\wedge\neg e}(\theta)$: Let $\Psi_k = \{(i,j)\in\Psi \,|\, \exists P\in\mathcal{P}_{ij} : k\in P\}$ be the set of origin-destination pairs for which at least one admissible path $P\in\mathcal{P}_{ij}$ passes survey location k. Let furthermore $\tilde{r}_k = \sum_{ij\in\Psi_k} r_{ij}$ be the sum of the r-parameters for these pairs. Then it

is easy to compute

$$\ell_{\Omega_e}^{0\wedge\neg e}(\theta) \;=\; \sum_{s\in S} \tilde{r}_{k_s} \ln\left(\tilde{p}_s(\theta)\right). \tag{A3.35}$$

Computing the values of $\tilde{r}_k$ for all used survey sites before evaluating equation (A3.35) saves the efforts of computing the same quantity multiple times. The worst-case runtime for computing the values of r_k is $\mathcal{O}(|L|\,|\Psi|)$, whereby L is the set of all survey locations. Computing $\ell_{\Omega_e}^{0\wedge\neg e}(\theta)$ runs in $\mathcal{O}(|S| + |L|\,|\Psi|)$.

$\ell_{\Omega_{\neg 0\wedge\neg e}}^{0\wedge\neg e}(\theta)$: The set $\Omega_{\neg 0\wedge\neg e}$ contains the indices of those observations where the agents were certainly driving along inadmissible routes. As most agents drive along admissible routes, the set $\Omega_{\neg 0\wedge\neg e}$ is small. Hence, we do not need any optimizations to compute the value of $\ell_{\Omega_{\neg 0\wedge\neg e}}^{0\wedge\neg e}(\theta)$:

$$\ell_{\Omega_{\neg 0\wedge\neg e}}^{0\wedge\neg e}(\theta) \;=\; \sum_{(i,j,s)\in\Omega_{\neg 0\wedge\neg e}} r_{ij} \ln\left(\tilde{p}_{ijs}(\theta)\right). \tag{A3.36}$$

This runs in $\mathcal{O}(|\Omega_{\neg 0\wedge\neg e}|)$.

$\ell_{\Omega_e}^{0}(\theta)$: Computing the value of $\ell_{\Omega_e}^{0}(\theta)$ may be the most challenging part of the likelihood computation, as the set Ω_e is large and the likelihood function is not simple. Therefore, we apply Taylor approximations in $\nu_s := \xi\tau_s$ to split the nested sums into separate sums that can be computed more efficiently. The approximation point of the Taylor expansion will be the mean $\bar{\nu} := \frac{\xi}{|S|}\sum_{s\in S}\tau_s$. We get

$$
\begin{aligned}
\ell^0_{\Omega_e}(\theta) &= \sum_{s\in S}\sum_{(i,j)\in\Psi_{k_s}} r_{ij}\ln\left(\tilde{p}_{ijs}\right) \\
&= \sum_{s\in S}\sum_{(i,j)\in\Psi_{k_s}} r_{ij}\ln\left(\frac{p}{p+(1-p)\,\rho_{ijk_s}\nu_s}\right) \\
\overset{\text{Taylor expansion}}{\approx} &\sum_{s\in S}\sum_{(i,j)\in\Psi_{k_s}} r_{ij}\left(\ln\left(\frac{p}{p+(1-p)\,\rho_{ijk_s}\bar{\nu}}\right)+\sum_{m=1}^{M}\frac{1}{m}\left(\frac{-(1-p)\,\rho_{ijk_s}\left(\nu_s-\bar{\nu}\right)}{p+(1-p)\,\rho_{ijk_s}\bar{\nu}}\right)^m\right) \\
&= \sum_{s\in S}\sum_{(i,j)\in\Psi_{k_s}} r_{ij}\ln\left(\frac{p}{p+(1-p)\,\rho_{ijk_s}\bar{\nu}}\right) \\
&\qquad +\sum_{s\in S}\sum_{m=1}^{M}\frac{1}{m}\left(\nu_s-\bar{\nu}\right)^m\sum_{(i,j)\in\Psi_{k_s}} r_{ij}\left(\frac{-(1-p)\,\rho_{ijk_s}}{p+(1-p)\,\rho_{ijk_s}\bar{\nu}}\right)^k \\
&= \sum_{s\in S}R_{k_s}+\sum_{s\in S}\sum_{m=1}^{M}\frac{1}{m}\left(\nu_s-\bar{\nu}\right)^m\tilde{R}_{k_s m} \qquad \text{(A3.37)}
\end{aligned}
$$

with

$$
R_k = \sum_{(i,j)\in\Psi_k} r_{ij}\ln\left(\frac{p}{p+(1-p)\,\rho_{ijk}\bar{\nu}}\right), \qquad \text{(A3.38)}
$$

$$
\tilde{R}_{km} = \sum_{(i,j)\in\Psi_k} r_{ij}\left(\frac{-(1-p)\,\rho_{ijk}}{p+(1-p)\,\rho_{ijk}\bar{\nu}}\right)^m. \qquad \text{(A3.39)}
$$

Note that p and r_{ij}, and therefore also $\tilde{p}_{ij}$, R_k, and $\tilde{R}_{km}$ depend on θ. The parameter M determines the precision of the Taylor approximation.

We can estimate the error introduced by the Taylor approximation by considering the term $\frac{-(1-p)\rho_{ijk_s}(\nu_s-\bar{\nu})}{p+(1-p)\rho_{ijk_s}\bar{\nu}}$. Recall that ν_s, ρ_{ijk_s}, and p can be interpreted as probabilities and are therefore bounded between 0 and 1. Consequently, choosing $\bar{\nu}=\frac{1}{2}$ would imply $|\nu_s-\bar{\nu}|\leq\frac{1}{2}$. In this case,

$$
\left|\frac{-(1-p)\,\rho_{ijk_s}\left(\nu_s-\bar{\nu}\right)}{p+(1-p)\,\rho_{ijk_s}\bar{\nu}}\right| \leq \frac{(1-p)\,\rho_{ijk_s}\bar{\nu}}{p+(1-p)\,\rho_{ijk_s}\bar{\nu}}, \qquad \text{(A3.40)}
$$

which is a function decreasing in p and increasing in ρ_{ijk_s}. As $p=\frac{\mu}{\sigma^2}>0$, we know that the full term is less than 1, which in turn guarantees that the series

converges. Moreover, it is reasonable to assume that the overdispersion is not extreme and $p = \frac{\mu}{\sigma^2} \geq \frac{1}{10}$. This would imply that

$$\left| \frac{-(1-p)\,\rho_{ijk_s}\,(\nu_s - \bar{\nu})}{p + (1-p)\,\rho_{ijk_s}\bar{\nu}} \right| \;\leq\; \frac{9}{11}, \tag{A3.41}$$

and the error would be bounded by a quantity proportional to $\frac{1}{M+1}\left(\frac{9}{11}\right)^{M+1}$. In practice, the error can be checked by investigating the change in the computed log-likelihood as M is increased. In our application, the error was small for $M = 3$.

To see how the Taylor expansion simplifies the computation, note that both R_k and $\tilde{R}_{km}$ do not have to be computed for each survey shift $s \in S$ but rather for each used survey location $k \in L$. Therefore, computing these values runs in $\mathcal{O}(M\,|L|\,|\Psi|)$. Evaluating the right hand side of equation (A3.37) runs in $\mathcal{O}(M\,|S|)$. Thus, the Taylor expansion allows us to compute $\ell^0_{\Omega_e}(\theta)$ in $\mathcal{O}(M\,|L|\,|\Psi| + M\,|S|)$ instead of $\mathcal{O}(|\Psi|\,|S|)$.

$\ell^0_{\Omega_{\neg 0 \wedge e}}(\theta)$: As the number of non-zero counts is moderate, so is $|\Omega_{\neg 0 \wedge e}|$. Therefore, we could compute $\ell^0_{\Omega_{\neg 0 \wedge e}}(\theta)$ without further optimizations. However, we compute $\ell^0_{\Omega_{\neg 0 \wedge e}}(\theta)$ to reduce $\ell^0_{\Omega_e}(\theta)$ by the amount corresponding to the data for which statement "0" was incorrect. Therefore, we apply the same Taylor approximation as above. That is,

$$\begin{aligned}
\ell^0_{\Omega_{\neg 0 \wedge e}}(\theta) \;&=\; \sum_{(i,j,s) \in \Omega_{\neg 0 \wedge e}} r_{ij} \ln\left(1 - \tilde{q}_k(\theta)\right) \\
&\approx\; \sum_{(i,j,s) \in \Omega_{\neg 0 \wedge e}} r_{ij} \left(\ln\left(\frac{p}{p + (1-p)\,\rho_{ijk_s}\bar{\nu}} \right) \right. \\
&\qquad \left. + \sum_{m=1}^{M} \frac{1}{m} \left(\frac{-(1-p)\,\rho_{ijk_s}\,(\nu_s - \bar{\nu})}{p + (1-p)\,\rho_{ijk_s}\bar{\nu}} \right)^m \right).
\end{aligned} \tag{A3.42}$$

This computation runs in $\mathcal{O}(M\,|\Omega_{\neg 0 \wedge e}|)$.

$\ell_{\Omega_{\neg 0 \wedge e}}(\theta)$: The number of non-zero observations is small. Therefore, we can compute $\ell_{\Omega_{\neg 0 \wedge e}}(\theta)$ directly:

$$\ell_{\Omega_{\neg 0 \wedge e}}(\theta) = \sum_{(i,j,s) \in \Omega_{\neg 0 \wedge e}} \ln f_{\mathrm{NB}}\left(n_{ijs} \,|\, r_{ij}(\theta), \tilde{p}_{ijs}(\theta)\right), \tag{A3.43}$$

which runs in $\mathcal{O}(|\Omega_{\neg 0 \wedge e}|)$.

$\ell_{\Omega_{\neg 0 \wedge \neg e}}(\theta)$: We have already noted that the set $\Omega_{\neg 0 \wedge \neg e}$ of traffic noise observations is small. Therefore, we compute $\ell_{\Omega_{\neg 0 \wedge \neg e}}(\theta)$ directly:

$$\ell_{\Omega_{\neg 0 \wedge \neg e}}(\theta) := \sum_{(i,j,s) \in \Omega_{\neg 0 \wedge \neg e}} \ln f_{\mathrm{NB}}\left(n_{ijs} \,|\, r_{ij}(\theta), \tilde{p}_{ijs}(\theta)\right). \tag{A3.44}$$

This runs in $\mathcal{O}(|\Omega_{\neg 0 \wedge \neg e}|)$.

In conclusion, we can compute the likelihood in $\mathcal{O}(M|S| + M|L||\Psi| + M|\Omega_{\neg 0}|)$. The set $\Omega_{\neg 0}$ contains all those observations that are non-zero and has a size proportional to $|S|$ in general.

There are more (minor) optimizations that can be applied to compute intermediate terms independent of θ before the optimization process. We do not list these details here.

3.D.5 Maximizing the likelihood

We apply different optimization techniques consecutively to maximize the likelihood. All algorithms we used were implemented in the Scipy package "optimize" version 1.0 (Jones et al., 2001). We started with the "differential evolution" algorithm by Storn and Price (1997), a meta-heuristic global optimization algorithm that does not require an initial guess. We chose the region of admissible parameters liberally. With the differential evolution result as initial guess, we applied the L-BFGS-G algorithm (Byrd et al., 1995), which proved to be robust and efficient even if the result from the genetic algorithm was far from the optimum. In a next step, we applied sequential least squares programming (Kraft, 1988) due to its high

efficiency and finally a trust-region Newton-Raphson method (Nocedal and Wright, 2006), which is guaranteed to converge very fast if the initial guess is close to the optimum.

Whenever necessary, we determined derivatives of the likelihood function using algorithmic differentiation in reverse mode, which is much more efficient and precise than numerical differentiation. We used the python package "autograd" for this task.

Though some of the optimization algorithms we applied can deal with constraints on parameters, we enforced constraints with parameter transformations. Let c be a parameter as it appears in the model, and $\tilde{c}$ the same parameter as used in computations.

- Parameters constrained to be positive were expressed as

$$c = \begin{cases} \exp \tilde{c} & \text{if } \exp \tilde{c} < 0 \\ \tilde{c} + 1 & \text{else.} \end{cases} \tag{A3.45}$$

- Parameters constrained to the interval $(0, 1]$ were expressed as $c = \frac{1}{\pi} \arctan(\tilde{c}) + \frac{1}{2}$.

This allowed us to avoid numerical instabilities arising when the results are close to the boundaries.

3.E Model selection and confidence intervals based on the composite likelihood

3.E.1 Model selection

We used an information criterion to determine which of our gravity models fits the data best without overfitting. The most widely used criteria for model selection (Aho et al., 2014) are the information criterion by Akaike (AIC, Akaike, 1974) and the Bayesian information criterion (BIC, Schwarz, 1978). Both AIC and BIC are based on the log-likelihood of the compared models.

When working with composite likelihood, as we did to determine the best structure for the gravity model, AIC and BIC loose their validity (Varin and Vidoni, 2005). However, the corrected information criterion that Varin and Vidoni (2005) derived for composite likelihood models is hard to compute. Furthermore, only a small portion of our data violate the independence assumption. Therefore, we proceeded using the classical model selection criteria.

3.E.2 Confidence intervals

For practical reasons, we computed the confidence intervals for our parameters (see Tables A3.2 and A3.3 below) under simplifying assumptions. The first simplification is that we determined the confidence intervals for each submodel individually. This approach measures the credibility of the fitted parameters under the assumption that the previously fitted submodels are known. However, if all submodels were fitted simultaneously, changes to the parameters of one model would also affect the parameter estimates for the other model. Consequently, the confidence intervals would increase. The second simplification is that we computed the confidence intervals based on the composite likelihood. Though this does not bias our parameter estimate, more sophisticated methods would be necessary to determine the confidence intervals accurately (Varin, 2008).

Though these limitations decrease the rigorous meaning of the confidence intervals we computed, the presented confidence intervals still provide valuable insights into the levels of credibility of our estimates, since only small portions of our data are dependent on each other. Since our primary goal is to estimate propagule pressure rather than building a mechanistic model, the heuristic nature of the confidence intervals is sufficient for our purposes.

3.F Details of the model for the inflow of potentially mussel-infested boaters to British Columbia

In this appendix, we provide details of the model that we used to estimate the number of potentially mussel-infested boats brought to BC. In the first section of this appendix, we describe the specific structure of the gravity model. In the second section, we present details of the fitted model and give parameter estimates along with confidence intervals.

3.F.1 The structure of the gravity model

The covariates available to fit the gravity model need to be appropriately combined to yield useful measures of the repulsiveness m_i of donor jurisdictions i and the attractiveness a_j of destination lakes j. As described in section 2.1, the specific functional form of the gravity model depends on assumptions on how the covariates interact with each other to make jurisdictions repulsive or lakes attractive. We list these assumptions below.

We assumed that the nation and the boater count of a jurisdiction act together in yielding high counts of travelling boaters. As the number of boaters residing in the jurisdictions is unknown, we tested both population and angler number as proxies for the boater number. For destination lakes, we assumed that both a sufficient size and presence of tourist facilities are necessary to attract many boaters. Thereby, the type of the facilities is of minor importance. We tested both lake area and lake perimeter as measures for the lake size. A list of the covariates and parameters can be found in Table A3.3.

Connecting all building blocks, we arrived at the following model for the daily mean number of travelling agents:

$$\begin{aligned}\mu_{ij} &= c \cdot \left(\frac{\mathrm{pop}_i}{\mathrm{pop}_i + \mathrm{pop}_0}\right)^{\alpha_{\mathrm{pop}}} \cdot \beta_{\mathrm{CA}}^{\mathrm{CA}_i} \cdot \left(\frac{A_j}{A_j + A_0}\right)^{\alpha_{\mathrm{A}}} \\ &\quad \cdot \left(1 + \beta_{\mathrm{camp}}\mathrm{camp}_j + \beta_{\mathrm{fac}}\mathrm{fac}_j + \beta_{\mathrm{mar}}\mathrm{mar}_j + \beta_{\mathrm{lpop}}\left(\frac{\mathrm{lpop}_j}{\mathrm{lpop}_j + \mathrm{lpop}_0}\right)^{\alpha_{\mathrm{lpop}}}\right) \cdot d_{ij}^{-\alpha_{\mathrm{d}}}. \quad \text{(A3.46)}\end{aligned}$$

3.F.2 Resulting model

Gravity model

The gravity model with minimal AIC value included 8 covariates and 11 parameters. The parameter values can be found in Table A3.3 along with their confidence intervals. Since gravity models are phenomenological models, the parameter values have limited meaning. Nonetheless, we can make some comparative statements concerning the roles of the different covariates in our model.

The submodel for the lake attractiveness a_j included the covariates lake area, presence of campgrounds, marinas, and other points of interest, and the population living close to the lakes. The presence of campgrounds weighed 45% more than the presence of "other facilities" (public toilets, viewpoints, etc.; see Table A3.3). The presence of a marina, in turn, weighed more than four times as much as the presence of a campground. An equally important factor for lake attractiveness was the population close to lakes: $23,800$ persons living in a 5 km buffer around the lake were equivalent to the presence of a marina.

The repulsiveness m_i of source jurisdictions was estimated based on their population count and nation. Canadian provinces were weighed about 15 times higher than American states. The numbers of anglers in the jurisdictions were not included.

The travel times between jurisdictions and recipient lakes had a huge effect on the expected numbers of travelling boaters. Numbers decreased in cubic order of the travel time.

Route choice model

The fitted route choice model suggests that boaters have a strong preference for the shortest route. According to the model, an alternative route only 10% longer than the shortest route attracts only half as many agents. The parameters for the best-fitting route choice model are displayed in Table A3.2.

Parameter	Parameter Explanation	Estimate	Profile CI	
γ	Maximal stretch of admissible paths	1.4	–	–
δ	Required local optimality of admissible paths	0.2	–	–
η_c	Probability to travel on an inadmissible path	0.049	0.013	0.05
η_o	Probability to choose a path via a given survey location if travelling on an inadmissible path	0.062	0.044	0.47
λ	Travel time exponent	7.4	6.53	8.29

Table A3.2: Parameters and estimates along with 95% confidence intervals for the route choice model. As η_c and η_o are not estimable, we bounded $\eta_c \leq 0.05$ to obtain the final parameter estimates. Since the likelihood function is not continuous in the parameters γ and δ and computing admissible routes is computationally expensive, we did not construct confidence intervals for these parameters.

The probability η_c that boaters choose an inadmissible route and the probability η_o that such boaters drive via a survey location are not estimable: we do not know how many boaters went along inadmissible routes that were not covered by a survey station. Hence, we cannot draw inference on traffic along inadmissible routes (see Appendix 3.D.3).

Temporal pattern model

We used a von Mises distribution stretched over the 24 hours of the day to model temporal variations in traffic density. The estimated traffic peak was at 2 : 00 PM with 95% confidence interval [1:48 PM, 2:20 PM]. For the scale parameter, which determines how "spiky" the traffic peak is, we obtained a value of 1.34 with confidence interval [1.11, 1.56]. This implies that the boater traffic density during mid-day is about 15 times as high as at night. The probability density function of the traffic time model is plotted in Figure 4 in the main text.

Covariate	Covariate Explanation	Parameter	Estimate	Profile CI	
–	Scaling factor	c	$3.73e_{-8}$	$2.36e_{-8}$	$5.83e_{-8}$
–	$^{\text{mean}}/_{\text{variance}}$	p	0.23	0.21	0.25
pop_i	Population of jurisdiction i [$1e_6$]	pop_0	0.16	0.09	0.26
		α_{pop}	1	–	–
CA_i	1 if jurisdiction i is Canadian, else 0	β_{CA}	14.79	12.82	17.15
camp_j	1 if major campgrounds are present at lake j, else 0	β_{camp}	6.55	4.66	9.3
fac_j	1 if other facilities (toilets, viewpoints, tourist infos, parks, attractions, picnic sites) are present at lake j, else 0	β_{fac}	4.51	3.04	6.63
mar_j	1 if marinas are present at lake j, else 0	β_{mar}	26.4	19.41	36.59
lpop_j	Population living closer than 5km to the lake j [$1e_3$]	β_{lpop}	1011	396	$> 6e_9$
		lpop_0	888	318	$> 1e_{10}$
		α_{lpop}	1	–	–
A_j	Area of lake j [km^2]	A_0	1236	1044	1464
		α_{A}	1	–	–
d_{ij}	Shortest traveltime between jurisdiction i and lake j [$1e_4$ min]	α_{d}	3.45	3.35	3.54

Table A3.3: Covariates, parameters, and estimated parameter values along with 95% confidence intervals for the best-fitting gravity model. Parameters without confidence intervals ("–") were not part of the model with the best AIC value and fixed beforehand. Further covariates tested but not included in the model with the best AIC value were the numbers of anglers in jurisdictions and the lake perimeters. Refer to Appendix 3.H for a discussion of the large confidence intervals for β_{lpop} and lpop_0.

Compliance model

The estimated proportion of boaters participating in the survey was 80%. Out of these boaters, 93% delivered consistent and complete data. The overall rate of boaters providing useful information was thus 74.4%.

3.G Model validation

In this appendix, we present model validation results and the methods that we applied to obtain these results. Specifically, we confirm that the distribution choices for our temporal pattern model (von Mises distribution) and the count data (negative binomial distribution) are appropriate. Furthermore, we check our model for an overall bias and assess the precision of the model's predictions. We start with a description of our methods, continue with the results, and conclude the Appendix with a short discussion of both validation results and methods.

3.G.1 Methods

Before we start describing our methods in detail, we make a general note on model validation. In general, it is hard to apply classical hypothesis testing for model validation, as the distribution of the data under the null hypothesis "model is incorrect" is unknown. We therefore validate our model by ascertaining that it cannot be rejected on a high confidence level. That is, our null hypothesis is "model is correct", and high p-values indicate that the test statistic results computed with the data we observed are likely to occur if the model is correct. Though this method can provide some insights into whether the model is appropriate, the approach does not yield a rigorous measure for the model validity. Therefore, we also perform validation steps based on graphical comparison.

3.G.1.1 Homogenized samples

Some of the tests we are about to apply require samples from count data distributions. That is, we need a set of independent and identically distributed (i.i.d.) observations. Both the survey location and the survey time affect the distribution of count data of observed agents. Therefore, we will get i.i.d. observations only if we consider count data collected at the same survey location and during the same time interval.

We generated such samples by considering count data collected in a time interval that overlapped with many of our observation shifts. We proceeded as follows:

1. We considered all survey shifts that started at 11AM or earlier and ended at 4PM or later. We neglected all other survey shifts.

2. For each of the above survey shifts, we counted the agents surveyed between 11AM and 4PM.

3. For each survey location, we noted how many survey shifts were considered in step 2. To ensure we had enough data for a meaningful statistical analysis, we neglected samples with sizes below 20.

3.G.1.2 Shape of the temporal traffic pattern

In this section, we describe a test to check whether our temporal traffic model has an appropriate shape. We used the von Mises distribution to model the temporal variations of agent traffic. This distribution has a specific unimodal shape. This shape may differ significantly from the observed traffic profile, which may have multiple peaks.

To ensure that the von Mises distribution is appropriate to model the daily traffic pattern, we compared it to fitted step function distributions, which do not have a predefined shape. Let $I_{\text{tot}} = [t_0, t_n]$ denote the portion of the day that was covered by at least one survey shift. A step function distribution splits I_{tot} in n equally sized disjoint intervals $I_1, \ldots, I_n \subseteq I_{\text{tot}}$

with $I_i = [t_{i-1}, t_i)$. The probability density function is given by

$$f_{\text{step}}\left(t\middle|p_1, \dots p_n\right) = \begin{cases} p_1 & \text{if } t \in I_1 \\ \vdots & \vdots \\ p_n & \text{if } t \in I_n. \end{cases} \tag{A3.47}$$

We fitted the parameters p_i with a maximum likelihood approach and repeated this procedure for distributions with different interval numbers n. Then, we compared the resulting AIC values and probability density functions with the best-fit von Mises distribution. Both the AIC value and graphical comparison yield insights into whether the von Mises distribution is appropriate.

3.G.1.3 Distribution of count data

In this section, our goal is to check whether the negative binomial distribution is appropriate to model the distribution of our count data. To that end, we use the homogenized count data described in section 3.G.1.1. We have count data $x_i = \{x_{i1}, \dots, x_{in_i}\}$ for different origin destination pairs, obtained at different locations. Here, i enumerates all combinations of origins, destinations and sampling locations for which we have sufficient data. The numbers $n_i \in \mathbb{N}$ denote the respective sample sizes. Below, we write X_i for the random variable that x_i has been drawn from.

Since we expect that both the sampling location as well as origin and destination affect the count distribution, we need to check that all data come from negative binomial distributions, i.e. $X_i \sim NB(\mu_i, p_i)$, without assuming that all data come from the *same* distribution. That is, μ_i and p_i may differ dependent on i. In this section, we describe a method to test this hypothesis.

Famoye (1998) compared the power of different empirical distribution function tests to test whether observations come from a generalized negative binomial distribution. In their

simulations, the discrete Anderson-Darling test performed best. The Anderson-Darling test compares the cumulative mass function (cmf) of a null distribution with its empirical counterpart generated from the considered sample. Thereby, the Anderson-Darling test puts higher weight on the tails of the distribution than other comparable tests, like the Kolmogorov-Smirnov test. If the empirical and the hypothesized cmf differ significantly, the null hypothesis is rejected.

The distribution of the Anderson-Darling statistic is known for fully specified continuous null distributions. We, however, consider a discrete distribution and do not have prior knowledge of the parameters. Instead, we are only interested in whether the observed data come from *some* negative binomial distribution. To generate the cmf of the null distribution, which is needed for comparison with the empirical cmf, we would have to estimate the distribution's parameters first. This, in turn, affects the distribution of the test statistic.

We are not aware of any result providing a closed-form expression for the distribution of the Anderson-Darling statistic applied to negative binomial random variables. Therefore, we determine the p-values for our samples by adjusting the the parametric bootstrap procedure used by Famoye (1998). Parametric bootstrap methods approximate the distribution of a test statistic by repeated application of the statistic to samples randomly generated from the null distribution. Therefore, parametric bootstrap methods are not exact but easy to implement.

The p-value of a test statistic T applied to a sample x_i is the probability to observe $T(x_i)$ if the null hypothesis is true. Consequently, a high p-value indicates that the null distribution may be appropriate to model the data. Thus it seems reasonable to assume that if the p-values for all individual samples x_i, $i \in \{1, \ldots, N\}$, are large, the null distribution can be assumed to be a good model for all our count data. This is the main idea of our approach.

Note that since each computed p-value depends on the randomly drawn sample x_i, the p-values itself are random variables as well. To test our count distribution hypothesis on all N samples, we may check whether the N computed p-values come from the distribution of

p-values that we would expect under the null hypothesis. By this means, we could summarize all individual tests in one joint test.

For such a joint test, we need to know the distribution of p-values under the null hypothesis. Since, by construction of the p-value, 80% of the samples randomly drawn from the null distribution lead to a p-value less than or equal to 0.8, 60% of the samples lead to a p-value less than or equal to 0.6, and so on, it is intuitive to assume that the p-values follow a uniform distribution on the interval $(0, 1]$. This is in fact true for continuous null distributions. For samples from discrete distributions, however, things are more complicated.

Discrete random variables attain their values with positive probabilities. Hence, the same applies to samples drawn from this distribution and thus for computed p-values. Suppose, for example, that we have drawn a sample x_i from the null distribution and computed the p-value $\phi(x_i)$, say $\phi(x_i) = 1$. Then the probability to obtain a p-value of 1 is at least $\mathbb{P}(x_i)$, which could be arbitrarily high. In fact, since samples taken from a single distribution are permutation-invariant, $\mathbb{P}(\phi(X_i) = 1)$ can attain relatively large numbers in practice. Therefore, the distribution of $\phi(X_i)$ may not even be close to a uniform distribution, and we have to determine the distribution of $\phi(X_i)$ under the null hypothesis before we can test our joint hypothesis.

We present our overall approach by breaking it down into parts. First, we describe the parametric bootstrap algorithm we use to compute p-values for a single sample x_i. Then, we show how we estimate the joint distribution of the p-values for all samples $x_1, \ldots, x_N$. Third, we describe how a second parametric bootstrap procedure can be applied to compute the p-value for our joint hypothesis. In a fourth step, we study the distribution of our count data samples under the null hypothesis and provide computationally efficient parameter estimators. Fifth, we describe how random numbers can be drawn from the null distribution. We conclude this section by showing how partial results can be reused to speed up computations and discussing how the accuracy of the resulting p-value can be determined.

3.G.1.3.1 Computing p-values with the Anderson-Darling test for a null distribution with unknown parameters

In this subsection, we describe the parametric bootstrap procedure based on Famoye (1998) that we apply to determine the p-values for the Anderson-Darling tests for individual samples. Let $T(x_i, \theta)$ be the function that maps a sample x_i to the Anderson-Darling statistic based on the null distribution with parameters θ. Furthermore, let $\Theta(x_i)$ be an estimate of the parameters θ of the null distribution based on sample x_i. Let x_0 be the sample that we want to study, $n := |x_0|$, and $M_1 \in \mathbb{N}_+$ be a positive integer. Throughout this Appendix, $|A|$ denotes the number of entries in a vector or set A.

The parametric bootstrap method works as follows:

1. Use the sample x_0 to find an estimate $\hat{\theta}_0 := \Theta(x_0)$ of the parameters of the null distribution.

2. Compute the test statistic $t_0 := T(x_0, \hat{\theta}_0)$ under the null distribution with the fitted parameters.

3. Generate M_1 samples $\tilde{x}_i$, $i := 1, \ldots, M_1$, of size n from the null distribution with parameters $\hat{\theta}_0$.

4. For each generated sample $\tilde{x}_i$:

   (a) Find an estimate of the parameters $\hat{\theta}_i := \Theta(\tilde{x}_i)$ based on sample $\tilde{x}_i$.

   (b) Compute $t_i := T(\tilde{x}_i, \hat{\theta}_i)$.

5. The approximate p-value is given by the fraction of samples that had an at least equally large test statistic: $\phi(x_0) := \frac{1}{M_1} |i : t_i \geq t_0|$.

3.G.1.3.2 Determining the null distribution of p-values

To test which distribution of p-values we would expect under the null distribution, we apply a Monte Carlo simulation. That is, we draw many samples from the null distribution and determine the respective p-values. Then, we determine the empirical distribution function of these samples.

Recall that the null-distribution of p-values may be different for each sample x_i, because we do not require that all samples come from the same distribution. Therefore, the true distribution of p-values is a multi-variate distribution. However, to compute a statistic from the samples, we have to reduce the dimension somehow. We therefore consider the random variable Φ resulting from the following random process:

1. Choose $i \in \{1, \ldots, N\}$ randomly from a uniform distribution.
2. Set $\Phi = \phi(X_i)$.

That is, we suppose that Φ assumes p-values from each dimension with the same probability.

To ease the explanation of our method, let us now assume that the parameters θ_i of the null distribution for sample i are known, i.e., that the null distribution is fully specified. We will extend the method to not fully specified null distributions in the next section.

Let $x_1, \ldots, x_N$ be our count data samples, and let $n_i := |x_i|$ and $M_2 \in \mathbb{N}_+$ be a positive integer. To determine the distribution of Φ under the null hypothesis, we proceed as follows:

1. For $i \in \{1, \ldots, N\}$:
   (a) Given the parameters θ_i of the null distribution for sample i, draw M_2 samples $\tilde{x}_{ij}$, $j := 1, \ldots, M_2$, of size n_i from the null distribution.
   (b) Determine $\phi(\tilde{x}_{ij})$ as described in section 3.G.1.3.1.
2. The probability mass function $\hat{f}_\Phi$ of Φ is approximately given by $\mathbb{P}(\Phi = p) := \frac{1}{NM_2} |i, j : \phi(\tilde{x}_{ij}) = p|$.

3.G.1.3.3 Computing the p-values for the joint hypothesis

In the previous subsection, we have shown how the distribution of p-values under the null hypothesis can be estimated if the null distribution is fully specified. We, however, need to know the distribution of the p-values if the parameters $\theta_1, \ldots, \theta_N$ are unknown. Therefore, we have to apply a second level of parametric bootstrap to test the joint hypothesis that all data come from negative binomial distributions.

Again, let $x = (x_1, \ldots, x_N)$ be our count data samples, and let $n_i := |x_i|$ and $M_3 \in \mathbb{N}_+$ be a positive integer. Furthermore, let $\Theta(x_i)$ be the estimate of the parameters θ of the null distribution based on sample x_i, and let $T(y, f)$ be a statistic suitable to test whether sample y comes from a distribution with probability mass function (pmf) f. We proceed as given below:

1. For $i \in \{1, \ldots, N\}$:
   (a) Find an estimate $\hat{\theta}_i := \Theta(x_i)$ of the parameters of the null distribution.
   (b) Find the p-value $p_i := \phi(x_i)$ using the method from section 3.G.1.3.1.
2. With $\hat{\theta} := \left(\hat{\theta}_1, \ldots, \hat{\theta}_N\right)$ compute $\hat{f}_\Phi$ as described in section 3.G.1.3.2.
3. With $p := (p_1, \ldots, p_N)$, determine $t_0 := T(p, \hat{f}_\Phi)$.
4. For $j \in \{1, \ldots, M_3\}$:
   (a) For $i \in \{1, \ldots, N\}$, draw a sample $\tilde{x}_{ij}$ of size n_i from the null distribution with parameters $\hat{\theta}_i$.
   (b) Compute t_j with the steps 1-3 applied to the joint sample $\tilde{x}_j := (\tilde{x}_{1j}, \ldots, \tilde{x}_{Nj})$.
5. The approximate p-value for the joint hypothesis is given by the fraction of samples that had an at least equally large test statistic: $\phi(x) := \frac{1}{NM_3} |j : t_j \geq t_0|$.

3.G.1.3.4 Estimating the parameters

The procedures outlined above require parameter estimates that fit the observed data well. In this subsection, we describe how the parameters can be estimated efficiently based on sample data. However, as not all samples may contain useful information to test our base hypothesis, we start by discussing how ignoring non-informative samples could be of advantage.

Samples consisting only of zero-counts do not contain useful information on the distribution family they have been drawn from. Many distribution families have parameters that make zeros arbitrarily likely. Therefore, we would be unable to determine from which of these distributions a zero-sample has been drawn from. As a consequence, considering zero-samples could decrease the power of a test applied to check from which distribution family samples were drawn. For this reason (and to save computation time), it is beneficial to neglect samples consisting of zeros only and to focus on samples with at least one non-zero observation.

Considering only samples with at least one non-zero observation changes the hypothesized null distribution. Even if the true distribution yields zero-samples frequently, we will only consider samples with at least one non-zero observation. We therefore have to adjust our parameter estimates accordingly.

Our goal is to check whether the negative binomial distribution is appropriate to model our count data. If we disregard zero-samples, we therefore consider a negative binomial distribution conditioned such that zero-samples are impossible. In this paper, we call this distribution the "zero-sample truncated negative binomial distribution" (ZSTNB).

Besides the ZSTNB, we also regard the analogously defined "zero-sample truncated Poisson distribution" (ZSTP), which is a limiting distribution of the ZSTNB. The ZSTP is important if parameter estimates for the ZSTNB do not exist. In addition, we use the ZSTP to check the power of our approach. In this subsection, we focus on deriving estimators for the

parameters of the ZSTNB and ZSTP, whereas we provide instructions on how to generate samples from these distributions in the subsection below.

There are different methods to estimate parameters based on a sample of observations. Commonly used techniques are maximum likelihood estimation and method of moment estimation (Casella and Berger, 2002). While maximum likelihood estimators have favourable statistical properties and are highly efficient in general, the method of moment estimators are often easier to compute. Because our testing procedure requires us to estimate parameters an excessive number of times, we follow Famoye (1998) in estimating parameters with the method of moments.

The idea behind the method of moments is to compute the moments of a distribution based on its parameters θ and equate the results with the respective sample moments. Then, the parameter estimates $\hat{\theta}$ are computed by solving this equation system. For example, consider a distribution with the parameters θ_1 and θ_2 and let μ_S and σ_S^2 be the sample mean and variance. The true mean μ and variance σ^2 of the distribution can be computed as functions of the parameters:

$$\begin{aligned} \mu &= g_\mu(\theta_1, \theta_2) \\ \sigma^2 &= g_{\sigma^2}(\theta_1, \theta_2). \end{aligned} \tag{A3.48}$$

The method of moments parameter estimates $\hat{\theta}_1$ and $\hat{\theta}_2$ are computed by replacing μ and σ^2 on the right hand side of equation system (A3.48) with their respective sample equivalents μ_S and σ_S^2 and solving the system for θ_1 and θ_2.

Before we proceed, we formalize the notion of zero-sample truncated distributions.

Definition 3.1. Let $Y := (Y_1, \ldots, Y_n)$ be a random vector consisting of independently and identically distributed random variables. Then we say that $X := Y \,\big|\, (\exists i \in \{1, \ldots, n\} : Y_i \neq 0)$ follows a zero-sample truncated distribution.

Strictly speaking, zero-sample truncated distributions are multivariate distributions, because the individual sampling results X_i are not independent of each other. What we regarded as a sample consisting of multiple identical independent draws before turns out to be a *single* draw from a multivariate distribution. Therefore, the distribution does not have a univariate mean and variance, as would be required for the method of moments.

However, we can still apply the method of moments if we consider quantities analog to the sample mean and variance in the univariate case. Let $X := (X_1, \ldots, X_n)$ be a zero-sample truncated random variable derived from the independent random variables $Y := (Y_1, \ldots, Y_n)$ with probability mass function (pmf) f, mean μ, and variance σ^2 respectively. Our sample mean $\mu_S = \frac{1}{n}\sum_i X_i$ and sample variance $\sigma_S^2 = \frac{1}{n-1}\sum_i (X_i - \mu_S)^2$ will resemble the mean and variance of a single entry of X. Therefore, we make the following definitions:

Definition 3.2. We say that $\bar{f}(x_1) := \mathbb{P}(X_1 = x_1)$ is the sample pmf and $\bar{F}(x_1) := \mathbb{P}(X_1 \leq x_1)$ is the sample cmf.

Definition 3.3. We say that $\bar{\mu} := \mathbb{E}(X_1)$ is the expected sample mean, and $\bar{\sigma}^2 := \mathbb{V}(X_1)$ is the expected sample variance.

Note that the index "1" used above is not of importance, because the random variables $X_1, \ldots, X_n$ are identically distributed and thus exchangeable.

To ease computation of the quantities defined above, we make the following observations:

Lemma 3.1. *It is* $\bar{f}(x_1) = \begin{cases} \frac{f(0)-f(0)^n}{1-f(0)^n} & \text{if } x_1 = 0 \\ \frac{f(x_1)}{1-f(0)^n} & \text{else.} \end{cases}$

Proof. If $x_1 = 0$, then

$$\begin{aligned}
\bar{f}(0) &= \mathbb{P}(X_1 = 0) \\
&= \mathbb{P}\big(Y_1 = 0 \big| \exists i \in \{1, \dots, n\} : Y_i \neq 0\big) \\
&= \frac{\mathbb{P}(Y_1 = 0 \wedge \exists i \in \{1, \dots, n\} : Y_i \neq 0)}{1 - \mathbb{P}(Y = 0)} \\
&= \frac{\mathbb{P}(Y_1 = 0 \wedge \exists i \in \{2, \dots, n\} : Y_i \neq 0)}{1 - \mathbb{P}(Y = 0)} \\
&= \frac{\mathbb{P}(Y_1 = 0)\,(1 - \mathbb{P}(Y_1 = 0)^{n-1})}{1 - \mathbb{P}(Y_1 = 0)^n} \\
&= \frac{f(0) - f(0)^n}{1 - f(0)^n}.
\end{aligned}$$

Here, we used that the entries of the vector Y are identically independently distributed.

If $x_1 \neq 0$, then $\exists i \in \{1, \dots, n\} : Y_i \neq 0$. Hence,

$$\begin{aligned}
\bar{f}(x_1) &= \mathbb{P}(X_1 = x_1) \\
&= \mathbb{P}\big(Y_1 = x_1 \big| \exists i \in \{1, \dots, n\} : Y_i \neq 0\big) \\
&= \frac{\mathbb{P}(Y_1 = x_1 \wedge \exists i \in \{1, \dots, n\} : Y_i \neq 0)}{1 - \mathbb{P}(Y = 0)} \\
&= \frac{\mathbb{P}(Y_1 = x_1)}{1 - \mathbb{P}(Y_1 = 0)^n} \\
&= \frac{f(x_1)}{1 - f(0)^n}.
\end{aligned}$$

This concludes the proof. □

Corollary 3.1. *It is* $\bar{\mu} = \frac{\mu}{1 - f(0)^n}$ *and* $\bar{\sigma}^2 = \frac{\sigma^2 + \mu^2}{1 - f(0)^n} - \bar{\mu}^2 = \frac{\sigma^2}{1 - f(0)^n} - \frac{f(0)^n \mu^2}{(1 - f(0)^n)^2}$.

Proof. Direct computation yields

$$\begin{aligned}
\bar{\mu} &= \mathbb{E}(X_1) \\
&= \sum_{i \in \mathbb{N}_+} i \frac{f(i)}{1 - f(0)^n} \\
&= \frac{1}{1 - f(0)^n} \sum_{i \in \mathbb{N}_+} i f(x_1) \\
&= \frac{\mu}{1 - f(0)^n}.
\end{aligned}$$

Similarly, for the expected sample variance,

$$\begin{aligned}
\bar{\sigma}^2 &= \mathbb{E}\left(X_1^2\right) - \bar{\mu}^2 \\
&= \sum_{i \in \mathbb{N}_+} i^2 \frac{f(i)}{1 - f(0)^n} - \bar{\mu}^2 \\
&= \frac{1}{1 - f(0)^n} \sum_{i \in \mathbb{N}_+} i^2 f(i) - \bar{\mu}^2 \\
&= \frac{1}{1 - f(0)^n} \left(\sigma^2 + \mu^2\right) - \bar{\mu}^2 \\
&= \frac{\sigma^2}{1 - f(0)^n} - \frac{f(0)^n \mu^2}{\left(1 - f(0)^n\right)^2}.
\end{aligned}$$

□

Now we can apply our general findings to find method of moments estimators for the parameters of the ZSTNB and ZSTP. For convenience, we parameterize the negative binomial distribution with the parameters r and p as described in Appendix 3.D.4.1 (equation (A3.17)).

We start by considering the expected sample mean of the ZSTNB. The negative binomial distribution has mean $\mu = \frac{r(1-p)}{p}$ and variance $\sigma^2 = \frac{r(1-p)}{p^2}$. Hence,

$$\begin{aligned}
\bar{\mu} &= \frac{\mu}{1 - f(0)^n} \\
&= \frac{r\,(1-p)}{p\,(1 - p^{rn})}.
\end{aligned} \tag{A3.49}$$

This is equivalent to

$$1 - p^{rn} = \frac{\mu}{\bar{\mu}}. \tag{A3.50}$$

For the expected sample variance, we get

$$\begin{aligned} \bar{\sigma}^2 &= \frac{\sigma^2 + \mu^2}{1 - p^{rn}} - \bar{\mu}^2 \\ \text{with (A3.50)} \quad &= \frac{\bar{\mu}\left(\sigma^2 + \mu^2\right)}{\mu} - \bar{\mu}^2 \\ &= \bar{\mu}\frac{1 + r\left(1 - p\right)}{p} - \bar{\mu}^2, \end{aligned} \tag{A3.51}$$

which is equivalent to

$$r = \frac{p\left(\bar{\sigma}^2 + \bar{\mu}^2\right) - \bar{\mu}}{\bar{\mu}\left(1 - p\right)}. \tag{A3.52}$$

Inserting (A3.52) in (A3.49) leads after some algebra to

$$\begin{aligned} 0 &= \frac{\bar{\mu}}{p} - \bar{\mu}^2 p^{rn} - \bar{\sigma}^2 \\ \text{with (A3.52)} \quad &= \frac{\bar{\mu}}{p} - \bar{\mu}^2 p^{n\frac{p\left(\bar{\sigma}^2 + \bar{\mu}^2\right) - \bar{\mu}}{\bar{\mu}(1-p)}} - \bar{\sigma}^2, \end{aligned} \tag{A3.53}$$

which can be numerically solved for p if the expected sample mean and variance $\bar{\mu}$ and $\bar{\sigma}^2$ are replaced with the observed sample mean and variance μ_S and σ_S^2.

It can be shown with basic techniques that equation (A3.53) has at most two zeros in the interval $(0, 1)$, one of which is $p_l := \frac{\bar{\mu}}{\bar{\sigma}^2 + \bar{\mu}^2}$. However, inserting p_l in equation (A3.52) would lead to an r-estimate of 0. We know that $r > 0$. Therefore, p_l cannot be a valid parameter estimate. Because $r > 0$, we also know that the true estimate $\hat{p}$ must be larger than p_l. We thus can use a simple bisection method to find $\hat{p}$ in the interval $(p_l, 1)$. After computing $\hat{p}$ by this means, we insert $\hat{p}$ into equation (A3.52) to get our estimate $\hat{r}$ for the parameter r.

It is possible that the method of moments estimator $\hat{p}$ does not exist. This happens if equation (A3.53) does not have a root in $(p_l, 1)$, which is the case if, and only if,

$$0 \;>\; \bar{\mu} - \bar{\mu}^2 e^{-n\left(\frac{\bar{\sigma}^2}{\bar{\mu}} + \bar{\mu} - 1\right)} - \bar{\sigma}^2. \tag{A3.54}$$

If this happens, we will assume that the sample came from the ZSTP, which is a limiting case of the ZSTNB. As we will see below, the methods of moments estimator exists for the ZSTP in most instances.

We proceed by deriving an estimator for the parameter of the ZSTP. Oftentimes, the Poisson distribution is directly parameterized by its mean μ. The expected sample mean is given by

$$\begin{aligned} \bar{\mu} &= \frac{\mu}{1 - f(0)^n} \\ &= \frac{\mu}{1 - e^{-n\mu}}, \end{aligned} \tag{A3.55}$$

which is equivalent to

$$\mu \;=\; \frac{1}{n} \mathrm{W}\left(-n\bar{\mu}e^{-n\bar{\mu}}\right) + \bar{\mu}. \tag{A3.56}$$

Here, W denotes the Lambert W-function, which is the inverse function of $h(W) := We^W$. Packages with efficient implementations of the Lambert W-function exist for many programming languages. This makes it easy to compute the parameter estimate for μ.

The right hand side of equation (A3.56) assumes a real value if $\bar{\mu} > \frac{1}{n}$. However, if, and only if, there is exactly one non-zero count value in the sample and this count value is 1, then $\bar{\mu} = \frac{1}{n}$. In this case, the parameter estimate does not exist, because the sampling result could be made arbitrarily likely by choosing a very small value for μ. Therefore, we adjust the procedures outlined in the sections above so that samples with $\bar{\mu} = \frac{1}{n}$ always lead to p-values

of 1. Furthermore, we say that the distribution of p-values based on a null distribution whose parameters were estimated based on such a sample returns 1 with probability 1.

3.G.1.3.5 Generating random numbers

In this subsection, we describe algorithms to draw random numbers $(x_1, \ldots, x_n)$ form the ZSTNB and the ZSTP. Drawing numbers from these distributions is a crucial component of the algorithms described in the subsections above. Though efficient random number generators are available for the negative binomial distribution and the Poisson distribution, drawing numbers from zero-sample truncated distributions is a more complicated task. However, with a combination of the algorithms given below, samples can be generated with almost the same efficiency as samples from the "classical" negative binomial and Poisson distribution.

The naive approach to drawing samples from zero-sample truncated distributions is to generate samples from the original distribution until a sample with at least one non-zero entry is obtained. This approach is very efficient if the probability that the sample consists of zeros only is small. If f is the pmf of the original function and n is the sample size, then $f(0)^n$ is the probability that the sample consists of zeros only. If this quantity is small, only few samples have to be generated until a suitable one is found. Therefore, the alternative approaches below should be applied only if $f(0)^n$ is large.

To avoid an excessive number of trials until a suitable sample is found, we propose to first draw the sum $x_\Sigma := \sum_{i=1}^{n} x_i$ of all entries of the sample and to determine the values of the summands afterwards. Recall that $x_\Sigma \neq 0$ for zero-sample truncated distributions. For both the negative binomial and the Poisson distribution, the sum of n independent and identical trials is known to be negative binomially and Poisson distributed as well. Hence, the distribution of x_Σ, which is constrained to be positive, is easy to derive. If f_Σ is the pmf of the random variable Y_Σ modelling the sum of n independent draws from the original

distribution and X_Σ is the random variable from which x_Σ is drawn, then for $x_\Sigma \neq 0$,

$$\begin{aligned} \mathbb{P}(X_\Sigma = x_\Sigma) &= \mathbb{P}\big(Y_\Sigma = x_\Sigma \big| Y_\Sigma \neq 0\big) \\ &= \frac{f_\Sigma(x_\Sigma)}{f_\Sigma(0)}. \end{aligned} \tag{A3.57}$$

If $f(0)^n$ is large, $\frac{f_\Sigma(x_\Sigma)}{f_\Sigma(0)}$ is usually small, unless x_Σ is small. We therefore suggest the following procedure:

1. Compute a high quantile $x_{\max}$, e.g. the $q = 0.99999$ quantile, of Y_Σ.

2. For $1 \leq x_\Sigma \leq x_{\max}$, compute $\mathbb{P}(X_\Sigma = x_\Sigma)$.

3. Draw an integer x_Σ, $1 \leq x_\Sigma \leq x_{\max}$, according to the probabilities computed above.

Using $x_{\max}$ as upper bound for x_Σ introduces a potential error, because for both the negative binomial and the Poisson distribution arbitrarily high values occur with a positive probability. However, bounding x_Σ makes it easy to apply common random number generators to draw from a zero-truncated distribution. If a hard boundary for the error introduced by using a finite $x_{\max}$ is desired, the quantile q can be chosen as $q = f(0) + (1 - \epsilon)\,(1 - f(0))$. Then, a value larger than $x_{\max}$ occurs only with probability ϵ.

After drawing the sum x_Σ, we need to determine the individual count values x_i. For small values of x_Σ, only few different configurations of count values are possible. Each of these configurations has a probability, which can be computed easily. Then, the final configuration can be drawn according to these probabilities. For large values of x_Σ, we propose to use a Metropolis-Hastings algorithm to determine the final configuration. We provide details below.

If $x_\Sigma = 1$, we can just set $x_1 := 1$ and $x_i := 0$ for $2 \leq i \leq n$. The order of the sample does not matter in this paper. Therefore, it is appropriate to set the first entry to 1 always. If, for a different application, the order of the entries is important, a random shuffling algorithm can be applied to make the ordering unbiased.

If $x_\Sigma = 2$, there are two possible configurations: 2 entries of 1 or 1 entry of 2 while all remaining entries of the sample are 0, respectively. The probabilities for these configurations are easy to compute. Since the computations are simple but tedious, we do not present them here. After the probabilities of the configurations have been determined, the configuration of the sample is drawn randomly according to the probabilities.

If $x_\Sigma = 3$, the number of possible configurations is still small and the respective probabilities are easy to compute explicitly. As above, we do not present the computations here. The final configuration is then drawn according to the computed probabilities.

As $x_\Sigma \geq 4$ becomes large, the number of possible configurations increases quickly. In practice it happens rarely that $x_\Sigma \geq 4$ if $f(0)^n$ is large. In fact, often $x_{\max} < 4$. Nonetheless, dependent on when $f(0)^n$ is considered large, it can indeed happen that $x_\Sigma \geq 4$. In this case, we propose to use a Metropolis-Hastings algorithm to determine the configuration of the sample. This algorithm accepts and rejects changes to a given distribution based on the likelihood ratio of the original and new sample. The algorithm is as follows:

1. Set $x := (x_1, \ldots, x_n)$ to some arbitrary initial condition with $\sum_{i=1}^n x_i = x_\Sigma$.
2. Randomly draw two distinct indices i, j with $1 \leq i, j, \leq n$, $x_i \neq 0$, and $x_i \neq x_j$.
3. Create a copy x' of x and set $x'_i := x_i - 1$ and $x'_j := x_j + 1$.
4. Determine $\mathbb{P}(x')$ and $\mathbb{P}(x)$.
5. If $\mathbb{P}(x') \geq \mathbb{P}(x)$ set $x := x'$. Otherwise, set $x := x'$ with probability $\frac{\mathbb{P}(x')}{\mathbb{P}(x)}$.
6. Repeat steps 2 to 5 a large number of times.

If a sample from the ZSTP distribution shall be drawn, the process can be replaced by a simple draw from a multinomial distribution with n bins and uniform probabilities $\frac{1}{n}$.

3.G.1.3.6 Reusing partial results

The approach outlined above requires us to draw $M_1 M_2 M_3$ samples for each count sample x_i, $i = 1, \ldots, N$, and to determine parameter estimates and evaluate the Anderson-Darling statistic for each of these samples. Hence, the nested parametric bootstrap method is computationally costly. However, computations can be sped up if earlier results are reused.

As the distribution for the samples $x_i = (x_{i1}, \ldots, x_{in_i})$ is permutation-invariant, the only information that we use is how often each possible count value occurred. That is, if $\nu_{ik} := |j : x_{ij} = k|$, then a set $\nu_i := \left\{(k, \nu_{ik}) \,\middle|\, \nu_{ik} \neq 0\right\}$ containing all non-zero ν_{ik} suffices to describe x_i. Furthermore, each sample x_i has a specific parameter estimate $\Theta(x_i)$, statistic value $T(x_i, \Theta(x_i))$, and p-value $\phi(x_i)$ associated to it. Therefore, it is sufficient to compute $\Theta(x_i)$, $T(x_i, \Theta(x_i))$, $\phi(x_i)$ for each ν_i only once. This can be implemented efficiently via hash-maps with hashes of ν_{ik} as keys.

Dependent on which partial results are reused, reusing results can lead to precision loss of the overall algorithm. The quantities $\Theta(x_i)$ and $T(x_i, \Theta(x_i))$ are computed with deterministic algorithms. Therefore, reusing these quantities comes with no additional cost. The p-values $\phi(x_i)$, however, are computed with a parametric bootstrap technique. Therefore, reusing these results can lead to an increased variance of the results. Nonetheless, the performance gain obtained from reusing partial results usually outweighs the precision loss. In fact, since p-values do not have to be computed as frequently if results are reused, a large value M_1 can be chosen with minor increase in computation time. This usually leads to more precise results in the end.

3.G.1.3.7 Determining the accuracy of the approach

The nested bootstrap method for testing the distribution of count data is based on frequent resampling and therefore subject to error. The p-value resulting from the nested bootstrapping is a random variable. The variance of the result can be arbitrarily decreased by choosing

large sample numbers M_1, M_2, and M_3. Nonetheless, it would be desirable to get an estimate of the error. We therefore suggest to repeat the procedure M_4 times and to determine the standard deviation of the resulting p-values as measure for the error.

Repeating the procedure also decreases the error further, as the resulting mean value will be close to the actual p-value than each result individually. Since the nested bootstrap method is computationally expensive, we chose a moderate $M_4 = 20$ in this paper.

3.G.1.4 Check for model bias

We tested our model for bias with an observed versus predicted regression as described by Haefner (2005). If the model is accurate, predictions should be close to the observed data. Hence, all data should be close to a line with slope 1 and intercept 0, when observed data are plotted against predictions. The test described by Haefner (2005) checks the null hypothesis "slope = 1 and intercept = 0". If the model is unbiased, the resulting p-value should be high so that the null hypothesis cannot be rejected.

The test requires that all predictions follow normal distributions with similar variances. Therefore, a transformation step was required to make the test applicable to our model. To obtain normally distributed predictions, we considered sums of identically and independently distributed (i.i.d.) random variables. These sums are approximately normally distributed according to the central limit theorem. We generated the sets of i.i.d. random variables by considering the homogenized count data obtained as described in section 3.G.1.1. We considered the total number of boaters observed in each shift. Then we proceeded as follows:

1. Using our fitted model and knowing the sample sizes at each survey location, we computed the predicted standard deviation of the count data for each survey location.

2. We normalized the count data so that they had a predicted standard deviation of 1.

3. We normalized our model predictions accordingly.

4. We applied the method by Haefner (2005) to the normalized observations and predictions and computed the p-value.

We applied the method described above to a validation data set distinct from the data set used to fit the model. To generate the validation data set, we randomly selected 30% of all survey shifts. The rest of the data were used to fit the model.

3.G.1.5 Accuracy of the predicted mean boater flow

In this section, we describe the method we applied to assess the accuracy of our model's predictions. A commonly used measure for model accuracy is the coefficient of determination R^2. However, R^2 is not applicable in our case, because we assume that the variance of our count data increases proportional to the respective mean values. That is, R^2 would put higher emphasis on large count data than desired. Furthermore, R^2 would provide a measure for the "absolute" error, while the relative error is often of higher interest to managers. Considering the relative error, in turn, is hard if the data are dominated by low counts.

Since R^2 is not an appropriate measure of accuracy for our model, we conducted a graphical comparison of predicted and observed count values. We determined predicted and observed count values based on our survey set up. That is, our predictions took into account where and when we conducted surveys. Then we plotted the observed count values against predicted mean values. Since the model is stochastic, we expect the observed values to deviate from the predictions. Nonetheless, the predicted-observed pairs can be expected to be close to a line with slope 1 and intercept 0 if the model is accurate.

To identify strengths and weaknesses of our model, we conducted the analyses from four perspectives:

1. To assess the model's ability to predict the flow between individual origin-destination pairs, we plotted for each donor-recipient pair the number of observed and predicted boaters.

2. To assess the model's ability to determine the repulsiveness of donor jurisdictions, we plotted observed and predicted boaters for each individual donor jurisdiction.

3. To assess the model's ability to estimate the attractiveness of recipient lakes, we plotted observed and predicted boaters for each recipient lake.

4. To assess the model's ability to predict the boater flow along roads, we plotted observed and predicted boaters for each survey location.

Similar to the check for model bias, we applied the check for model accuracy to a validation data set distinct from the data set used to fit the model. We tested model accuracy based on the same validation set used to check the model for an overall bias.

3.G.2 Results

3.G.2.1 Shape of the temporal traffic pattern

To check whether the von Mises distribution is appropriate to model the temporal traffic pattern, we compared the fitted von Mises distribution to step function distributions fitted to the data. In Figure A3.2, it is visible that the distributions resemble each other in shape. Besides a graphical comparison, we also compared the distributions based on the model selection criterion AIC. The AIC values of the distributions were close, though the best step function model ($n = 10$) was slightly lower than the von Mises distribution ($\Delta\text{AIC} = 2.9$).

3.G.2.2 Distribution of count data

We tested whether our count data came from a negative binomial distribution. We obtained a p-value of 0.27 with standard deviation 0.05. To test the power of our approach, we also applied the nested parametric bootstrap method to test whether the count data are Poisson distributed. This hypothesis resulted in a p-value of 0.

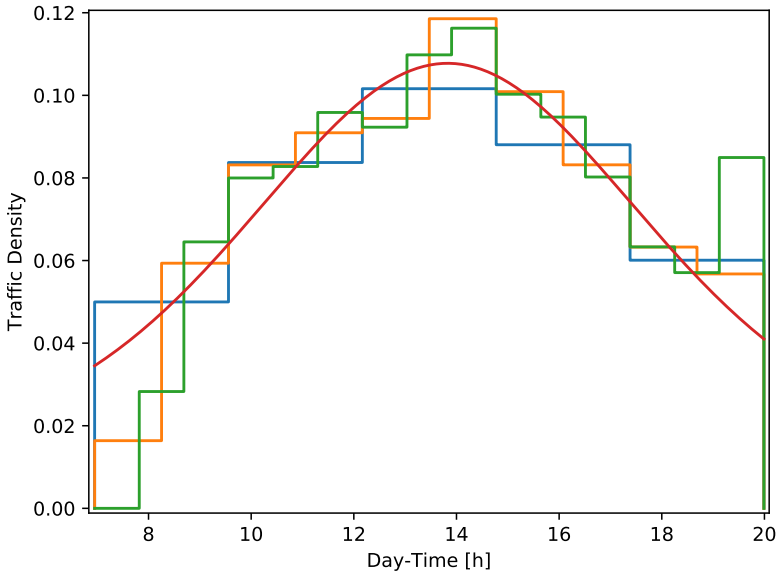


Figure A3.2: Comparison of different step-function distributions with the von Mises distribution. The curves depict the probability density functions of the best-fit step distributions with $n = 5$ intervals (blue), $n = 10$ intervals (orange), $n = 15$ intervals (green), and the von Mises distribution (red). To ease comparison, all curves were normalized to be probability distributions on the time interval 7AM till 8PM, for which we have count data. It is visible that the shapes of the step functions resemble the shape of the von Mises distribution.

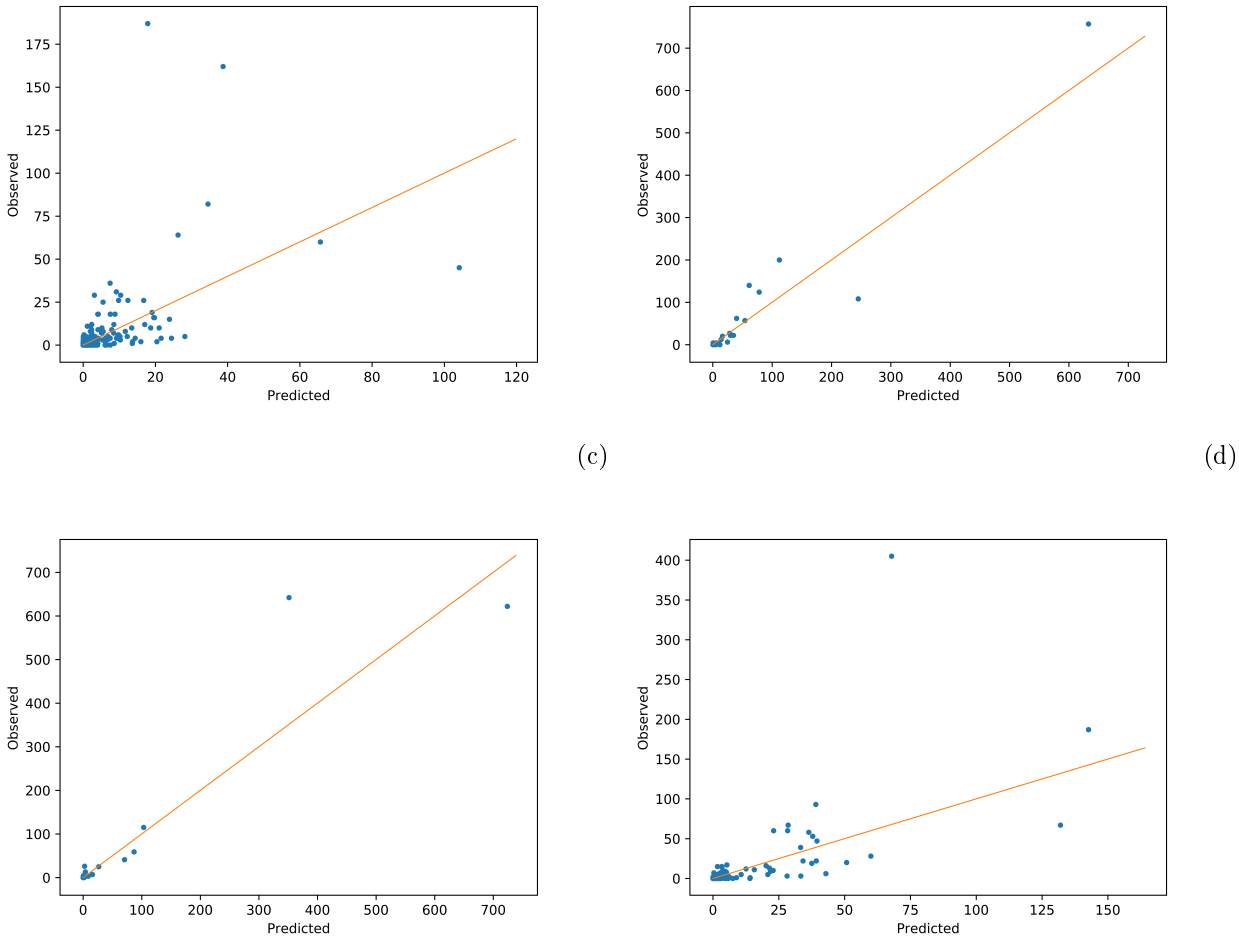
3.G.2.3 Check for model bias

The check for model bias resulted in a p-value of 0.22.

3.G.2.4 Accuracy of the predicted mean boater flow

The observed versus predicted plots that we generated to test the accuracy of our model are displayed in Figure A3.3. It is visible that our model has difficulties to predict the number of travelling boaters for separate origin-destination combinations (Figure A3.3a). There are several jurisdiction-lake pairs for which the observed value is far from the mean of the estimated distribution. The same applies to the plot displaying the model's ability to predict the total inflow to lakes (Figure A3.3d). However, the predicted and observed values match relatively well for the total outflow of jurisdictions (Figure A3.3c) and the flow through the survey locations (Figure A3.3b).

(a) (b)



(c) (d)

Figure A3.3: Observed versus predicted count values. The blue dots depict the the predicted mean and the observed count value of boaters for (A3.3a) each jurisdiction-lake pair, (A3.3b) each survey location, (A3.3c) each source jurisdiction, and (A3.3d) each recipient lake. If the model were perfect, all points would be close to the solid line, at which predicted mean and observed value are equal. The predicted stadard deviation is twice the square root of the respective predicted mean.

3.G.3 Discussion

3.G.3.1 Methods

Before discussing the main validation results, we discuss the model validation methods that we applied.

We used a graphical comparison method to check whether the von Mises distribution is appropriate to model the temporal variations of traffic. Of course, a more rigorous statistical test, e.g. the Anderson-Darling test, would have been possible, too. However, since such tests require identically distributed samples in general, we would not have been able to use all available data for these tests. Furthermore, statistical tests may be suitable to show that a hypothesis is wrong, but other methods may be more appropriate to confirm a null hypothesis. The observation that distributions without pre-imposed shape mimic the von Mises distribution is a strong hint suggesting that the von Mises distribution is appropriate to model temporal traffic variations.

Our nested parametric bootstrap method for testing whether the count data come from a negative binomial distribution is computationally expensive and can lead to imprecise results. However, in simulations (not shown here) the method proved to be powerful in discerning negative binomially distributed data from data coming from other distributions. This observation goes in line with the low p-value with which the nested parametric bootstrap showed that the count data did not come from a Poisson distribution. Though the computational constraints make it impossible to generate a large number of bootstrap samples, the error of the method was sufficiently small to allow well-informed inference.

The observed versus predicted analysis that we used to check for model bias is a suitable method to confirm that the model is implemented correctly. However, though the method is able to identify an overall bias in our predictions, the method would be unable to identify biases in subsets of our data. For example, if our model would underestimate the traffic to attractive lakes and overestimate the traffic to unattractive lakes, the aggregate predictions

would not show a bias. Therefore, the method cannot be used to measure the accuracy of our predictions.

The graphical observed versus predicted analysis we conducted to assess the accuracy of our model is a suitable tool to measure model performance, as it is easy to check which parts of the model fit the data well and where inaccuracies result from. As an alternative to a graphical analysis, a nested bootstrap method could be applied to check whether a statistic applied to the observed data would be likely to return the observed value if the model is correct. However, given the apparent inaccuracies, which far exceed expected standard deviations, it is not necessary to apply additional tests to confirm that the model is inaccurate. Therefore, we abstained from implementing this computationally expensive validation method.

3.G.3.2 Results

We have checked two main hypotheses our model is based on and validated the accuracy of the model's predictions. Overall, our test results indicate that the model assumptions are appropriate. However, the model's predictions turned out to suffer from inaccuracies.

For the temporal traffic pattern model, the fitted step function distributions resembled the von Mises distribution and resulted in only slightly better AIC values. This justifies the choice of the von Mises distribution to model the temporal traffic pattern, also considering that (1) the von Mises distribution has a lower risk of being overfitted to the data, and (2) the von Mises distribution provides reasonable estimates for night-time traffic, for which we have no data. Hence, it is appropriate to model the temporal traffic pattern with the von Mises distribution.

For the distribution of the count data, we obtained a relatively high p-value for the null hypothesis that our data are negative binomially distributed. Even though this does not prove that the data are negative binomially distributed, this result does not allow us to conclude the opposite. Since a distribution test with the null hypothesis that the data are Poisson distributed resulted in a very small p-value, our test appears to be sufficiently powerful to

reject wrong hypotheses. This supports the negative binomial hypothesis further. Hence, the negative binomial distribution seems appropriate to model our count data.

Our test for an overall model bias resulted in a moderate p-value. Hence, the null hypothesis that the model yields unbiased results cannot be rejected. Therefore, we have no reason to believe that the model predictions are subject to an overall bias.

Our comparison of predicted and observed values has shown that our model suffers from inaccuracies. As we conducted separate checks for the model's ability to predict the outflow of donor jurisdictions and the inflow to recipient lakes, we can make informed guesses about which model component is responsible for the errors. Both the temporal pattern model and the route choice model are likely to affect all predictions similarly strongly. If these model components were the main cause for the inaccuracies, we would see the same level of inaccuracy on all predicted versus observed plots. However, we observed that our model's predictions of the outflow from jurisdictions were much more accurate that the predictions of the inflow to jurisdiction lakes (compare Figures A3.3c and A3.3d). The outlier in Figure A3.3c corresponds to boaters coming from the middle part of Alberta, a neighbouring province of BC, and may be partially caused by difficulties to determine the origin of boaters on a sub-provincial scale. Therefore, it is likely that our model's inaccuracies result from its inability to precisely estimate the attractiveness of lakes rather than from other model components.

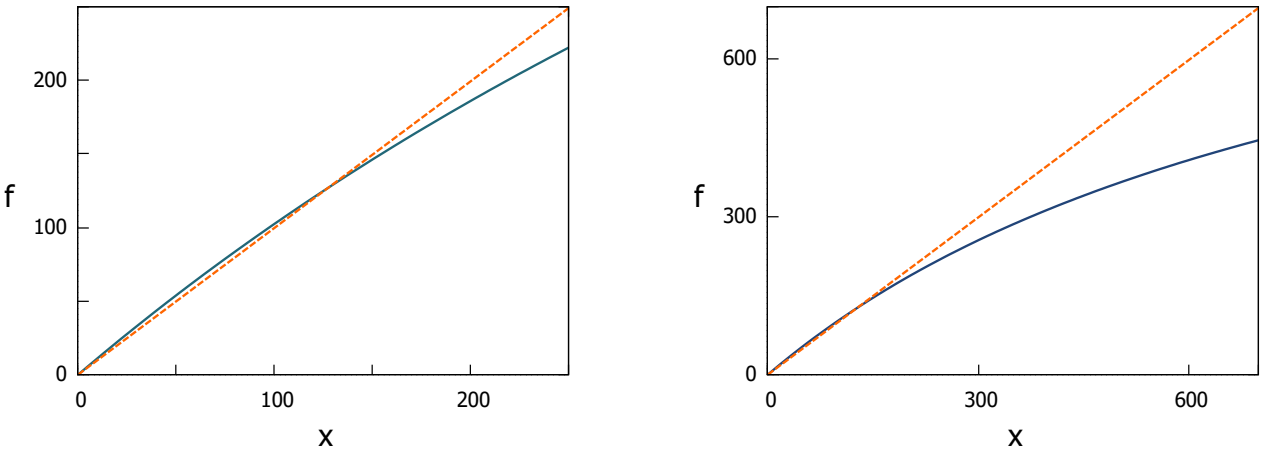
We can conclude from the model validation results above that a more accurate model would require a more sophisticated submodel for lake attractiveness. Improving the other model components may also enhance the model accuracy but presumably not to the same extent as an improved gravity model. A more sophisticated model for lake attractiveness, however, would likely also require more covariates to distinguish between attractive unattractive lakes. This is a constraint that all models for agent traffic would face. Therefore, the limited accuracy of our model does not generally outweigh the methodological advancements of this study.

3.H Identifiability of the parameters β_{lpop} and lpop_0

The confidence intervals for the parameters β_{lpop} and lpop_0 given in Table A3.3 are very large. That is, the correct values of these parameters are not estimable with the data that we used to fit the model. Often, such estimability issued decrease the credibility of inference and predictions drawn from a model. However, we argue that though the parameter values appear to be not estimable, our model and resulting predictions are reliable.

In Figure A3.4, we have plotted the contribution $f(x) := \beta_{\text{lpop}} \left(\frac{x}{x+\text{lpop}_0} \right)$ of the covariate "population in a 5 km range of a lake" (here denoted x) to the lake attractiveness for two extreme parameter choices. It is visible that the contribution curves differ by no more than factor 1.5. The difference is maximal for lakes with a high surrounding population. Note that only two of the considered lakes have a surrounding population exceeding $250,000$. These lakes have a small area and therefore do not attract many boaters. Hence, these lakes do not contribute to the results significantly. For lakes with surrounding population counts below $250,000$, in turn, the estimates differ by no more than 15%. For the middle section of Okanagan Lake, the only lake section that has both a high surrounding population count and a large size, the contribution of the surrounding population count to the lake attractiveness differs by less than 5% between the models.

Note that the large range of permissible parameter values suggest that even a model without the parameter lpop_0 could fit the data well. Indeed, the AIC difference between the models with and without this parameter is less than 2, so that both models can be considered well-fitting. Nonetheless, the model with the additional parameter has the minimal AIC value and was thus chosen.



(a) (b)

Figure A3.4: Contribution of the covariate "population in a 5 km range of a lake" (in thousand; denoted x) to the lake attractiveness for two extreme parameter choices. The two functions differ moderately for large population counts. The left panel shows a subsection of the right panel. Parameters: solid blue: $\beta_{\mathrm{lpop}} = 1011$, $\mathrm{lpop}_0 = 887$; dashed orange: $\mathrm{lpop}_0 = 1.182\mathrm{e}_{10}$, $\alpha_{\mathrm{lpop}} = 1.186\mathrm{e}_{10}$.

Chapter 4

Managing aquatic invasions: optimal locations and operating times for watercraft inspection stations

4.1 Introduction

Human traffic and trade are major vectors for invasive species (Lockwood et al., 2013). Due to the significant ecological and economic damages invasive species cause (Pimentel et al., 2005), government regulations restrict the import of certain goods and require treatment of potentially infested freight and carriers (Shine et al., 2010; Johnson et al., 2017; Turbelin et al., 2017). While such regulations may be enforced comparatively easily at ports, air ports, and border crossings, control of inland traffic is more difficult, as a vast number of routes need to be monitored. This applies for example to the spread of zebra and quagga mussels (*Dreissena spp.*) and other aquatic invasive species (AIS), which often spread with watercraft and equipment transported from invaded to uninvaded waterbodies (Johnson et al., 2001). Zebra and quagga mussels are invasive in North America and have negative effects on native species and ecosystems, water quality, tourism, and infrastructure (Rosaen et al., 2012; Karatayev et al., 2015b).

To counteract the spread of these AIS, watercraft inspection stations are set up on roads, where transported watercraft are inspected for AIS and decontaminated if at risk for carrying AIS (Mangin, 2011; Alberta Environment and Parks Fish and Wildlife Policy, 2015; Inter-

Ministry Invasive Species Working Group, 2015). However, since budgets for inspections are limited, not all pathways can be monitored around the clock, and managers need to prioritize certain locations and day times. Though several theoretical studies provide managers with helpful guidelines for their work (Leung et al., 2002; Potapov and Lewis, 2008; Potapov et al., 2008; Vander Zanden and Olden, 2008; Finnoff et al., 2010; Hyytiäinen et al., 2013), more specific results are needed in practice to determine the locations and times where and when control is most effective. To date it has been difficult to tackle these questions rigorously, as comprehensive models for road traffic of potential vectors were missing. Therefore, AIS managers have relied on past watercraft inspection data, shared experience between jurisdictions, and iterative improvements of control policies. The modelling advances made in chapter 3, however, now permit the application of quantitative methods to optimize control measures in road networks and to evaluate their effectiveness. This will be the subject of this paper.

Our goal will be to minimize the number of boaters reaching uninvaded waterbodies without being inspected for AIS. Thereby, we will assume that a fixed budget is available for AIS control. This problem setup differs from scenarios considered in other studies on optimal control of invasive species (Hastings et al., 2006; Potapov and Lewis, 2008; Potapov et al., 2008; Finnoff et al., 2010; Epanchin-Niell and Wilen, 2012), where budget allocation over time is optimized along with the control actions. However, to optimize the budget, invasions need to be assigned "cost labels". This is an often difficult and politically sensitive task. Furthermore, the budget available for AIS control may be subject to political and social influences and determined on a different decision hierarchy than the management actions. Therefore, AIS managers may seek to spend a fixed yearly budget optimally rather than to determine the theoretically best control budget. The presence of fixed budget constraints also reduces the need to consider the invasion as a dynamic process.

Identifying the locations where a maximal number of boaters could be screened for AIS is similar to the problem of finding optimal locations for road-side infrastructure (Trullols et al., 2010). A well-known technique to solve such problems is linear integer programming

(Conforti et al., 2014). The idea is to model the optimization problem with functions linear in the decision variables. Though solving linear integer programs is a computationally difficult task in general, good approximate solutions can often be determined, and a variety of software tools are available to compute solutions. Therefore, linear integer programming has also been used in the context of invasive species management (Epanchin-Niell and Wilen, 2012; Kıbış and Büyüktahtakın, 2017).

A crucial step in linear integer optimization is to find a problem formulation that facilitates good approximations (Ageev and Sviridenko, 2004). In this paper, we provide such a formulation to optimize locations and operating times of watercraft inspection stations. This problem differs from comparable resource allocation problems (Surkov et al., 2008; Trullols et al., 2010), as we need to account for the temporal variations in traffic. These variations are key when we consider the trade-off between operating few inspection stations intensely, e.g. around the clock, and distributing resources over many locations operated at peak traffic times only.

We demonstrate the potential of our approach by applying it to optimize watercraft inspection policies for the Canadian province British Columbia (BC). We show how uncertainty, different cost constraints, and additional propagule sources impact the optimal policy. Thereby, we identify control principles applicable beyond the considered scenario.

This paper is structured as follows: we start by introducing model components required to optimize watercraft inspection station operation. Then, we show how the considered optimization task can be formulated as linear integer problem. Thereby, we focus first solely on inspection station placement before we introduce the full problem, in which also operating times of inspection stations are optimized. After this general description of our approach, we apply the method to AIS management in BC and present results under different scenarios. Lastly, we discuss our results and the limitations of our approach and draw general conclusions on AIS management.

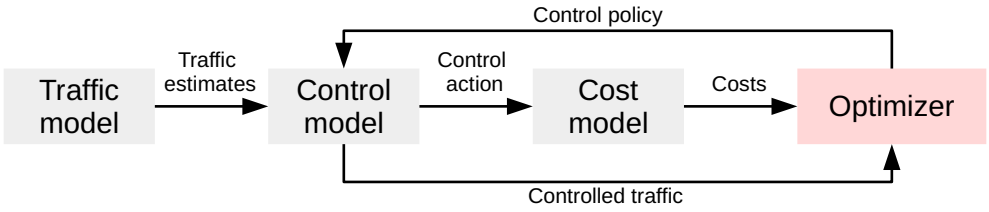


Figure 4.1: Components of our approach. The control model determines how the traffic estimated by the traffic model changes under a given control policy. The cost model yields the costs for control actions. The optimizer maximizes the controlled traffic subject to a cost constraint.

4.2 Method

4.2.1 Model

Our goal is to identify how limited resources can be allocated most effectively to minimize the number of boaters arriving at uninvaded waterbodies without being inspected for AIS. We assume that two aspects of the control strategy can be changed: the locations and operating times of watercraft inspection stations. As traffic typically follows cyclic patterns, we consider one such cycle as the time horizon for the control optimization.

To find an optimal inspection policy, we need three models (see Figure 4.1): (1) a model for boater traffic, (2) a model for control, and (3) a model for control costs. The traffic model gives us estimates of when, where, and along which routes boaters travel. The control model shows us when and where inspections could be conducted and how effective they are. Lastly, the cost model measures the costs for inspections. The information from the three models serve as input for a control optimizer that determines a good – or, if possible, the best – watercraft inspection strategy. Below, we describe each of the models in greater detail before we introduce suitable optimization routines in the next section.

4.2.1.1 Traffic model

The traffic model provides estimates of when, where, and along which routes boaters drive. Knowledge about routes is key to understanding whether boaters passing one control location

have already been inspected at another location. For each considered route, the traffic model provides us with a traffic estimate. In this study, we use the hybrid gravity and route choice model from chapter 3 to estimate the traffic. The model includes components accounting for boaters' travel incentive, their route choice, the timing of traffic, and boaters' compliance with inspections.

In practice it is rarely feasible to consider all routes that boaters could possibly take, and we need to focus on some set of "reasonable" routes (Bovy, 2009; see also chapter 2). As a consequence, there may be some agents travelling along unexpected routes. When boaters travelling along such routes arrive at inspection locations, we do not know whether their watercraft have been inspected earlier. This makes it difficult to optimize inspection strategies. Nonetheless, we may want to account for these boaters by introducing a "noise" term to our model. To that end, we assume that a fraction of the travelling boaters could be observed at any inspection location with a small probability (see chapter 3).

As road traffic is rarely uniform over time, we furthermore need a submodel predicting how traffic varies with time. While it may be comparatively easy to determine the temporal distribution of traffic at a specific location, it can be difficult to identify the temporal relationship between traffic at two locations on the same route. For example, agents passing one location in the morning may not be able to reach another location before the afternoon. Modelling such relationships is particularly difficult for locations far from each other, as boaters may have different travel speeds. We therefore apply a simplification and assume all boaters travelling along a route have the same speed.

4.2.1.2 Control model

We assume that there is a specific set of locations where watercraft inspections could be conducted. For example, these locations could be pullouts large enough to provide a safe environment for inspections. We suppose that compliant boaters stop for an inspection whenever they pass an operated inspection station. Conversely, uncompliant boaters are

assumed to bypass any inspection station on their route. Consequently, we seek to maximize the number of boaters that pass at least one operated watercraft inspection.

As with the inspection locations, we assume that there are specific time intervals when inspection can be conducted. The admissible time intervals may be determined by safety concerns or practical considerations and can be location dependent. As staff cannot move between distant inspection locations easily, and the working hours of inspection staff are subject to legal and practical constraints, we may furthermore assume that every inspection station can be operated in shifts of given lengths only.

4.2.1.3 Cost model

Inspection costs may be split in two classes: infrastructure costs that apply once for each chosen inspection location, and operational costs, which depend on when and for how long an inspection station is operated. The operational costs may also account for ongoing equipment maintenance costs and training of staff. The control costs may be location and time dependent. For example, it may be expensive to conduct inspections at remote locations if staff must travel long distances to their work place. Furthermore, some locations will require significantly more infrastructure costs (e.g. lighting and washrooms) in order to operate overnight shifts. In addition, wages are often higher in overnight shifts.

4.2.2 Optimizing control locations

With the submodels from the previous section at hand, we can proceed optimizing the inspection strategy. Optimizing both locations and operating times of watercraft inspection stations at the same time is conceptually and computationally challenging. To ease the introduction of our approach, we first consider a scenario in which inspection stations are operational around the clock. In this case, we can ignore the temporal variations of traffic and focus on choosing optimal control locations (cf. Trullols et al., 2010).

In this section, we show how the corresponding optimization problem can be formulated as a linear integer problem. To that end, we let L be the set of all admissible inspection locations and introduce for each location $l \in L$ a binary variable x_l that assumes the value 1 if and only if an inspection station is set up at l. Let R be the set of potential routes that boaters may choose, n_r the expected number of complying boaters travelling along route $r \in R$, and $L_r \subseteq L$ the set of locations where the boaters travelling on route r could be inspected.

As noted earlier, one inspection station suffices to control all complying boaters driving along a route r. Consequently, boaters travelling on route r will be controlled if and only if

$$\sum_{l \in L_r} x_l \quad \geq \quad 1. \tag{4.1}$$

Otherwise, the left hand side of equation (4.1) will be 0. Therefore, we can express the total number of inspected boaters by

$$F_{\text{loc}}(\boldsymbol{x}) \quad := \quad \sum_{r \in R} \min \left\{ 1, \sum_{l \in L_r} x_l \right\} n_r. \tag{4.2}$$

To formulate the cost constraint, let c_l be the cost for operating control location $l \in L$ and B the available budget. As we assume that all inspection stations are operated for the same time, we do not need to distinguish between infrastructure and operation costs. Hence, we can write the cost constraint as

$$\sum_{l \in L} c_l x_l \quad \leq \quad B. \tag{4.3}$$

The optimal placement policy can be identified by maximizing $F_{\text{loc}}(\boldsymbol{x})$ over all $\boldsymbol{x} \in \{0,1\}^{|L|}$ subject to constraint (4.3). Though F_{loc} contains a "minimum" function, F_{loc} can be easily transformed to a linear function by introducing further variables and linear inequality constraints (see e.g. Ageev and Sviridenko, 1999). Since the left hand side of the cost constraint

(4.3) is linear in $\boldsymbol{x}$ as well, and $\boldsymbol{x}$ is constrained to be a vector of integers, the considered optimization problem is a linear integer problem. This can be solved with a suitable general linear integer programming solver or a specifically tailored rounding algorithm (Ageev and Sviridenko, 2004). We discuss possible optimization routines in section 4.2.5.

4.2.3 Optimizing control locations and timing

After focusing on inspection station placement, we now extend our approach to permit free choice of inspection station operating times. In this extended scenario, we need to balance the trade-off between operating few highly frequented inspection stations around the clock and distributing efforts over many locations operated at peak traffic times only. This trade-off makes combined optimization of location choice and timing more challenging than separate optimization of location choice and timing (cf. Epanchin-Niell and Wilen, 2012).

While location choice is a discrete optimization problem – each potential inspection location is either chosen or not – optimization of operating times is a continuous problem, since inspections could be started at any time. To exploit the toolset of discrete optimization anyway, we simplify our problem by discretizing time. That is, we split the boater traffic corresponding to boaters' departure times and consider only discrete sets of admissible inspection shifts.

Let T be a set of disjunct time intervals covering the complete time span of interest. We write n_{rt} for the expected number of boaters who travel on route $r \in R$, start their journey in time interval $t \in T$, and are willing to comply with inspections. Let furthermore S_l be the set of admissible inspection shifts for location $l \in L$. Each shift corresponds to a time interval in which the inspection station is operated. Since the shift lengths are given, the set S_l can be fully characterized by the shifts' start times.

As we assume that all boaters travelling along a route have the same speed, we can determine the set $S_{lrt} \subseteq S_l$ of control shifts during which boaters who started their journey in time interval $t \in T$ arrive at location $l \in L$ via route $r \in R$. Under reasonable error

allowance, it is usually possible to construct the sets S_{lrt} in a way that each shift covers the departure time intervals either completely or not at all, respectively. This setup prevents issues arising if some intervals overlap only partially.

To formulate our optimization problem as linear integer problem, we describe the control policy again with binary variables $x_{ls} \in \{0,1\}$. Here, x_{ls} is 1 if and only if an inspection station at location $l \in L$ is operated in shift $s \in S_{lrs}$. Agents travelling on route $r \in R$ who departed in time interval $t \in T$ are controlled if and only if

$$\sum_{l \in L_r} \sum_{s \in S_{lrt}} x_{ls} \quad \geq \quad 1. \tag{4.4}$$

Consequently, the total controlled agent flow is given by

$$F_{\text{full}}(\boldsymbol{x}) \quad := \quad \sum_{r \in R} \sum_{t \in T} \min \left\{ 1, \sum_{l \in L_r} \sum_{s \in S_{lrt}} x_{ls} \right\} n_{rt}. \tag{4.5}$$

To derive the cost constraint, recall that we distinguish between infrastructure costs c_l^{loc} for using location l and operating costs c_{ls}^{shift} payable per control shift s conducted at l. Consequently, the total costs for control at l are given by

$$\sum_{s \in S_l} c_{ls}^{\text{shift}} x_{ls} + c_l^{\text{loc}} \max_{r \in R,\, t \in T} \left(\sum_{s \in S_{lrt}} x_{ls} \right), \tag{4.6}$$

and the cost constraint reads

$$\sum_{l \in L} \left(\sum_{s \in S_l} c_{ls}^{\text{shift}} x_{ls} + c_l^{\text{loc}} \max_{r \in R,\, t \in T} \left(\sum_{s \in S_{lrt}} x_{ls} \right) \right) \quad \leq \quad B. \tag{4.7}$$

As in the previous section, B denotes the available budget. Optimizing F_{full} subject to (4.7) is a linear integer problem, since the "minimum" term in (4.5) and the "maximum" terms in (4.7) can be replaced by introducing correspondingly constrained variables.

4.2.4 Noise

Even if the traffic model accounts for most routes boaters use, some boaters may travel along unexpected routes. It is difficult to optimize inspection station operation with regards to these boaters, as we do not know which inspection stations cover the same routes. Nonetheless, it can be desirable to account for noise, since the level of uncertainty may affect the optimal inspection policy.

In the absence of a mechanistic model for traffic noise, we may assume that boaters who are travelling on unexpected routes are passing any inspection location with a small probability η_o, whereby they choose the passing time randomly. Under this assumption, the expected number of inspected boaters travelling along unknown routes is given by

$$F_{\text{noise}} = \left(1 - \prod_{l \in L} \left(1 - \eta_o \sum_{s \in S_l} x_{ls} \tau_{sl}\right)\right) n_{\text{noise}}. \tag{4.8}$$

Here, n_{noise} denotes the expected number of boaters travelling on unknown routes.

As F_{noise} is not a convex function, adding this noise term to the objective function would make optimization difficult. However, as η_0 is typically small, equation (4.8) is well approximated by

$$\hat{F}_{\text{noise}} = \eta_o n_{\text{noise}} \sum_{l \in L} \sum_{s \in S_l} x_{ls} \tau_{sl}, \tag{4.9}$$

which is linear and can thus be easily added to the linear integer problem. This approximation is most precise if $x_{ls} = 0$ for most l and s. If the budget is high enough to operate many inspection stations for long times, the noise may be overestimated. However, since n_{noise} is typically small compared to the total boater traffic, inaccuracies in the noise model are unlikely to alter the overall optimization results significantly.

4.2.5 Solving the optimization problems

Having derived the problem formulation in the previous sections, we now proceed by discussing suitable solution methods. The inspection station placement problem described in section 4.2.2 is equivalent to the budgeted maximum coverage problem (Khuller et al., 1999), also called maximum coverage problem with knapsack constraint (Ageev and Sviridenko, 2004). This problem is well studied in computing science, and it has been shown that finding a solution better than factor $(1 - e^{-1})$ of the optimum is an NP-hard, often infeasibly difficult, problem (Feige, 1998). This result applies also to the extended problem introduced in section 4.2.3, as it is more general than the placement problem. Though these theoretical results show that scenarios exist in which the problems considered in this paper cannot be solved exactly within reasonable time, good approximate or even optimal solutions can often be obtained in practical applications.

When seeking a good solution, we can exploit that the linear integer formulation of our problem helps us to obtain upper and lower bounds to solutions efficiently. Consider a slightly changed optimization problem in which the management variables $\boldsymbol{x}$ are not constrained to be integers but drawn from the continuous domain $[0, 1]^N$. Here, N is the dimension of the problem. In this case, the problems can be solved with linear programming techniques within seconds even if N is large. Clearly, the integer domain $\{0, 1\}^N$ is a subset of the continuous domain $[0, 1]^N$. Therefore, the solution to the problem with relaxed integer constraint is an upper bound to the desired integer solution.

Often it is possible to obtain good integer solutions by rounding the solution to the continuous problem. Ageev and Sviridenko (2004) present an algorithm that always achieves the approximation bound $(1 - e^{-1})$ for the inspection station placement problem, in which operating times are fixed. Nonetheless, general solvers with possibly poorer worst-case performance may yield better solutions in "benign" cases. A number of generally applicable methods exist (Conforti et al., 2014). In this study, we use branch and bound methods, in

which the distance between upper and lower bounds on the optimal objective are found by solving continuously relaxed subproblems with some constrained variables.

A challenge that general solvers face is to find a good initial feasible solution that they can improve on. For the pure inspection station placement problem, we could apply the rounding algorithm by Ageev and Sviridenko (2004), which would also guarantee us the best approximation bound. However, for the joint optimization of both placement and operating times of watercraft inspection station, we are not aware of any algorithm with such a guarantee. We therefore propose a "greedy" rounding algorithm to obtain good initial solutions. The idea is to solve the relaxed linear programming problem and to determine the largest non-integer decision variable that can be rounded up without violating the cost constraint. We applied this procedure with some improvements described in Appendix 4.A. In applications, we consistently obtained solutions better than 80% of the optimum with this approach.

4.3 Application

To show the potential of our approach, we applied it to optimize watercraft inspections in the Canadian province British Columbia (BC). Below we provide an overview of the scenario-specific submodels we used. Furthermore, we briefly describe our implementation of the presented approach.

4.3.1 Scenario-specific submodels

4.3.1.1 Traffic model

To model boater traffic, we used the hierarchical gravity and route choice model for boater traffic developed in chapter 3. The model was fitted to data collected at British Columbian watercraft inspection stations in the years 2015 and 2016. At the time this study was conducted, dreissenid mussels were not known to be established anywhere in BC. As sources of potentially infested boaters, we therefore considered the Canadian provinces and American

states that (1) were known to be invaded by dreissenid mussels or (2) had connected waterway to an infested jurisdiction and no coordinated mussel detection program in place at the time the data were collected. As sinks we identified 5981 potentially boater accessible lakes in BC.

To estimate the boater traffic between an origin and destination, the model considered characteristics of the donor jurisdiction, the recipient lake, and the distance between the two. Major sources of high-risk boaters were characterized by high population counts. Furthermore, Canadian provinces were found to have higher boater traffic to BC than American states. Attractiveness of destination lakes increased with their surface area, the population counts of surrounding towns and cities, and the availability of close-by touristic facilities, such as campgrounds. Lastly, the boater flow was estimated to decay in cubic order of the distance between an origin and a destination. For a detailed description of the model along with precise parameter estimates, refer to chapter 3.

To identify potential boater pathways, we computed locally optimal routes (see chapter 2) between the considered origins and destinations. These routes arise if routing decisions on local scales are rational and based on simple criteria (here: minimizing travel time) whereas unknown factors may affect routing decisions on larger scales. Consequently, the model accounts for routes arising from a multitude of mechanisms. The attractiveness of the routes was computed based on their length measured in travel time. Again, a more in-depth description of the model and the fitted parameter values can be found in chapter 3.

The fraction of boaters travelling on routes not covered by our traffic model was estimated as 4.9%. However, this number is not estimable from survey data obtained at watercraft inspection stations, because it is negatively correlated with the parameter η_o (section 4.2.4), denoting the probability to observe a boater travelling on an unknown route at an arbitrary inspection location. Therefore, we introduced an additional model assumption bounding the noise term below 5% (see chapter 3). Note that due to the dependency of η_o on the noise level, the estimability issue has little effect on the noise level observed at watercraft inspection

stations and thus on inspection policy. Based on a noise level of 4.9%, η_o was estimated as 0.06 (chapter 3).

The temporal distribution of traffic was modelled with a von Mises distribution. This is a unimodal circular distribution often used in models (Lee, 2010). The temporal pattern was assumed to have a period of one day. The traffic high was estimated to be at 2 PM, whereby the estimated peak traffic was 15 times higher than the estimated traffic volume at night. As traffic data were available for specific inspection locations only, we assume that the temporal traffic distribution is uniform over all locations.

Assuming an equal temporal traffic distribution for all potential inspection locations makes it difficult to account for the time boaters need to travel between two sites. This, is a model limitation but not of major concern in the considered scenario of boater traffic to BC. First, note that we seek locations that are *not* on the same pathway. If boaters do not pass multiple operated inspection locations, we are safe to neglect the travel time between sites. Furthermore, we can exploit that the considered boater origins are located outside of the province and boaters drive, with minor exceptions, along highways in one particular direction. Consequently, the temporal traffic distribution of close-by locations on such a highway would be equal up to a shifting term, and the optimized inspection times could be adjusted accordingly.

4.3.1.2 Control model

As described in section 4.2.1.2, we assume that every complying boater passing an operated inspection location is inspected for invasive mussels. The compliance rate across all inspection stations was estimated to be 80% (chapter 3). To find potentially suitable locations for inspections, we identified pullouts across BC. We reduced the number of possible options by disregarding some pullouts in close proximity to others. In total, we considered 249 location candidates.

Due to the large number of location candidates, we did not conduct a detailed evaluation of the operational suitability of all considered locations (e.g. pullout size, signage, and safety). Instead, we consulted with the BC Invasive Mussel Defence Program to gauge the general suitability of the locations suggested by the optimizer. If a suggested location seemed unsuitable, we removed it from the candidate set and repeated the optimization procedure. Despite this superficial suitability check, a more detailed analysis would be necessary to account for all potential practical constraints. These must be considered independent of the model before an inspection station can be placed.

For each location, we assumed that 8 h long inspection shifts could be started at each full hour of the day. Note that "shift" here refers to the time inspections are conducted and does not include time required for staff to access or set up an inspection station. The work time of staff will therefore be longer in practice. The assumed length of the inspection shifts aligns with average operation patterns of watercraft inspection stations in BC and divides each day in three equally sized shifts, which simplifies the model. Though the effective operation time (limited by access time of staff) is lower at remote locations, our time model provides a good first approximation.

4.3.1.3 Cost model

We determined the inspection costs based on correspondence with the BC Invasive Mussel Defence Program. The considered optimization problem is often easier to solve if costs are rounded to well aligned cost units. Therefore, we set the infrastructure costs for setting up an inspection station as our base cost unit. The costs per conducted inspection shift are then 3.5 units during day-time hours and 5.5 units between 9 PM and 5 AM. These costs include salary, training, and equipment for inspection staff. In 2017, the BC Invasive Mussel Defence Program was operating on a budget of approximately 80 cost units.

As in-depth location-specific cost analysis would have been difficult, we assumed that the inspection costs are equal for all considered locations. Note, however, that site specific

costs can vary significantly and may be a limitation when assessing a location for overnight operations.

4.3.2 Implementation

As we considered about $300,000$ origin-destination pairs connected by 6.7 routes on average, considering all boater pathways individually would be difficult. Therefore, we merged traffic of boaters passing the same sets of potential inspection locations. As a result, the number of distinct boater flows reduced to 2026.

We determined the optimal inspection locations and operating times under different budget scenarios. This allowed us to determine the budget required to minimize the fraction of uninspected high-risk boaters to a desired level. We also varied the model's noise term to test how inspection strategies change under increased uncertainty. To see how new infestations in close-by jurisdictions change the inspection policy, we furthermore considered a scenario in which the American states Idaho, Wyoming, and Oregon are invaded.

We implemented the model in the high-level programming language Python version 3.7. To formulate the linear integer problem, we used the modelling software CVXPY version 1.0.25 with added support for initial guesses. To solve the linear integer problem, we used the commercial solver MOSEK. We computed initial guesses with the greedy rounding procedure described in section 4.2.5. We let the solver terminate if a solution with guaranteed accuracy of 99.5% was found or if 50 minutes had passed. We conducted the computations on a Linux server with a 20 core Intel Xeon 640 E5-2689 CPU (3.1GHz per CPU) and with 512GB RAM.

4.4 Results

In 72% of the considered scenarios, we were able to identify a solution with the desired accuracy of 99.5%. In the remaining cases, the guaranteed solution quality never fell below 92%; in scenarios with budgets $B \geq 25$ units, we could always identify solutions with 98% accuracy and above. The greedy algorithm used to compute an initial guess provided a

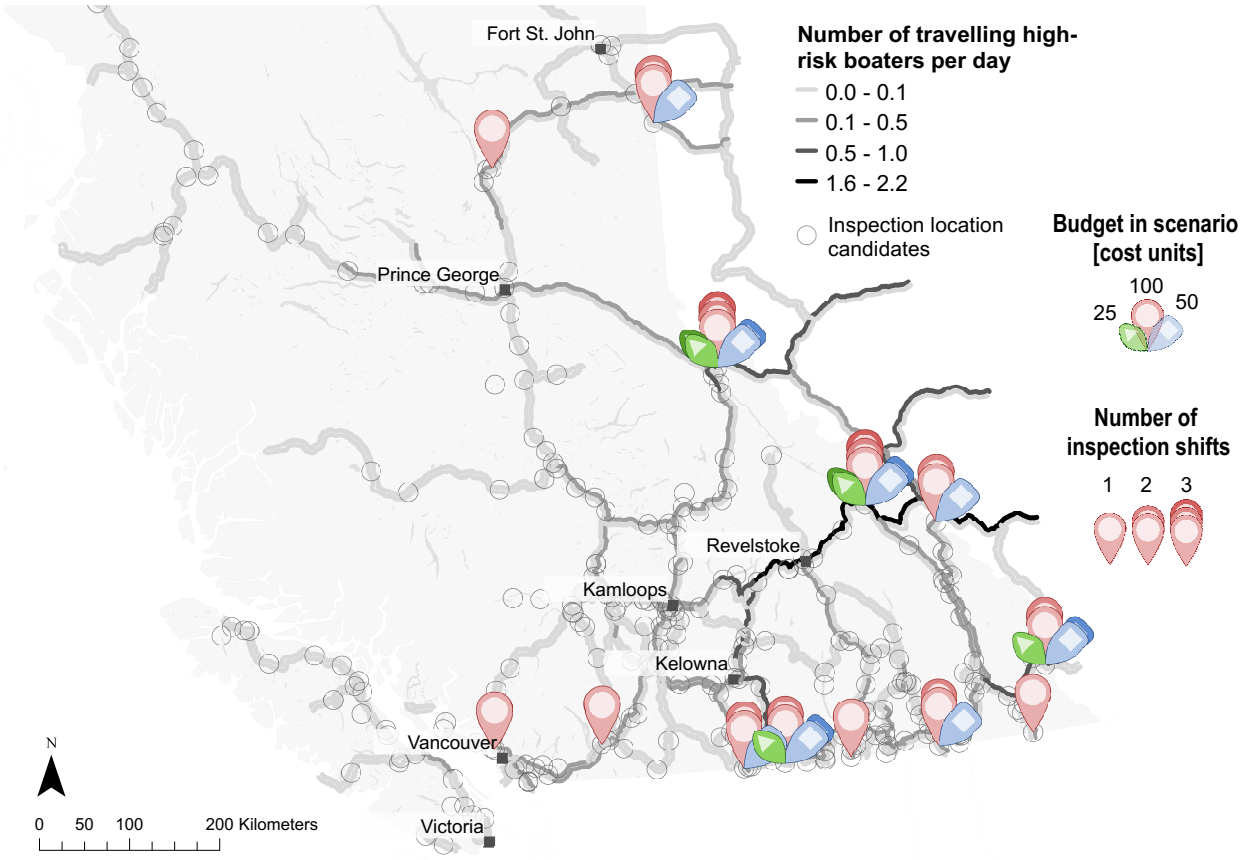


Figure 4.2: Optimal locations and operation shifts for three different budget scenarios. Most inspection stations are placed close to the British Columbian border. The markers depict the optimal inspection locations for each scenario. Green (triangle): optimal locations with a budget of 25 units; blue (square) 50 unit budget; red (circle) 100 unit budget. The number of markers stacked on top of each other corresponds to the optimal numbers of inspection shifts. The darkness of the roads show the estimated boater traffic volume. The hollow circles depict the considered candidates for inspection locations.

solution with 99.5% accuracy in 58% of the considered cases. The initial guesses always had a quality above 90%.

Figure 4.2 displays the optimized locations and operating times for watercraft inspection stations in the considered model scenario. We depict the respective optimal policy under three different budget constraints. The optimal locations for inspections are located close to border crossings if suitable locations are available. However, where the traffic through many border crossings merges on a major highway (e.g. in the Vancouver metropolitan area), it is optimal to place the inspection stations farther inland.

Figure 4.3 depicts characteristics of the optimal inspection stations in different scenarios. The expected traffic volume at an inspection station coincides with the optimized operating

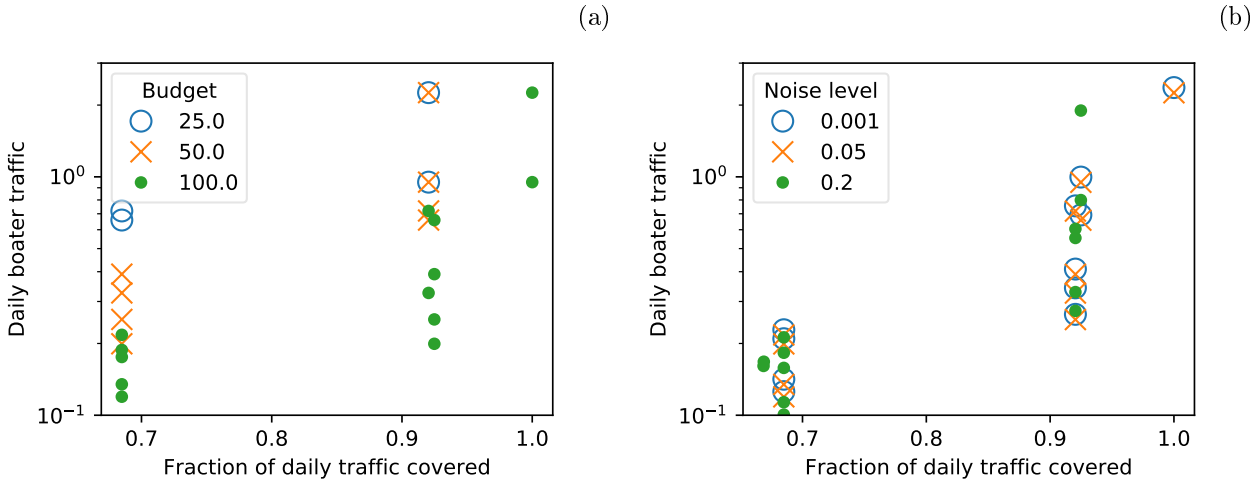


Figure 4.3: Characteristics of the optimized inspection stations in scenarios with (a) different budget constraints, and (b) different levels of uncertainty. Additional budget is preferably spent on additional inspection locations rather than longer operating hours. Increased uncertainty results in resources being distributed over more locations at cost of decreasing operating hours. Overall, however, uncertainty does not have a strong effect on the inspection policy. Each marker corresponds to an inspection station. The position of a marker depicts the daily traffic volume expected at the location and the fraction of daily traffic covered under the optimal operation policy (compliance supposed). The "noise level" denotes the fraction η_c of boaters travelling on routes not covered by the route choice model. Note that the noise level also affects the daily traffic volume at the inspection locations.

times: stations with high expected boater traffic are operated longer than stations with lower traffic. If the budget is increased, some stations are assigned longer operating times. However, larger portions of the additional budget are spent on additional locations (see also Figure 4.2). If the uncertainty in the traffic predictions increases, more inspection stations are set up at the cost of shorter operations. Overall, however, the noise level has little effect on the inspection policy.

Optimizing inspection station operation under a range of different budget allowances showed that a moderate inspection budget, corresponding to about half the 2017 BC inspection budget, suffices to inspect half of the incoming high-risk boaters (Figure 4.4). However, the resources required for inspections increase quickly if more boaters shall be controlled. Thereby, the faction of inspected boaters is limited by boaters' compliance with inspections.

The considered change in the invasion state of three American states had only a moderate impact on inspection policy. The results are depicted in Appendix 4.B. As the additional

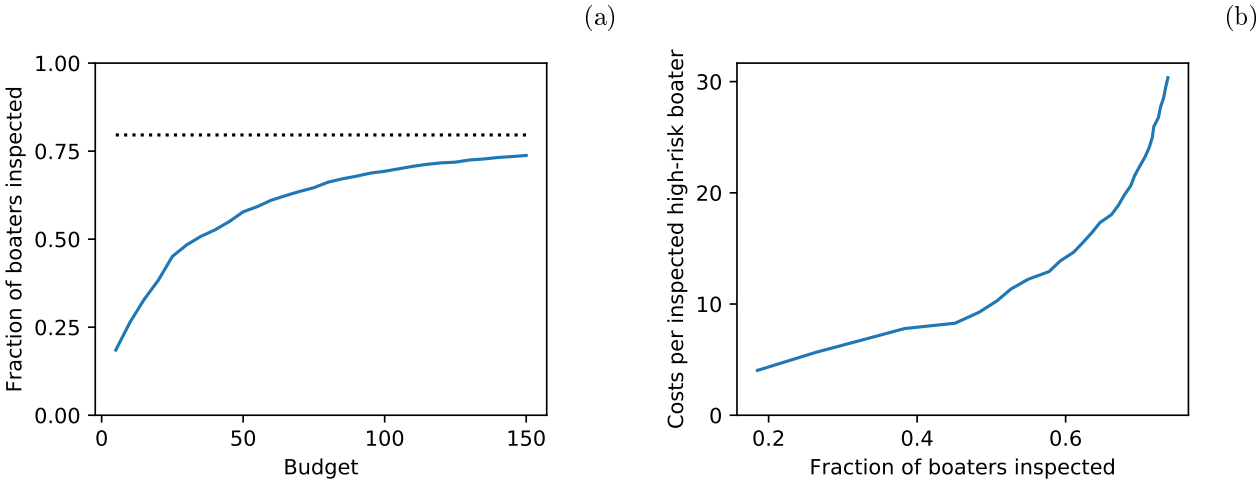


Figure 4.4: Inspection effectiveness dependent on the budget constraint (a) and price per inspected high-risk boater dependent on the proportion of inspected boaters (b). While a large fraction of high-risk boaters can be covered with moderate effort, inspecting all complying boaters is costly. Panel (a) shows the expected fraction of incoming high-risk boaters that can be inspected under the optimal policy. The dotted line shows the level of complying boaters, which is the maximal fraction of boaters that can be inspected.

propagule sources were located south of BC, the inspection effort increased at the southern border under the optimal policy. Furthermore, the optimal policy contained less overnight inspections and distributed resources more evenly across inspection stations.

4.5 Discussion

We presented a method to optimize placement and operating times of watercraft inspection stations. The approach is suited to model management scenarios on a detailed level and gives specific advice for management actions. We applied our approach to invasive mussel management in BC and investigated the impact of budget constraints, model uncertainty, and potential future invasions on management actions and efficiency. However, it must be recognized that our model did not account for all critical operational factors, such as site safety. Nonetheless, the presented results provide valuable insights into optimal management of AIS when combined with critical operational factors.

Most of our results are consistent with common sense. In general, it is optimal to inspect boaters as soon as they enter the managed region. That way, waterbodies close to the border

can be protected. If multiple routes via different border crossings merge close to the border, it can be optimal to inspect boaters after this merging point. Inspections stations should operate longer at locations with high traffic volume. Furthermore, uncertainty in traffic predictions increases the benefit of spreading the inspection efforts over many locations. Driven by these simple principles, our results were remarkably robust throughout considered scenarios and agree well with the watercraft inspection policy currently implemented in BC.

While these qualitative principles may seem obvious, it can be challenging to identify quantitative definitions of terms like "close to the border" and "longer". The difficulty in optimizing management policies is in balancing trade-offs, such as between leaving some waterbodies close to the border unprotected and maximizing the overall number of inspected boaters, or between long-time operation of few highly frequented inspection stations and distribution of resources over many locations. As the approach proposed in this paper is suited to account for these trade-offs, it is a valuable extension to earlier more theoretical results on AIS management (Potapov and Lewis, 2008; Potapov et al., 2008; Finnoff et al., 2010).

Considering scenarios with different budget constraints allowed us to investigate the trade-off between resources invested in AIS control and the number of inspected high-risk boaters. In combination with the expected monetary damage caused by the arrival of an uncontrolled boater at an uninvaded lake, this trade-off curve can be used to identify the optimal budget for inspections. Since both invasion risk and damages due to invasions are difficult to quantify, a rigorous computation of the optimal inspection budget may not always be feasible in practice. Nonetheless, the cost-effectiveness curve provides an estimate of the efficacy of control efforts and shows which budget is required to achieve a certain management goal.

In the case of AIS control in BC, a moderate budget suffices to inspect a significant portion of the incoming high-risk boaters. This is because boater traffic in BC concentrates on a small number of major highways. Nevertheless, an attempt to inspect all high-risk boaters would be very costly, as many minor roads would have to be considered as well. It

could therefore be more cost-effective to implement measures to increase the compliance of boaters, e.g. through additional road signs or public outreach and education.

We see particular use of our approach in its potential to optimize rapid response actions under scenarios of interest. The extended invasion scenario considered in this paper shows that slight adjustments to the inspection policy may suffice to react on the new conditions. In a similar manner, our approach could be used to assess the benefit from cross-border collaborations, in which inspection efforts are combined to control the boater inflow to a large joint area. Due to the flexibility of our model, managers can consider a variety of scenarios at little cost.

4.5.1 Limitations and possible extensions

The accuracy of our approach in real-world applications is strongly dependent on the accuracy and level of detail of the utilized data and models. Therefore, the results should be combined with expert knowledge and refined iteratively if necessary. Nonetheless, our approach can be extended to account for many management constraints and is thus a helpful tool to optimize inspection policies.

Limitations exist with respect to the considered objective function. Though the number of potentially infested watercraft arriving at a waterbody is a valuable proxy for invasion risk, the establishment probability of dreissenid mussels is not linear in propagule pressure (Leung et al., 2004). Hence, our approach is not suited to minimize invasion risk directly. However, high-dimensional non-convex optimization problems are difficult to solve, and minimizing a proxy for invasion risk may thus be the better option in practice. Nonetheless, significant realism could be added by considering the suitability of the destination waterbodies as habitat for AIS. This could be done by weighting boater flows differently dependent on the invasion risk of the destination waterbodies.

Since our traffic model does not explicitly account for the time boaters need to travel between locations, the optimized inspection station operating times may have to be adjusted

to local temporal traffic patterns. Though this shortage in model realism could affect the results significantly if boaters pass multiple inspection stations under the optimal policy, optimal inspection locations are often on independent routes. In the scenario considered in this study, the optimized operating times were all centered around the traffic peak. This indicates that interactions between locations did not affect the operating times and the error due to the simplifying model assumption is small.

Another modelling challenge is to account for uncertainty appropriately. The noise model used in this study is a non-informative null model that treats all potential inspection locations equally. As more boater traffic may be expected at major highways than at minor roads, the noise model could be improved by incorporating location-dependent covariates. However, since our results were not very sensitive to the noise level, a realistic noise model might not change the optimal policy significantly.

Our model did not incorporate site-specific costs and operational constraints. In high-budget scenarios, this let our model suggest overnight inspections at remote sites that are lacking the required infrastructure to safely operate at night. Requirements for overnight inspections include proper road infrastructure (lanes/barriers), lighting, access to safe communication and nearby living accommodations for staff. A lack of living accommodations for staff can also limit the number of staff based in remote locations. These constraints could be incorporated in a more detailed model as well as increased costs at remote locations. A more detailed model could also account for inspection stations operated by neighbouring jurisdictions. As an example, the BC program works closely with the Canadian Border Services Agency and neighbouring provinces and states to receive advanced notifications of high risk watercraft destined for BC. Nonetheless, the presented model includes major factors affecting inspection station operation. Therefore, the model can serve as a helpful resource to inform managers' decisions in parallel with operational constraints.

Another potential extension of our model is to incorporate location-specific or management-dependent compliance rates. At certain sites, such as cross-national border

crossings, compliance can be enforced more easily than at other locations. Compliance may furthermore depend on management efforts: it may be possible to increase the compliance rate of boaters at some costs. In Appendix 4.C, we show how non-uniform and flexible compliance rates can be considered with small model adjustments.

The computational method we used to optimize inspection station operation is well established and builds on a large body of theoretical insights (Ageev and Sviridenko, 2004; Conforti et al., 2014). Nonetheless, the problem is computational difficult, and there may be scenarios in which linear integer solvers fail to provide good solutions. Optimization failures are most prevalent in scenarios in which portions of the budget remain unused under the optimal policy or in which many boaters pass multiple inspection stations under optimized operation. In both cases, the solution to the continuous relaxation of the problem may differ significantly from the integer solution.

However, issues due to unused budget become minor if the considered budget is sufficiently large. Furthermore, the issue may be mitigated by adjusting the budget slightly. Issues with redundant inspection stations, in turn, are unlikely to occur if the propagule donors and recipients are in separate regions. Then, independent inspection locations can often be identified. This is often the case if invasion processes are considered on large scales. Therefore, our approach will yield good results in most applications. We provide more details in Appendix 4.D.

4.5.2 General conclusions for invasive species management

In this paper, we considered specific management scenarios with focus of AIS control in BC. Nonetheless, some common patterns were consistent throughout our results and may thus apply with greater generality. These principles may be used as rules of thumb if no comprehensive modelling and optimization effort is possible. Below we summarize these conclusions.

- Inspection stations should be placed close to the border of the uninfested region. Consequently, cross-border collaborations between uninvaded jurisdictions have a high potential of improving the cost-effectiveness of control.

- If traffic flows merge close to the border, inspections are more cost-effective after the merging point. Hence, identifying such points is crucial for successful management.

- If traffic predictions involve a high level of uncertainty, inspection efforts should be distributed over many locations at the cost lower inspection effort at each site.

- If a high reduction of the propagule inflow is desired, it may be most cost-effective to implement measures increasing the compliance rate rather than operating more inspection stations for longer hours.

Appendices

4.A Greedy rounding algorithm

In this Appendix, we describe the greedy rounding algorithm we applied to obtain initial guesses for the general branch and bound solvers. We start by introducing some helpful notation. Let P be the linear integer problem that we desire to solve and P_{cont} its continuous relaxation, in which decision variables may attain fractional values. We write $\boldsymbol{x}$ for the N-dimensional vector of decision variables, indexed by $(l, s) \in L \times S$. Let $\boldsymbol{e}_{ls}$ be a unit vector that is 0 everywhere except for the component corresponding to the index (l, s). Suppose that $C(\boldsymbol{x})$ denotes the cost for implementing a policy given by $\boldsymbol{x}$. We provide pseudo code for the greedy rounding algorithm in Algorithm A4.1.

The algorithm repeatedly solves the relaxed problem P_{cont} with different constraints fixing some decision variables to integer values. The algorithm proceeds in two phases. In the first phase, the maximal non-integral decision variable that can be rounded up without violating the budget constraint is determined. With this variable fixed, problem P_{cont} is solved again. When no additional component can be rounded up without violating the cost constraint, all previous constraints are removed, and the set of utilized locations is fixed instead. The algorithm sets a flag *locked* to `True` to show that the second phase of the algorithm has started.

In the second phase, components of $\boldsymbol{x}$ are still rounded up if possible. However, now we do not round up the largest non-integral component of $\boldsymbol{x}$. Instead, we determine for some location $l \in L$ with non-integral operation (i.e. $\exists \tilde{s} \in S_l : x_{l\tilde{s}} \notin \{0, 1\}$) the first time interval

$$t := \underset{t \in T}{\text{minargmax}} \left\{ \sum_{s \in S_{lt}} x_{ls} \middle| x_{ls} < 1 \forall s \in S_{lt} \right\} \tag{A4.1}$$

that is operated strongest at this location. Here, minargmax $\{\cdot\}$ refers to the minimal admissible value for argmax $\{\cdot\}$ if the maximum is not unique. Then, we round up the latest affordable shift $s \in S_l$ that covers the time interval t and add $x_{ls} = 1$ to the set of constraints. If no additional shift can be operated at location l, we add a constraint fixing the usage of this location: $x_{ls} = \lfloor x_{ls} \rfloor$ for all $s \in S_l$.

Distinguishing between the two phases of the algorithm yields optimized operating times. Suppose we are in phase 2, and consider the example depicted in Figure A4.2. The solution to the relaxed problem P_{cont} suggests that 3 inspection shifts s_1, s_2, and s_3 are conducted fractionally at the considered location l. Thereby, s_2 overlaps with s_1 and s_3. The respective operation intensities are $x_{ls_1} = x_{ls_3} = 0.8$ and $x_{ls_2} = 0.2$. The budget assigned to this location does not suffice to operate both s_1 and s_3 completely. Hence, only one shift can be operated at l. Naive greedy rounding would suggest to operate shift s_1, as it is the earliest shift with the maximal fractional operation. However, in the optimal solution, the time interval between 8 AM and 4 PM should be operated strongest. Therefore, shift s_2 would be the optimal choice.

In its second phase, the suggested algorithm rounds up shifts based on the maximal *cumulative* operation rather than choosing the shift with the highest operation variable. Nonetheless, it would be of disadvantage to apply this rounding scheme in phase 1 of the algorithm, in which the set of used locations is not fixed. In this case, shifts in the middle of the day would always be chosen with preference, which make operation of *two* shifts on a day less efficient. In the second phase, it is typically known how many shifts should be operated at each location.

Slight improvements to the suggested algorithm are possible. For example, we added constraints in phase 1 to suppress fractional operation of shifts that would not be affordable completely under the costs of the already constrained variables. However, this improvement is unlikely to have a major effect on the results.

Algorithm A4.1: Greedy rounding algorithm.

Function `lock_location`($\tilde{\boldsymbol{x}}$, l, Θ)**:**
  foreach $s \in S_l$ **do**
    $\Theta := \Theta \cup \{x_{ls} = \tilde{x}_{ls}\}$;
locked := `False`; $\Theta := \emptyset$;
while `True` **do**
  $\boldsymbol{x}$:= solution to P_{cont} subject to additional constraints in Θ;
  if $\boldsymbol{x} \in \mathbb{Z}^N$ **then**
    return $\boldsymbol{x}$;
  $\tilde{\boldsymbol{x}} := \lfloor \boldsymbol{x} \rfloor$;
  $\Omega := \{(l, s) \in L \times S \,|\, 0 < \boldsymbol{x}_{ls} < 1;\, C(\tilde{\boldsymbol{x}} + \boldsymbol{e}_{ls}) \leq B\}$;
  if $\Omega = \emptyset$ **then**
    if not *locked* **then**
      locked := `True`; $\Theta := \emptyset$;
      foreach $l \in L$ **with** $\max_{s \in S_l} x_{ls} = 1$ **do**
        $\Theta := \Theta \cup \{\max_{s \in S_l} x_{ls} = 1\}$;
    else
      l := some location with $0 < x_{ls} < 1$ for some $s \in S_l$;
      `lock_location`($\tilde{\boldsymbol{x}}$, l, Θ);
  else
    $(l, s) := \underset{(l,s) \in \Omega}{\text{minargmax}}\, x_{ls}$;
    if *locked* **then**
      $t := \underset{t \in T}{\text{minargmax}} \left\{ \sum_{s \in S_{lt}} x_{ls} \middle| x_{ls} < 1 \forall s \in S_{lt} \right\}$;
      $\Psi := \left\{ s \in S_{lt} \middle| C(\tilde{\boldsymbol{x}} + \boldsymbol{e}_{ls}) \leq B \right\}$;
      if $\Psi = \emptyset$ **then**
        `lock_location`($\tilde{\boldsymbol{x}}$, l, Θ);
        continue;
      else
        $s := \max S_{lt}$;
    $\Theta := \Theta \cup \{x_{ls} = 1\}$;

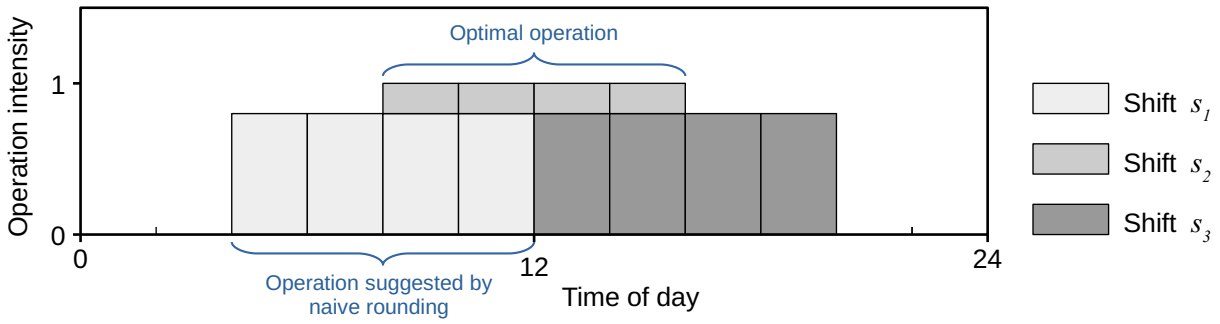


Figure A4.1: Motivation for the changed rounding procedure in phase 2 of the greedy rounding algorithm. The operation intensity is depicted as a function of time for some inspection location. The intervals on the time axis depict the discretization of the day time. The grey boxes show the extent to which the inspection station would be operated in the respective time intervals if fractional operation would be allowed. The boxes' colours correspond to the respective operation shifts. Naive greedy rounding would suggest to operate shift s_1. Improved rounding, however, would prefer the time interval in which the cumulative operation is maximal (shift s_2).

4.B Optimal inspection policy if additional parts of the USA are infested

To assess how the optimal inspection policy changes if additional states are infested, we considered a scenario in which boaters from Idaho, Oregon, and Wyoming were considered high-risk boaters. The results are depicted in Figure A4.2. As more high-risk boaters enter BC via the southern border, inspection efforts at this border are increased. The required resources are freed by operating fewer inspection stations over night and by abandoning inspection locations in the north. Nonetheless, the overall changes are moderate, because even in the changed invasion scenario most high-risk boaters are expected to enter the province via the eastern border.

4.C Flexible and location-specific compliance rates

It may be more cost-effective to implement measures enforcing boaters' compliance than to operate many inspection stations for long hours. Furthermore, compliance of boaters may

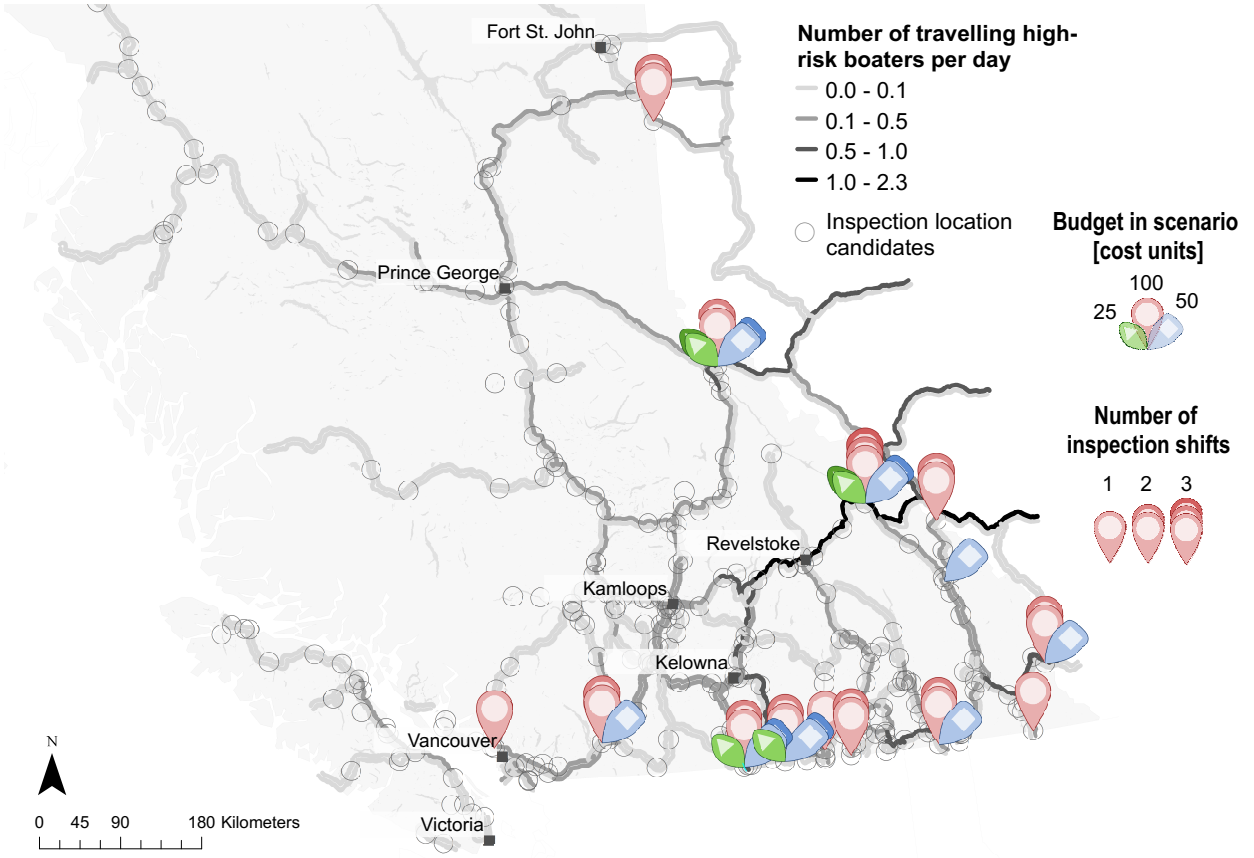


Figure A4.2: Optimal locations and operation shifts assuming that Idaho, Oregon, and Wyoming are mussel invaded. Compared to the base scenario with fewer infested States south of BC, more inspections are conducted at the southern border of the province. The symbols have the same meaning as in Figure 4.2 (main text).

be higher or enforced more easily at some specific locations. In this appendix, we show how the approach presented in this paper can be adjusted to take these factors into account.

4.C.1 Location-specific compliance rates

We start by considering the case of non-uniform compliance rates. To that end, we split the boater flows based on the compliance of the boaters. Let C be the set of possible compliance rates, $c_l \in C$ the expected compliance rate of boaters at location $l \in L$, and L_c the set of locations with compliance rate $\tilde{c} \geq c$. For a route $r \in R$ and a time interval $t \in T$ Let n_{rtc} be the expected number of boaters who travel along route $r \in R$, started their journey in time interval $t \in T$, and comply at all inspection locations l with $l_c \geq c$ but not at inspection locations with $l_c < c$. These boaters will be inspected if and only if

$$\sum_{l \in L_r \cap L_c} \sum_{s \in S_{lrt}} x_{ls} \quad \geq \quad 1. \tag{A4.2}$$

As in the main text, x_{ls} is a binary variable denoting whether inspections are conducted at location $l \in L$ in shift $s \in S$. Consequently, the total number of inspected boaters is given by

$$F_{\text{loc-compliance}}(\boldsymbol{x}) \quad := \quad \sum_{c \in C} \sum_{r \in R} \sum_{t \in T} \min \left\{ 1, \sum_{l \in L_r \cap L_c} \sum_{s \in S_{lrt}} x_{ls} \right\} n_{rtc}. \tag{A4.3}$$

This function can be optimized with the same method discussed in the main text. With a similar approach, time-dependent compliance rates could be incorporated, too.

4.C.2 Flexible compliance rates

In some applications, the compliance rate may be altered at a specific cost. If these costs can be expressed as a convex function of the achieved compliance rate, a flexible compliance

rate can be incorporated in our model easily. Below, we consider for simplicity the base case with a uniform compliance rate at all locations. Allowing location-specific flexible compliance rates can be done by combining the two approaches introduced in this appendix.

Let n_{rt} be the expected number of boaters travelling on route $r \in R$ and who started their journey in time interval $t \in T$. Note that other than in the main text, compliance of these boaters is not supposed. Altering equation (4.5) from the main text to

$$F_{\text{flex-compliance}}(\boldsymbol{x}) \;:=\; c \sum_{r \in R} \sum_{t \in T} \min \left\{ 1, \sum_{l \in L_r} \sum_{s \in S_{lrt}} x_{ls} \right\} n_{rt} \tag{A4.4}$$

accounts for the flexible compliance rate c.

Let us assume assume that the costs for enforcing a specific compliance rate c at a location $l \in L$ and during shift $s \in S$ are given by the linear function

$$\text{cost}_{ls}(c) \;=\; \alpha_l \left(c - c_0 \right), \tag{A4.5}$$

whereby c_0 is the base compliance rate if no actions are taken to increase compliance. More complex cost functions can be modelled with convex piece-wise linear functions or general convex functions. Adding these costs to the overall cost function changes the cost constraint to

$$\sum_{l \in L} \left(\sum_{s \in S_l} \left(c_{ls}^{\text{shift}} + \alpha_l \left(c - c_0 \right) \right) x_{ls} + c_l^{\text{loc}} \max_{r \in R,\, t \in T} \left(\sum_{s \in S_{lrt}} x_{ls} \right) \right) \;\leq\; B. \tag{A4.6}$$

In addition to changing the objective function and the cost constraint, we have to introduce one further constraint limiting the compliance rate to the feasible range:

$$c_0 \;\leq\; c \;\leq\; 1.$$

With these changes, the compliance rate can be optimized along with the inspection locations and operating times.

4.D Difficult inspection optimization scenarios

In many real-world instances, good solutions to the linear integer problems derived in this paper can be identified within reasonable time. Nonetheless there are examples in which the optimization is computationally challenging. In this appendix, we discuss two important mechanism that can make it difficult to find a highly optimal solution in short time. We also provide examples for the discussed mechanisms.

Difficulties can arise (1) if a significant fraction of the budget is unused under the optimal policy and (2) if many boaters pass multiple operated inspection locations under the optimal policy. We start by considering budget-related issues before we discuss problems arising from unfavourable relationships between potential inspection locations. At the end of this appendix we discuss why these challenges are not of major concern in many real-world applications. To simplify explanations, we consider the case of optimizing inspection station placement only. The described mechanisms extend easily to the full problem in which operating times must be optimized as well.

4.D.1 Difficulties due to cost constraints

Let us first consider a scenario in which a fraction of the given budget remains unused under the optimal policy. For example, suppose that operation of an inspection station costs 5 cost units and that we are given a budget of 9 units. Consequently, 4 cost units of the budget will remain unused. To obtain an approximate solution and obtain an upper bound on the optimal objective value, solvers consider the problem's continuous relaxation, in which partial use of inspection locations (and shifts) is permissible. In this relaxed scenario, all 9 cost units will be spent, which allows the inspection of more boaters than in the realistic scenario with binary choices. Consequently, the upper bound on the solution given by the solution to the

relaxed problem may be much higher than the true optimal solution. This makes it difficult to check whether an identified solution is highly optimal and thus increases computation time.

The problem described above becomes even more difficult if control actions with different costs are possible. Suppose that we may operate one of three inspection stations, which are passed by different sets of boaters, respectively. That is, no boater passes two of the potential inspection locations. Assume that per day $n_1 = n_2 = 5$ boaters pass stations l_1 and l_2, respectively, and that $n_3 = 8$ boaters may be inspected at location 3. Suppose we are given a budget of 9 units and that the costs for operating stations l_1 and l_2 are $c_1 = c_2 = 5$ cost units, whereas operation of station l_3 requires $c_3 = 9$ cost units.

Again, optimizers may consider the problem's continuous relaxation to find an approximate solution and a quality estimate. An optimal solution to the relaxed problem is to operate both station 1 and station 2 fractionally with weight $x_1 = x_2 = 0.9$. Then, the total costs $x_1c_1 + x_2c_2 = 9$ satisfy the budget constraint and the total number of inspected boaters is given by $x_1n_1 + x_2n_2 = 9$. However, in the original integer problem, stations cannot be operated fractionally, and only one station can be chosen. As more boaters pass location 3 than locations l_1 or l_2, it would be optimal to conduct inspections at location l_3, where 8 boaters can be inspected. Applying a greedy rounding algorithm to the relaxed solution, however, would suggest to operate either location l_1 or l_2, where only 5 boaters would be expected.

4.D.2 Difficulties due to unfavourable relations between inspection locations

Besides challenges induced by cost constraints, specific relationships between potential inspection locations can make the optimization difficult. Consider the example depicted in Figure A4.3, whereby an arbitrary number of boaters may drive from each origin/destination (black circle) to each other origin/destination. Suppose that operating an inspection location

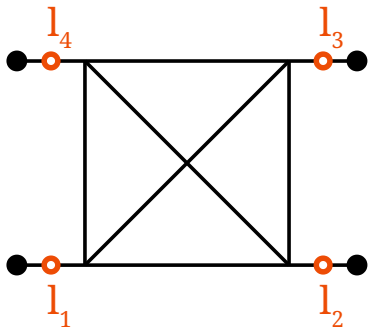


Figure A4.3: Inspection location setup that leads to a challenging optimization problem. The lines denote roads, the solid black circles origins and destination, and the hollow orange circles potential inspection locations.

at any of the permissible locations has unit cost and that we are provided a budget of 2 cost units. If the relaxed version of the problem is considered and fractional operation of stations is permitted, operating each location with intensity $\frac{1}{2}$ would cover all boater flows and hence be the optimal solution. However, if discrete choices must be made, some boaters will not be inspected. As all locations are operated equally in the optimal solution to the relaxed problem, this solution does not provide any hint towards which of the locations should be operated in the original scenario with binary decisions. Therefore, the problem is difficult to solve.

4.D.3 Prevalence of difficult scenarios in real-world applications

Any of the challenging scenarios discussed above can occur in real-world problems. However, certain characteristics of real-world scenarios lower the risk of running into optimization issues. In many management scenarios of interest, various inspection stations can be operated. Problems induced by the budget constraint become less significant if a large budget is considered so that a potential remainder of the budget becomes insignificant. For example, in all scenarios with a budget above 30 units considered in this paper, we reached a solution with at least 98% optimality within minutes. Furthermore, issues induced by budget constraints can be mitigated by investigating alternative scenarios with slightly adjusted budgets.

Scenarios with unfavourable relationships between potential inspection locations can be expected in real-world applications. Note that the issue with the setup in Figure A4.3 persists if the roads connecting the potential inspection locations have a shape different from the road pattern drawn in the figure. Furthermore, the depicted situation may refer to a portion of the road network only, with multiple origins and destinations connected to each of the depicted origin/destination vertices. In fact, situations such as the considered one could appear multiple times in a road network. Therefore, the considered challenges do not only occur in scenarios in which inspections are restricted to locations close to origins and destinations.

Nonetheless, invasion patterns frequent in real-world scenarios reduce the prevalence of such unfavourable inspection station relationships. As short distance dispersal of invasive species is typically more likely than long-distance dispersal, invaded habitat patches form clusters so that the inflow of potentially infested vectors, such as boaters, comes from specific directions only. For example, high-risk boaters enter BC through the southern and eastern border only. Therefore, it is often possible to identify inspection location configurations in which only few high-risk boaters pass multiple operated inspection stations. This simplifies optimization of the inspection policy. Greater optimization challenges can be expected if origins and destinations are intermixed.

Chapter 5

A robust and efficient algorithm to find profile likelihood confidence intervals

5.1 Introduction

5.1.1 Profile likelihood confidence intervals

Confidence intervals are an important tool for statistical inference, used not only to assess the range of predictions that are supported by a model and data but also to detect potential estimability issues (Raue et al., 2009). These estimability issues occur if the available data do not suffice to infer a statistical quantity on the desired confidence level, and the corresponding confidence intervals are infinite (Raue et al., 2009). Due to the broad range of applications, confidence intervals are an integral part of statistical model analysis and widely used across disciplines.

Often, confidence intervals are constructed via Wald's method, which exploits the asymptotic normality of the maximum likelihood estimator (MLE). Though Wald's method is accurate in "benign" use cases, the approach can be imprecise or fail if not enough data are available to reach the asymptotic properties of the MLE. This will be the case, in particular, if the MLE is not unique, i.e. parameters are not identifiable, or if the likelihood is very sensitive to parameter changes beyond some threshold, e.g. in dynamical systems undergoing

bifurcations. Therefore, other methods, such as profile likelihood techniques (Cox and Snell, 1989), are favourable in many use cases.

Both Wald-type and profile likelihood confidence intervals are constructed by inverting the likelihood likelihood ratio test. That is, the confidence interval for a parameter θ_0 encompasses all values $\bar{\theta}_0$ that might suit as acceptable null hypotheses if the parameter were to be fixed; i.e. $H_0 : \theta_0 = \bar{\theta}_0$ could not be rejected versus the alternative $H_1 : \theta_0 \neq \bar{\theta}_0$. As the likelihood ratio statistic is, under regularity conditions, approximately χ^2 distributed under the null hypothesis, the confidence interval is given by

$$I \;=\; \left[\bar{\theta}_0 \,\middle|\, 2\left(\max_{\boldsymbol{\theta}\in\Theta} \ell(\boldsymbol{\theta}) - \max_{\boldsymbol{\theta}\in\Theta\,:\,\theta_0=\bar{\theta}_0} \ell(\boldsymbol{\theta})\right) \leq \chi^2_{1,1-\alpha}\right], \tag{5.1}$$

whereby Θ is the parameter space, ℓ denotes the log-likelihood function, α is the desired confidence level, and $\chi^2_{k,1-\alpha}$ is the $(1-\alpha)$th quantile of the χ^2 distribution with k degrees of freedom.

The function that maps $\bar{\theta}_0$ to the constrained maximum

$$\ell_{\mathrm{PL}}\big(\bar{\theta}_0\big) \;:=\; \max_{\boldsymbol{\theta}\in\Theta\,:\,\theta_0=\bar{\theta}_0} \ell(\boldsymbol{\theta}) \tag{5.2}$$

is called the profile log-likelihood. While Wald's method approximates ℓ and ℓ_{PL} as quadratic functions, profile likelihood confidence intervals are constructed by exact computation of the profile log-likelihood ℓ_{PL}. This makes this method more accurate but also computationally challenging.

5.1.2 Existing approaches

Conceptually, the task of identifying the end points $\theta_0^{\min}$ and $\theta_0^{\max}$ of the confidence interval I is equivalent to finding the maximal (or minimal) value for θ_0 with

$$\ell_{\mathrm{PL}}(\theta_0) \;=\; \ell^* \;:=\; \ell\Big(\hat{\boldsymbol{\theta}}\Big) - \frac{1}{2}\chi^2_{1,1-\alpha}, \tag{5.3}$$

Here, $\hat{\boldsymbol{\theta}}$ denotes the MLE; the value ℓ^* follows from rearranging the terms in the inequality characterizing I (see equation (5.1)).

There are two major perspectives to address this problem. It could either be understood as a one-dimensional root finding problem on ℓ_{PL} or as the constrained maximization (or minimization) problem

$$\theta_0^{\max} = \max_{\boldsymbol{\theta} \in \Theta \,:\, \ell(\boldsymbol{\theta}) \geq \ell^*} \theta_0 \tag{5.4}$$

($\theta_0^{\min}$ analog). Approaches developed from either perspective face the challenge of balancing robustness against efficiency.

The root finding perspective (Cook and Weisberg, 1990; DiCiccio and Tibshirani, 1991; Stryhn and Christensen, 2003; Moerbeek et al., 2004; Ren and Xia, 2019) is robust if small steps are taken and solutions of the maximization problem (5.2) are good initial guesses for the maximizations in later steps. Nonetheless, the step size should be variable if parameters might be not estimable and the confidence intervals large. At the same time, care must be taken with large steps, as solving (5.2) can be difficult if the initial guesses are poor, and algorithms may fail to converge. Therefore, conservative step choices are often advisable even though they may decrease the overall efficiency of the approaches.

The constrained maximization perspective (Neale and Miller, 1997; Wu and Neale, 2012) has the advantage that efficient solvers for such problems are readily implemented in many optimization packages. If the likelihood function is "well behaved", these methods converge very quickly. However, in practical problems, the likelihood function may have local extrema, e.g. due to lack of data, or steep "cliffs" that may hinder these algorithms from converging to a feasible solution. Furthermore, general algorithms are typically not optimized for problems like (5.4), in which the target function is simple and the major challenge is in ensuring that the constraint is met. Therefore, an approach would be desirable that is specifically tailored to solve the constrained maximization (5.4) in a robust and efficient manner.

A first step in this direction is the algorithm by Venzon and Moolgavkar (1988), which solves (5.4) by repeated quadratic approximations of the likelihood surface. As the method is of Newton-Raphson type, it is very efficient as long as the local approximations are accurate. Therefore, the algorithm is fast if the asymptotic normality of the MLE is achieved approximately. Otherwise, the algorithm relies heavily on good initial guesses. Though methods to determine accurate initial guesses exist (Gimenez et al., 2005), the algorithm by Venzon and Moolgavkar (1988) (below abbreviated as VM) can get stuck in local extrema or fail to converge if the likelihood surface has unfavourable properties (see e.g. Ren and Xia, 2019). Moreover, the algorithm will break down if parameters are not identifiable. Thus, VM cannot be applied in important use cases of profile likelihood confidence intervals.

5.1.3 Our contributions

In this paper, we address the issues of VM by introducing an algorithm extending the ideas of Venzon and Moolgavkar (1988). Our algorithm, which we will call *Robust Venzon-Moolgavkar Algorithm* (RVM) below, combines the original procedure with a trust region approach (Conn et al., 2000). That is, the algorithm never steps outside of the region in which the likelihood approximation is sufficiently precise. Furthermore, RVM accounts for unidentifiable parameters, local minima and maxima, and sharp changes in the likelihood surface. Though RVM may not outcompete traditional approaches in problems with well-behaved likelihood functions or in the absence of estimability issues, we argue that RVM is a valuable alternative in the (common) cases that the likelihood function is hard to optimize and the model involves parameters that are not estimable.

Another well-known limitation of the approach by Venzon and Moolgavkar (1988) is that it is not directly applicable to construct confidence intervals for functions of parameters. Often the main research interest is not in identifying specific model parameters but in obtaining model predictions, which can be expressed as a function of the parameters. In addition to presenting a robust algorithm to find confidence intervals for model parameters, we show

how RVM (and the original VM) can also be applied to determine confidence intervals for functions of parameters.

This paper is structured as follows: in the first section, we start by outlining the main ideas behind RVM before we provide details of the applied procedures. Furthermore, we briefly describe how the algorithm can be used to determine confidence intervals of functions of parameters. In the second section, we apply RVM and alternative algorithms to benchmark problems with simulated data. Thereby, we review the implemented alternative algorithms before we present the results. We conclude this paper with a discussion of the benchmark results and the benefits and limitations of RVM in comparison to earlier methods.

All code used in this study, including a Python implementation of RVM, is available online in the supplementary material accompanying this paper.

5.2 Algorithm

5.2.1 Basic ideas

Suppose we consider a model with an n-dimensional parameter vector $\boldsymbol{\theta} := (\theta_0, \ldots, \theta_{n-1})$ and a twice continuously differentiable log-likelihood function ℓ. Assume without loss of generality that we seek to construct a level-α confidence interval for the parameter θ_0, and let $\widetilde{\boldsymbol{\theta}} := (\theta_1, \ldots, \theta_{n-1})^\top$ be the vector of all remaining parameters, called nuisance parameters. For convenience, we may write $\ell = \ell(\boldsymbol{\theta})$ as a function of the complete parameter vector or $\ell = \ell\left(\theta_0, \widetilde{\boldsymbol{\theta}}\right)$ as a function of the parameter of interest and the nuisance parameters.

The algorithm RVM introduced in this paper searches the right end point $\theta_0^{\max}$ (equation (5.4)) of the confidence interval I. The left end point can be identified with the same approach if a modified model is considered in which ℓ is flipped in θ_0. As RVM builds on the method by Venzon and Moolgavkar (1988), we start by recapitulating their algorithm VM below.

Let $\boldsymbol{\theta}^* \in \Theta$ be the parameter vector at which the parameter of interest is maximal, $\theta_0^* = \theta_0^{\max}$, and $\ell(\boldsymbol{\theta}^*) \geq \ell^*$. Venzon and Moolgavkar (1988) note that $\boldsymbol{\theta}^*$ satisfies the following necessary conditions:

1. $\ell(\boldsymbol{\theta}^*) = \ell^*$ and
2. ℓ is in a local maximum with respect to the nuisance parameters, which implies $\frac{\partial \ell}{\partial \widetilde{\boldsymbol{\theta}}}(\boldsymbol{\theta}^*) = 0$.

The algorithm VM searches for $\boldsymbol{\theta}^*$ by minimizing both the log-likelihood distance to the threshold $|\ell(\boldsymbol{\theta}) - \ell^*|$ and the gradient of the nuisance parameters $\frac{\partial \ell}{\partial \widetilde{\boldsymbol{\theta}}}$. To this end, the algorithm repeatedly approximates the log-likelihood surface ℓ with second order Taylor expansions $\hat{\ell}$. If $\boldsymbol{\theta}^{(i)}$ is the parameter vector in the i^{th} iteration of the algorithm, expanding ℓ around $\boldsymbol{\theta}^{(i)}$ yields

$$\begin{aligned} \hat{\ell}(\boldsymbol{\theta}) &:= \ell\left(\boldsymbol{\theta}^{(i)}\right) + \boldsymbol{g}^\top \left(\boldsymbol{\theta} - \boldsymbol{\theta}^{(i)}\right) + \frac{1}{2}\left(\boldsymbol{\theta} - \boldsymbol{\theta}^{(i)}\right)^\top \underline{\mathrm{H}} \left(\boldsymbol{\theta} - \boldsymbol{\theta}^{(i)}\right) \\ &= \bar{\ell} + \widetilde{\boldsymbol{g}}^\top \widetilde{\boldsymbol{\delta}} + g_0\delta_0 + \frac{1}{2}\widetilde{\boldsymbol{\delta}}^\top \underline{\widetilde{\mathrm{H}}} \widetilde{\boldsymbol{\delta}} + \delta_0 \widetilde{\mathbf{H}}_0^\top \widetilde{\boldsymbol{\delta}} + \frac{1}{2}\delta_0 \mathrm{H}_{00} \delta_0 =: \hat{\ell}^\delta\left(\delta_0, \widetilde{\boldsymbol{\delta}}\right). \end{aligned} \tag{5.5}$$

Here, $\boldsymbol{\delta} := \boldsymbol{\theta} - \boldsymbol{\theta}^{(i)}$, $\bar{\ell} := \ell\left(\boldsymbol{\theta}^{(i)}\right)$; $\boldsymbol{g} := \frac{\partial \ell}{\partial \boldsymbol{\theta}}\left(\boldsymbol{\theta}^{(i)}\right)$ is the gradient and $\underline{\mathrm{H}} := \frac{\partial^2 \ell}{\partial \boldsymbol{\theta}^2}\left(\boldsymbol{\theta}^{(i)}\right)$ the Hessian matrix of ℓ at $\boldsymbol{\theta}^{(i)}$. Analogously to notation used above, we split $\boldsymbol{\delta}$ into its first entry δ_0 and the remainder $\widetilde{\boldsymbol{\delta}}$, $\boldsymbol{g}$ into g_0 and $\widetilde{\boldsymbol{g}}$, and write $\mathbf{H}_0$ for the first column of $\underline{\mathrm{H}}$, $\underline{\widetilde{\mathrm{H}}}$ for $\underline{\mathrm{H}}$ without its first column and row, and split $\mathbf{H}_0$ into H_{00} and $\widetilde{\mathbf{H}}_0$.

In each iteration, VM seeks δ_0^* and $\widetilde{\boldsymbol{\delta}}^*$ that satisfy conditions 1 and 2. Applying condition 2 to the approximation $\hat{\ell}^\delta$ (equation (5.5)) yields

$$\widetilde{\boldsymbol{\delta}}^* = -\underline{\widetilde{\mathrm{H}}}^{-1}\left(\widetilde{\mathbf{H}}_0\delta_0 + \widetilde{\boldsymbol{g}}\right). \tag{5.6}$$

Inserting (5.5) and (5.6) into condition 1 gives us

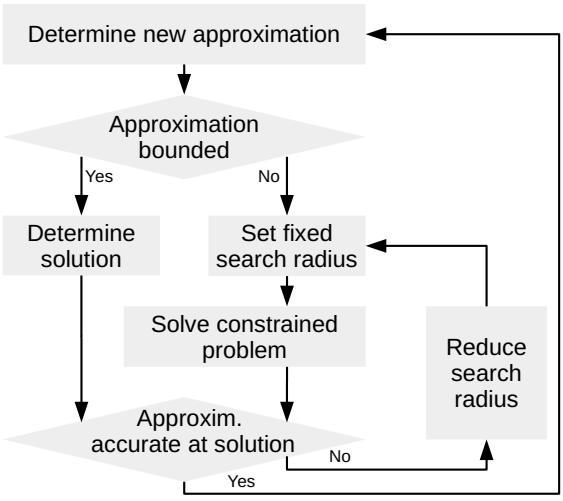


Figure 5.1: Flow chart for RVM. The procedure is repeated until the termination criterion is met and the result is returned.

$$\frac{1}{2}\left(\mathrm{H}_{00} - \widetilde{\mathbf{H}}_0^\top \underline{\widetilde{\mathrm{H}}}^{-1} \widetilde{\mathbf{H}}_0\right) \delta_0^{*2} + \left(g_0 - \widetilde{\boldsymbol{g}}^\top \underline{\widetilde{\mathrm{H}}}^{-1} \widetilde{\mathbf{H}}_0\right) \delta_0^* + \bar{\ell} - \frac{1}{2} \widetilde{\boldsymbol{g}}^\top \underline{\widetilde{\mathrm{H}}}^{-1} \widetilde{\boldsymbol{g}} \;=\; \ell^*, \tag{5.7}$$

which can be solved for δ_0^* if $\underline{\mathrm{H}}$ is negative definite. If equation (5.7) has multiple solutions, Venzon and Moolgavkar (1988) choose the one that minimizes $\boldsymbol{\delta}$ according to some norm. Our algorithm RVM applies a different procedure and chooses the root that minimizes the distance to $\theta_0^{\max}$ without stepping into a region in which the approximation (5.5) is inaccurate. In section 5.2.5, we provide further details and discuss the case in which equation (5.7) has no real solutions.

After each iteration, $\boldsymbol{\theta}$ is updated according to the above results:

$$\boldsymbol{\theta}^{(i+1)} \;=\; \boldsymbol{\theta}^{(i)} + \boldsymbol{\delta}^*. \tag{5.8}$$

If $\ell\left(\boldsymbol{\theta}^{(i+1)}\right) \approx \ell^*$ and $\frac{\partial \ell}{\partial \widetilde{\boldsymbol{\theta}}}\left(\boldsymbol{\theta}^{(i+1)}\right) \approx 0$ up to the desired precision, the search is terminated and $\boldsymbol{\theta}^{(i+1)}$ is returned.

The need to extend the original algorithm VM outlined above comes from the following issues: (1) The quadratic approximation $\hat{\ell}$ may be imprecise far from the approximation

point. In extreme cases, updating $\boldsymbol{\theta}$ as suggested could take us farther away from the target $\boldsymbol{\theta}^*$ rather than closer to it. (2) The approximation $\hat{\ell}$ may be constant in some directions or be not bounded above. In these cases, we may not be able to identify unique solutions for δ_0 and $\widetilde{\boldsymbol{\delta}}$, and the gradient criterion in condition 2 may not characterize a maximum but a saddle point or a minimum. (3) The limited precision of numerical operations can result in discontinuities corrupting the results of VM and hinder the algorithm from terminating.

To circumvent these problems, we introduce a number of extensions to VM. First, we address the limited precision of the Taylor approximation $\hat{\ell}$ with a trust region approach (Conn et al., 2000). That is, we constrain our search for $\boldsymbol{\delta}^*$ to a region in which the approximation $\hat{\ell}$ is sufficiently accurate. Second, we choose some parameters freely if $\hat{\ell}$ is constant in some directions and solve constrained maximization problems if $\hat{\ell}$ is not bounded above. In particular, we detect cases in which ℓ_{PL} approaches an asympote above ℓ^*, which means that θ_0 is not estimable. Lastly, we introduce a method to identify and jump over discontinuities as appropriate. An overview of the algorithm is depicted as flow chart in Figure 5.1. Below, we describe each of our extensions in detail.

5.2.2 The trust region

In practice, the quadratic approximation (5.5) may not be good enough to reach a point close to $\boldsymbol{\theta}^*$ within one step. In fact, since ℓ may be very "non-quadratic", we might obtain a parameter vector for which ℓ and $\frac{\partial \ell}{\partial \widetilde{\boldsymbol{\theta}}}$ are farther from ℓ^* and $\mathbf{0}$ than in the previous iteration. Therefore, we accept changes in θ only if the approximation is sufficiently accurate in the new point.

In each iteration i, we compute the new parameter vector, compare the values of $\hat{\ell}$ and ℓ at the obtained point $\boldsymbol{\theta}^{(i)} + \boldsymbol{\delta}^*$, and accept the step if, and only if, $\hat{\ell}$ and ℓ are close together with respect to a given distance measure. If $\bar{\ell}$ is near the target ℓ^*, we may also check the precision of the gradient approximation $\frac{\partial \hat{\ell}}{\partial \widetilde{\boldsymbol{\theta}}}$ to enforce timely convergence of the algorithm.

If we reject a step, we decrease the value δ_0^* obtained before, reduce the maximal admissible length r of the nuisance parameter vector and solve the constrained maximization problem

$$\widetilde{\boldsymbol{\delta}}^* \;=\; \max_{\widetilde{\boldsymbol{\delta}}\,:\,|\widetilde{\boldsymbol{\delta}}|\leq r} \hat{\ell}^{\delta}\left(\delta_0, \widetilde{\boldsymbol{\delta}}\right). \tag{5.9}$$

As the quadratic subproblem (5.9) appears in classical trust-region algorithms, efficient solvers are available (Conn et al., 2000) and implemented in optimization software, such as in the Python package Scipy (Jones et al., 2001).

We check the accuracy of the approximation at the resulting point $\boldsymbol{\theta}^{(i)} + \boldsymbol{\delta}^*$, decrease the search radius if necessary, and continue with this procedure until the approximation is sufficiently precise. The metric and the tolerance applied to measure the approximation's precision may depend on how far the current log-likelihood $\bar{\ell}$ is from the target ℓ^*. We suggest suitable precision measures in section 5.2.8.

Since it is often computationally expensive to compute the Hessian $\underline{\mathrm{H}}$, we desire to take as large steps δ_0 as possible. However, it is also inefficient to adjust the search radius very often to find the maximal admissible δ_0^*. Therefore, RVM first attempts to make the unconstrained step given by equation (5.5). If this step is rejected, RVM determines the search radius with a log-scale binary search between the radius of the unconstrained step and the search radius accepted in the previous iteration. If even the latter radius does not lead to a sufficiently precise result, we update δ_0^* and r by factors $\beta_0, \beta_1 \in (0,1)$ so that $\delta_0^* \leftarrow \beta_0 \delta_0^*$ and $r \leftarrow \beta_1 r$.

5.2.3 Linearly dependent parameters

The right hand side of equation (5.6) is defined only if the nuisance Hessian $\underline{\widetilde{\mathrm{H}}}$ is invertible. If $\underline{\widetilde{\mathrm{H}}}$ is singular, the maximum with respect to the nuisance parameters is not uniquely defined or does not exist at all. We will consider the second case in the next section and focus on the first case here.

There are multiple options to compute a psudo-inverse of a singular matrix to solve underspecified linear equation systems (Rao, 1967). A commonly used approach is the Moore-Penrose inverse (Penrose, 1955), which yields a solution with minimal norm (Rao, 1967). This is a desirable property for our purposes, as the quadratic approximation is generally most precise close to the approximation point. The Moore-Penrose inverse can be computed efficiently with singular value decompositions (Golub and Kahan, 1965), which have also been applied to determine the number of identifiable parameters in a model (Eubank and Webster, 1985; Viallefont et al., 1998).

Whether or not a matrix is singular is often difficult to know precisely due to numerical inaccuracies. The Moore-Penrose inverse is therefore highly sensitive to a threshold parameter determining when the considered matrix is deemed singular. As the Hessian matrix is typically computed with numerical methods subject to error, it is often beneficial to choose a high value for this threshold parameter to increase the robustness of the method. Too large threshold values, however, can slow down or even hinder convergence of the algorithm.

An alternative method to account for singular Hessian matrices is to hold linearly dependent parameters constant until the remaining parameters form a non-singular system. In tests, this approach appeared to be more robust than applying the Moore-Penrose inverse. Therefore, we used this method in our implementation. We provide details on this method as well as test results in Appendix 5.A. Note that we write $\underline{\widetilde{\mathrm{H}}}^{-1}$ for this generalized inverse below.

To determine whether the approximate system has any solution when $\underline{\widetilde{\mathrm{H}}}$ is singular, we test whether $\widetilde{\boldsymbol{\delta}}^*$ computed according to equations (5.6) and (5.7) indeed satisfies the necessary conditions for a maximum in the nuisance parameters. That is, we check whether

$$0 \approx \frac{\partial}{\partial \widetilde{\boldsymbol{\delta}}} \hat{\ell}^{\boldsymbol{\delta}} = \underline{\widetilde{\mathrm{H}}} \widetilde{\boldsymbol{\delta}}^* + \widetilde{\mathbf{H}}_0 \delta_0^* + \widetilde{\boldsymbol{g}} \tag{5.10}$$

holds up to a certain tolerance. If this is not the case, $\hat{\ell}$ is unbounded, and we proceed as outlined in the next section.

5.2.4 Solving unbounded subproblems

In each iteration, we seek the nuisance parameters $\tilde{\theta}$ that maximize ℓ for the computed value of θ_0. Since the log-likelihood function ℓ is bounded above, such a maximum must exist in theory. However, the *approximate* log-likelihood $\hat{\ell}$ could be unbounded at times, which would imply that the approximation is imprecise for large steps. Since we cannot identify a global maximum of $\hat{\ell}$ if it is unbounded, we instead seek the point maximizing $\hat{\ell}$ in the range where $\hat{\ell}$ is sufficiently accurate.

To test whether $\hat{\ell}$ is unbounded in the nuisance parameters, we first check whether $\widetilde{\underline{\mathrm{H}}}$ is negative semi-definite. If $\widetilde{\underline{\mathrm{H}}}$ is invertible, this test can be conducted by applying a Cholesky decomposition on $-\widetilde{\underline{\mathrm{H}}}$, which succeeds if and only if $\widetilde{\underline{\mathrm{H}}}$ is negative definite. If $\widetilde{\underline{\mathrm{H}}}$ is singular, we use an eigenvalue decomposition. If all eigenvalues are below a small threshold, $\widetilde{\underline{\mathrm{H}}}$ is negative semi-definite. To confirm that $\hat{\ell}$ is bounded, we also test whether equation (5.10) holds approximately if $\widetilde{\underline{\mathrm{H}}}$ is singular (see section 5.2.3).

If either of these tests fails, $\hat{\ell}$ is unbounded. In this case, we set $\delta_0^* \leftarrow r_0$, $r \leftarrow r_1$, for some parameters $r_0, r_1 > 0$ and solve the maximization problem (5.9). The parameters r_0 and r_1 can be adjusted and saved for future iterations to efficiently identify the maximal admissible step. That is, we may increase (or reduce) δ_0^* and r as long as (or until) $\hat{\ell}$ is sufficiently precise. Thereby, we adjust the ratio of δ_0^* and r so that the likelihood increases: $\hat{\ell}^{\delta}\left(\delta_0^*, \widetilde{\boldsymbol{\delta}}^*\right) > \bar{\ell}$.

5.2.5 Step choice for the parameter of interest

Whenever $\hat{\ell}$ has a unique maximum in the nuisance parameters, we compute δ_0^* by solving equation (5.7). This equation can have one, two, or no roots. To discuss how δ_0^* should be chosen in either of these cases, we introduce some helpful notation. First, we write $\hat{\ell}_{\mathrm{PL}}(\theta_0) := \max_{\tilde{\theta}} \hat{\ell}\left(\theta_0, \tilde{\theta}\right)$ for the profile log-likelihood function of the quadratic approximation.

Furthermore, we write in accordance with previous notation

$$\hat{\ell}^{\delta}_{\mathrm{PL}}(\delta_0) \;:=\; \hat{\ell}_{\mathrm{PL}}\Big(\theta_0^{(i)} + \delta_0\Big) \;=\; a\delta_0^2 + p\delta_0 + q + \ell^* \tag{5.11}$$

with $a := \frac{1}{2}\left(\mathrm{H}_{00} - \widetilde{\mathbf{H}}_0\underline{\widetilde{\mathrm{H}}}^{-1}\widetilde{\mathbf{H}}_0\right)$, $p := g_0 - \widetilde{g}^\top\underline{\widetilde{\mathrm{H}}}^{-1}\widetilde{\mathbf{H}}_0$, and $q := \bar{\ell} - \frac{1}{2}\widetilde{g}^\top\underline{\widetilde{\mathrm{H}}}^{-1}\widetilde{g} - \ell^*$ (see equation (5.7)).

Our choices of δ_0^* attempt to increase θ_0 as much as possible while staying in a region in which the approximation $\hat{\ell}$ is reasonably accurate. The specific step choice depends on the slope of the profile likelihood $\hat{\ell}^{\delta}_{\mathrm{PL}}$ and on whether we have already exceeded $\theta_0^{\max}$ according to our approximation, i.e. $\hat{\ell}^{\delta}_{\mathrm{PL}}(0) < \ell^*$. Below, we will first assume that $\hat{\ell}^{\delta}_{\mathrm{PL}}(0) > \ell^*$ and discuss the opposite case later.

5.2.5.1 Case 1: decreasing profile likelihood

If the profile likelihood decreases at the approximation point, i.e. $p < 0$, we select the smallest positive root:

$$\delta_0^* \;=\; \begin{cases} -\frac{q}{p} & \text{if } a = 0 \\ -\frac{1}{2a}\left(p + \sqrt{p^2 - 4aq}\right) & \text{else.} \end{cases} \tag{5.12}$$

Choosing $\delta_0^* > 0$ ensures that the distance to the end point $\theta_0^{\max}$ decreases in this iteration. Choosing the smaller positive root increases our trust in the accuracy of the approximation and prevents potential convergence issues (see Figure 5.2a).

If $\hat{\ell}^{\delta}_{\mathrm{PL}}$ has a local minimum above the threshold ℓ^*, equation (5.11) does not have a solution, and we may attempt to decrease the distance between $\hat{\ell}^{\delta}_{\mathrm{PL}}$ and ℓ^* instead. This procedure, however, may let RVM converge to a local minimum in $\hat{\ell}^{\delta}_{\mathrm{PL}}$ rather than to a point with $\hat{\ell}^{\delta}_{\mathrm{PL}} = \ell^*$. Therefore, we "jump" over the extreme point by doubling the value of δ_0^*. That is, we choose

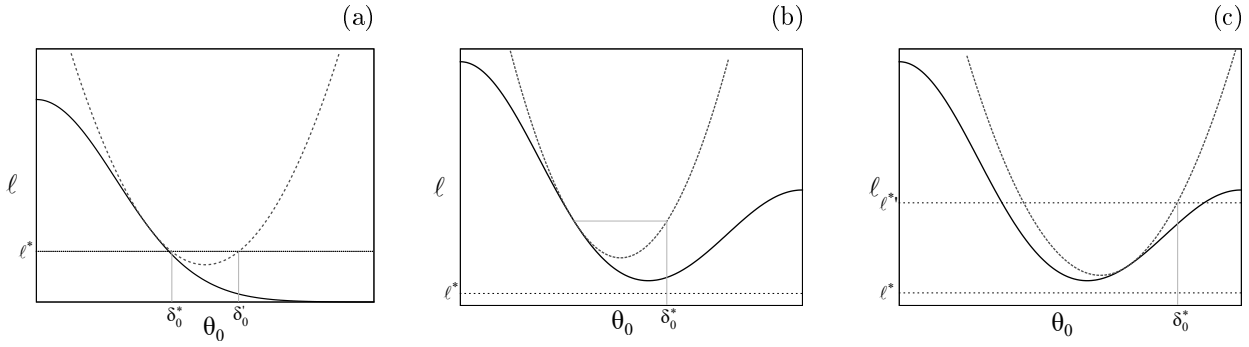


Figure 5.2: Step choice for θ_0 in special cases. The figures depict the profile likelihood function ℓ_{PL} (solid black), quadratic approximation $\hat{\ell}_{\mathrm{PL}}$ (dashed parabola), and the threshold log-likelihood ℓ^*. (a) The approximation has two roots δ_0^* and δ_0'. Though the largest root of ℓ is searched, the smaller root of $\hat{\ell}$ is closest to the desired result. In fact, consistently choosing the larger root would let the algorithm diverge. (b) If ℓ_{PL} is decreasing but $\hat{\ell}_{\mathrm{PL}}$ does not assume the threshold value ℓ^*, we "jump" over the local minimum. (c) If ℓ_{PL} is increasing but $\hat{\ell}_{\mathrm{PL}}$ does not assume the threshold value ℓ^*, we reset the target value to an increased value $\ell^{*\prime}$.

$$\delta_0^* \quad = \quad -\frac{p}{a} \tag{5.13}$$

if $p^2 < 4aq$ (see Figure 5.2b).

5.2.5.2 Case 2: increasing profile likelihood

If the profile likelihood increases at the approximation point, i.e. $p > 0$, equation (5.11) has a positive root if and only if $\hat{\ell}_{\mathrm{PL}}$ is concave down; $a < 0$. We choose this root whenever it exists:

$$\delta_0^* \quad = \quad -\frac{1}{2a}\left(p + \sqrt{p^2 - 4aq}\right). \tag{5.14}$$

However, if $\hat{\ell}_{\mathrm{PL}}$ grows unboundedly, equation (5.11) does not have a positive root. In this case, we change the threshold value ℓ^* temporarily to a value $\ell^{*\prime}$ chosen so that equation

(5.11) has a solution with the updated threshold (see Figure 5.2c). For example, we may set

$$\ell^{*\prime} \;:=\; \max\left\{\hat{\ell}^{\delta}_{\mathrm{PL}}(0)+1,\, \frac{\bar{\ell}+\ell\left(\hat{\boldsymbol{\theta}}\right)}{2}\right\}.$$

This choice ensures that a solution exists while at the same time reaching local likelihood maxima quickly. After resetting the threshold, we proceed as usual.

To memorize that we changed the threshold value ℓ^*, we set a flag `maximizing` := `True`. In future iterations $j > i$, we set the threshold ℓ^* back to its initial value ℓ^*_0 and `maximizing` := `False` as soon as $\ell\left(\boldsymbol{\theta}^{(j)}\right) < \ell^*_0$ or $\hat{\ell}_{\mathrm{PL}}$ is concave down at the approximation point $\boldsymbol{\theta}^{(j)}$.

5.2.5.3 Case 3: constant profile likelihood

If the profile likelihood has a local extremum at the approximation point, i.e. $p = 0$, $a \neq 0$, we proceed as in cases 1 and 2: if $a > 0$, we proceed as if $\hat{\ell}_{\mathrm{PL}}$ were increasing, and if $a < 0$, we proceed as if $\hat{\ell}_{\mathrm{PL}}$ were decreasing. However, the approximate profile likelihood could also be constant, $a = p = 0$. In this case, we attempt to make a very large step to check whether we can push θ_0 arbitrarily far. In section 5.2.6, we discuss this procedure in greater detail.

5.2.5.4 Profile likelihood below the threshold

If the profile likelihood at the approximation point is below the threshold, $\hat{\ell}^{\delta}_{\mathrm{PL}}(0) < \ell^*$, we always choose the smallest possible step:

$$\delta^*_0 \;=\; \begin{cases} -\frac{1}{2a}\left(p+\sqrt{p^2-4aq}\right) & \text{if } a \neq 0,\, p < 0 \\ -\frac{q}{p} & \text{if } a = 0,\, p \neq 0 \\ -\frac{1}{2a}\left(p-\sqrt{p^2-4aq}\right) & \text{if } a \neq 0,\, p > 0. \end{cases} \tag{5.15}$$

This shall bring us to the admissible parameter region as quickly as possible.

As RVM rarely steps far beyond the admissible region in practice, equation (5.15) usually suffices to define δ_0^*. Nonetheless, if we find that $\hat{\ell}_{PL}^{\boldsymbol{\delta}}$ has a local maximum below the threshold, i.e. $p^2 < 4qa$, we may instead maximize $\hat{\ell}_{PL}^{\delta}$ as far as possible:

$$\delta_0^* \;=\; -\frac{p}{2a}. \tag{5.16}$$

If we have already reached a local maximum ($p \approx 0$), we cannot make a sensible choice for δ_0. In this case, we may recall the iteration $k := \underset{j\,:\,\ell(\theta^{(j)}) \geq \ell^*}{\operatorname{argmax}}\ \theta_0^{(j)}$, in which the largest admissible θ_0 value with $\ell\big(\theta^{(k)}\big) \geq \ell^*$ has been found so far, and conduct a binary search between $\theta^{(i)}$ and $\theta^{(k)}$ until we find a point $\theta^{(i+1)}$ with $\ell\big(\theta^{(i+1)}\big) \geq \ell^*$.

5.2.6 Identifying inestimable parameters

In some practical scenarios, the profile log-likelihood ℓ_{PL} will never fall below the threshold ℓ^*, which means that the considered parameter is not estimable. In these cases, RVM may not converge. However, often it is possible to identify inestimable parameters by introducing a step size limit $\delta_0^{\max}$. If the computed step exceeds the maximal step size, $\delta_0^* > \delta_0^{\max}$ and the current function value exceeds the threshold value, i.e. $\bar{\ell} \geq \ell^*$, we set $\delta_0^* := \delta_0^{\max}$ and compute the corresponding nuisance parameters. If the resulting log-likelihood $\ell\big(\theta^{(i)} + \boldsymbol{\delta}^*\big)$ is not below the threshold ℓ^*, we let the algorithm terminate, raising a warning that the parameter θ_0 is not estimable. If $\ell\big(\theta^{(i)} + \boldsymbol{\delta}^*\big) < \ell^*$, however, we cannot draw this conclusion and decrease the step size until the approximation is sufficiently close to the original function.

The criterion suggested above may not always suffice to identify inestimable parameters. For example, if the profile likelihood is constant but the nuisance parameters maximizing the likelihood change non-linearly, RVM may not halt. For this reason, and also to prevent unexpected convergence issues, it is advisable to introduce an iteration limit to the algorithm. If the iteration limit is exceeded, potential estimability issues issues may be investigated further.

5.2.7 Discontinuities

RVM is based on quadratic approximations and requires therefore that ℓ is differentiable twice. Nonetheless, discontinuities can occur due to numerical imprecision even if the likelihood function is continuous in theory. Though we may still be able to compute the gradient g and the Hessian $\underline{\mathrm{H}}$ in these cases, the resulting quadratic approximation will be inaccurate even if we take very small steps. Therefore, these discontinuities could hinder the algorithm from terminating.

To identify discontinuities, we define a minimal step size ϵ, which may depend on the gradient $\boldsymbol{g}$. If we reject a step with small length $|\boldsymbol{\delta}^*|\,\mathrm{length}\,(\epsilon)$, we may conclude that ℓ is discontinuous at the current approximation point $\boldsymbol{\theta}^{(i)}$. To determine the set D of parameters responsible for the issue, we decompose $\boldsymbol{\delta}^*$ into its components. We initialize $D \leftarrow \emptyset$ and consider, with the j^{th} unit vector $\boldsymbol{e}_j$, the step $\boldsymbol{\delta}^{*\prime} := \sum_{j \le k,\, j \neq D} \boldsymbol{e}_j \delta_j^*$ until $\hat{\ell}^{\delta}(\boldsymbol{\delta}^{*\prime}) \not\approx \ell^{\delta}(\boldsymbol{\delta}^{*\prime})$ for some $k < n$. When we identify such a component, we add it to the set D and continue the procedure.

If we find that ℓ is discontinuous in θ_0, we check whether the current nuisance parameters maximize the likelihood, i.e. ℓ is bounded above and $\widetilde{\boldsymbol{g}}$ is approximately $\mathbf{0}$. If the nuisance parameters are not optimal, we hold θ_0 constant and maximize ℓ with respect to the nuisance parameters. Otherwise, we conclude that the profile likelihood function has a jump discontinuity. In this case, our action depends on the current log-likelihood value $\bar{\ell}$, the value of ℓ at the other end of the discontinuity, and the threshold ℓ^*.

- If $\ell\left(\boldsymbol{\theta}^{(i)} + \boldsymbol{e}_0\delta_0^*\right) \geq \ell^*$ or $\ell\left(\boldsymbol{\theta}^{(i)}\right) < \ell\left(\boldsymbol{\theta}^{(i)} + \boldsymbol{e}_0\delta_0^*\right)$, we accept the step regardless of the undesirably large error.
- If $\ell\left(\boldsymbol{\theta}^{(i)} + \boldsymbol{e}_0\delta_0^*\right) < \ell^*$ and $\ell\left(\boldsymbol{\theta}^{(i)}\right) \geq \ell^*$, we terminate and return $\theta_0^{(i)}$ as the bound of the confidence interval.
- Otherwise, we cannot make a sensible step and try to get back into the admissible region by conducting the binary search procedure we have described in section 5.2.5.4.

If ℓ is discontinuous in variables other than θ_0, we hold the variables constant whose change decreases the likelihood and repeat the iteration with a reduced system. After a given number of iterations, we release these parameters again, as θ may have left the point of discontinuity.

Since we may require that not only $\hat{\ell}$ but also its gradient are well approximated, a robust implementation of RVM should also handle potential gradient discontinuities. The nuisance parameters causing the issues can be identified analogously to the procedure outlined above. All components in which the gradient changes its sign from positive to negative should be held constant, as the likelihood appears to be in a local maximum in these components. The step in the remaining components may be accepted regardless of the large error.

5.2.8 Suitable parameters and distance measures

The efficiency of RVM depends highly on the distance measures and parameters applied when assessing the accuracy of the approximation and updating the search radius of the constrained optimization problems. If the precision measures are overly conservative, then many steps will be needed to find $\boldsymbol{\theta}^*$. If the precision measure is too liberal, in turn, RVM may take detrimental steps and might not even converge.

We suggest the following procedure: (1) we always accept forward steps with $\delta_0^* \geq 0$ if the true likelihood is larger than the approximate likelihood, $\ell^\delta(\boldsymbol{\delta}^*) \geq \hat{\ell}^\delta(\boldsymbol{\delta}^*)$. (2) If the approximate likelihood function is unbounded, we require that the likelihood increases $\ell^\delta(\boldsymbol{\delta}^*) \geq \bar{\ell}$. This requirement helps RVM to return quickly to a region in which the approximation is bounded. However, if the step size falls below the threshold used to detect discontinuities, we may relax this constraint so that less time must be spent to detect potential discontinuities. (3) If we are outside the admissible region, i.e. $\bar{\ell} < \ell^*$, we enforce that we get closer to the target likelihood: $\left|\ell^\delta(\boldsymbol{\delta}^*) - \ell^*\right| < \left|\bar{\ell} - \ell^*\right|$. This reduces potential convergence issues. (4) We require that

$$\frac{\left|\hat{\ell}^\delta(\boldsymbol{\delta}^*) - \ell^\delta(\boldsymbol{\delta}^*)\right|}{\left|\bar{\ell} - \ell^*\right|} \quad \leq \quad \gamma \tag{5.17}$$

for a constant γ. That is, the required precision depends on how close we are to the target. This facilitates fast convergence of the algorithm. The constant $\gamma \in (0,1)$ controls how strict the precision requirement is. In tests, $\gamma = \frac{1}{2}$ appeared to be a good choice. (5) If we are close to the target, $\ell^\delta(\boldsymbol{\delta}^*) \approx \ell^*$, we also require that the gradient estimate is precise:

$$\frac{\left|\frac{\partial \hat{\ell}^\delta}{\partial \tilde{\theta}}(\boldsymbol{\delta}^*) - \frac{\partial \ell^\delta}{\partial \theta}(\boldsymbol{\delta}^*)\right|}{|\boldsymbol{g}|} \quad \leq \quad \gamma. \tag{5.18}$$

This constraint helps us to get closer to a maximum in the nuisance parameters. Here, we use the $\mathcal{L}_2$ norm.

When we reject a step because the approximation is not sufficiently accurate, we adjust δ_0^* and solve the constrained maximization problem (5.9) requiring $\left|\widetilde{\boldsymbol{\delta}}\right| \leq r$. To ensure that the resulting step does not push the log-likelihood below the target ℓ^*, the radius r should not be decreased more strongly than δ_0^*. In tests, adjusting r by a factor $\beta_1 := 1.5$ whenever δ_0^* is adjusted by factor $\beta_0 := 2$ appeared to be a good choice.

5.2.9 Confidence intervals for functions of parameters

Often, modellers are interested in confidence intervals for functions $f(\boldsymbol{\theta})$ of the parameters. A limitation of VM and VMR is that such confidence intervals cannot be computed directly with these algorithms. However, this problem can be solved approximately by considering a slightly changed likelihood function. We aim to find

$$\phi^{\max} \quad = \quad \max_{\boldsymbol{\theta} \in \Theta \,:\, \ell(\boldsymbol{\theta}) \geq \ell^*} f(\boldsymbol{\theta}) \tag{5.19}$$

or the respective minimum. Define

$$\check{\ell}(\phi, \boldsymbol{\theta}) \quad := \quad \ell(\boldsymbol{\theta}) - \frac{1}{2}\left(\frac{f(\boldsymbol{\theta}) - \phi}{\varepsilon}\right)^2 \chi^2_{1,1-\alpha}, \tag{5.20}$$

with a small constant ε. Consider the altered maximization problem

$$\check{\phi}^{\max} \quad = \quad \max_{\boldsymbol{\theta}\in\Theta\,:\,\check{\ell}(\phi,\boldsymbol{\theta})\geq\ell^*} \phi, \tag{5.21}$$

which can be solved with VM or RVM.

We argue that a solution to (5.21) is an approximate solution to (5.19), whereby the error is bounded by ε. Let $(\phi^{\max}, \boldsymbol{\theta}^*)$ be a solution to problem (5.19) and $\left(\check{\phi}^{\max}, \check{\boldsymbol{\theta}}^*\right)$ a solution to problem (5.21). Since $\phi^{\max} = f(\boldsymbol{\theta}^*)$, it is $\check{\ell}(\phi^{\max}, \boldsymbol{\theta}^*) = \ell(\boldsymbol{\theta}^*) \geq \ell^*$. Therefore, $(\phi^{\max}, \boldsymbol{\theta}^*)$ is also a feasible solution to (5.19), and it follows that $\check{\phi}^{\max} \geq \phi^{\max}$. At the same time, $\check{\ell}(\phi, \boldsymbol{\theta}) \leq \ell(\boldsymbol{\theta})$, which implies that $f\left(\check{\boldsymbol{\theta}}^*\right) \leq f(\boldsymbol{\theta}^*)$, since $\boldsymbol{\theta}^*$ maximizes f over a domain larger than the feasibility domain of (5.21). In conclusion, $f\left(\check{\boldsymbol{\theta}}^*\right) \leq f(\boldsymbol{\theta}^*) = \phi^{\max} \leq \check{\phi}^{\max}$. Lastly,

$$\ell^* \;=\; \ell\left(\hat{\boldsymbol{\theta}}\right) - \frac{1}{2}\chi^2_{1,1-\alpha} \;\leq\; \check{\ell}\left(\check{\phi}^{\max}, \check{\boldsymbol{\theta}}^*\right) \;=\; \ell\left(\check{\boldsymbol{\theta}}^*\right) - \frac{1}{2}\left(\frac{f\left(\check{\boldsymbol{\theta}}^*\right) - \check{\phi}^{\max}}{\varepsilon}\right)^2 \chi^2_{1,1-\alpha}. \tag{5.22}$$

Simplifying (5.22) yields $\left|f\left(\check{\boldsymbol{\theta}}^*\right) - \check{\phi}^{\max}\right| \leq \varepsilon$. Thus, $\left|\phi^{\max} - \check{\phi}^{\max}\right| \leq \varepsilon$.

Though it is possible to bound the error by an arbitrarily small constant ε in theory, care must be taken if the function $f(\boldsymbol{\theta})$ is not well-behaved, i.e. strongly nonlinear. In theses cases, overly small values for ε may slow down convergence.

Note that the suggested procedure may seem to resemble the approach of Neale and Miller (1997), who also account for constraints by adding the squared error to the target function. However, unlike Neale and Miller (1997), the approach suggested above bounds the error in the confidence interval bound, not the error of the constraint. Furthermore, we do not square the log-likelihood function, which would worsen nonlinearities and could thus make optimization difficult. Therefore, our approach is less error-prone than the method by Neale and Miller (1997).

5.3 Tests

To compare the presented algorithm to existing methods, we applied RVM, the classic VM, and five other algorithms to benchmark problems and compared the robustness and performance of the approaches. Below we review the implemented methods. Then we introduce the benchmark problems, before we finally present the benchmark results.

5.3.1 Methods implemented for comparison

Besides RVM and VM, we implemented three methods that repeatedly evaluate the profile likelihood function and two methods that search for the confidence intervals directly. We implemented all methods in the programming language Python version 3.7 and made use of different optimization routines implemented or wrapped in the scientific computing library Scipy (Jones et al., 2001).

First, we implemented a grid search for the confidence bounds. The approach uses repeated Lagrangian constrained optimizations and may resemble the method by DiCiccio and Tibshirani (1991); however, rather than implementing the algorithm by DiCiccio and Tibshirani (1991), we applied the constrained optimization algorithm by Lalee et al. (1998), which is a trust-region approach and may thus be more robust than the method by DiCiccio and Tibshirani (1991). Furthermore, the algorithm by Lalee et al. (1998) was readily implemented in Scipy.

We conducted the grid search with a naive step size of 0.2, which we repeatedly reduced by factor 2 close to the threshold log-likelihood ℓ^* until the desired precision was achieved. To account for unidentifiable parameters, we attempted one large step (1000 units) if the algorithm did not terminate in the given iteration limit. We considered a parameter as unbounded if this step yielded a log-likelihood above the target value ℓ^*.

Second, we implemented a quadratic bisection method for root finding on ℓ_{PL} (cf. Ren and Xia, 2019). Initially we chose a step size of 1. Afterwards, we computed the step of

θ_0 based on a quadratic interpolation between the MLE $\hat{\theta}_0$, the maximal value of θ_0 for which we found $\ell_{\text{PL}}(\theta_0) > \ell^*$ and the smallest identified value of θ_0 with $\ell_{\text{PL}}(\theta_0) < \ell^*$. Until a point θ_0 with $\ell_{\text{PL}}(\theta_0) < \ell^*$ was identified, we interpolated ℓ_{PL} between $\hat{\theta}_0$ and the two largest evaluated values θ_0. When only two points were available or the approximation of ℓ_{PL} did not assume the target value, we introduced the additional constraint $\frac{d\ell_{\text{PL}}}{d\theta_0} = 0$. Using a quadratic rather than a linear interpolation for bisection has the advantage that the algorithm converges faster if the profile log-likelihood function is convex or quadratic. To evaluate ℓ_{PL}, we applied sequential least squares programming (Kraft, 1988), which is the default method for constrained optimization in Scipy.

Third, we implemented a binary search with an initial step of 1. Until a value θ_0 with $\ell_{\text{PL}}(\theta_0) < \ell^*$ was found, we increased θ_0 by factor 10. This preserves the logarithmic runtime of the algorithm if the problem has a solution. To broaden the range of tested internal optimization routines, we used a different method to evaluate ℓ_{PL} than in the bisection method: we fixed θ_0 at the desired value and performed an unconstrained optimization on the nuisance parameters. Here, we used the quasi-Newton method by Broyden, Fletcher, Goldfarb, and Shanno (BFGS; see Nocedal and Wright, 2006, pp. 136).

To test methods that search for the confidence interval end points directly, we solved problem (5.4) with sequential least squares programming (Kraft, 1988). Furthermore, we implemented the approximate method by Neale and Miller (1997). They transform the constrained maximization problem (5.9) to an unconstrained problem by considering the sum of the parameter of interest θ_0 and the squared error between the target ℓ^* and the log-likelihood. Minimization of this target function yields a point in which the target log-likelihood is reached approximately and the parameter of interest is minimal. Again, we used the method BFGS for minimization (see above).

Finally, we implemented Wald's method to assess the need to apply any profile likelihood method.

5.3.2 Benchmark problem

To investigate the performances of the implemented methods, we applied the algorithms to a benchmark problem with variable parameter number and data set size. We considered a logistic regression problem with n count data covariates c_{ij}, $j \in \{1, \ldots, n\}$ for each data point $i \in \{1, \ldots, N\}$. We assumed that the impact of the covariates levels off at high values and considered therefore the transformed covariates $c_{ij}^{\alpha_j}$ with $\alpha \in (0, 1)$. This is not only reasonable in many real world problems but also makes likelihood maximization a computationally challenging problem if not enough data are available to achieve asymptotic normality of the MLE. Hence, this scenario gives insights into the performance of the implemented methods in challenging realistic problems. The benchmark model's probability mass function for a data point X_i was thus given by

$$\mathbb{P}(X_i = 1) \quad = \quad \left(1 + \exp\left(-\beta_0 - \sum_j \beta_j c_{ij}^{\alpha_j}\right)\right)^{-1} \tag{5.23}$$

and $\mathbb{P}(X_i = 0) = 1 - \mathbb{P}(X_i = 1)$.

We drew the covariate values randomly from a negative binomial distribution with mean 5 and variance 10. The negative binomial distribution is commonly used to model count data (Gardner et al., 1995) and thus suited to represent count covariates. To simulate the common case that covariates are correlated, we furthermore drew the value for every other covariate from a binomial distribution with the respective preceding covariate as count parameter. That is, for uneven j,

$$c_{i,j+1} \quad \sim \quad \text{Binomial}(c_{i,j}, p),$$

with $p = 0.2$ in our simulations. To avoid numerical problems arising when covariates with value 0 are raised to the power 0, we added a small positive perturbation to the count values. That way, we achieved that 0^0 was defined to be 1. We chose the parameters α_j and β_j so

that the data were balanced, i.e. the frequency of 0s and 1s was approximately even. Refer to Appendix 5.B for the parameter values we used.

5.3.3 Test procedure

To test the algorithms in a broad range of scenarios and assess how their performance is impacted by model characteristics, we considered a model with 1 covariate (3 parameters), a model with 5 covariates (11 parameters), and a generalized linear model (GLM) with 10 covariates, in which the powers α_j were set to 1 (11 parameters). Furthermore, we varied the sizes of the simulated data sets, ranging between $N = 500$ and $N = 10000$ for the models with transformed covariates and $N = 50$ and $N = 1000$ for the GLM. In Figure 5.3, we depict the impact of N on the shape of the likelihood function and thus the difficulty of the problem.

For each considered set of parameters, we generated 200 realizations of covariates and training data from the model described in the previous section. We determined the maximum likelihood estimator by maximizing the log-likelihood with the method BFGS and refined the estimate with an exact trust region optimizer (Conn et al., 2000). Then, we applied each of the implemented algorithms to each data set and determined the algorithms' success rates and efficiencies.

As the likelihood functions of the tested models decrease drastically at $\alpha_j = 0$, potentially causing some algorithms to fail, we constrained the α_j to non-negative values. Tho that end, we considered transformed parameters $\alpha'_j := \ln(\exp(\alpha_j) - 1)$. Such transformations are reasonable whenever the parameter range is naturally constrained from a modelling perspective. Nonetheless, we evaluated the results of the tested algorithms based on the back-transformed parameters α_j.

We measured the algorithms' success based on their ability to solve problem (5.4) rather than their capability to determine the true confidence intervals for the parameters. Though profile likelihood confidence intervals are usually highly accurate, they rely on the limit-

ing distribution of the likelihood ratio statistic. Therefore, algorithms could fail to solve optimization problem (5.4) but, by coincidence, return a result close to the true confidence interval bound and vice versa. To exclude such effects and circumvent the high computational effort required to determine highly precise confidence intervals with sampling methods, we determined the "true" confidence interval bound by choosing the widest confidence interval bound obtained by either of the tested methods provided it was admissible, i.e. $\ell(\theta^{\max}) \geq \ell^*$ up to a permissible error of 0.001.

We considered an algorithm successful if (1) the returned result was within a $\pm 5\%$ range of the true confidence interval bound or had an error below 0.001, and (2) the algorithm reported convergence. That is, to be deemed successful, an algorithm had to both return the correct result and also claim that it found the correct solution. The latter constraint ensures that if none of the algorithms converges successfully, even the one with the best result is not considered successful.

As many of the tested methods rely on general optimizers without specific routines to identify situations with divergent solutions, we considered parameters with confidence interval bounds exceeding $[-1000, 1000]$ in the transformed parameter space as unbounded. Consequently, all algorithms returning a larger confidence interval were considered successful.

We limited the runtime of all methods except the pre-implemented optimizers by introducing a step limit of 200. If convergence was not reached within this number of steps, the algorithms were viewed unsuccessful except for the case with inestimable parameters.

To test whether some methods tend return misleading results, we determined the mean absolute error between the returned and the true confidence interval bounds when algorithms reported success. As this quantity can be dominated by outliers, we also determined the mean of all errors below 10 and the frequency of errors beyond 10.

We measured the computational speed of the different methods by recording the number of function evaluations required until termination. This provides us with precise benchmark

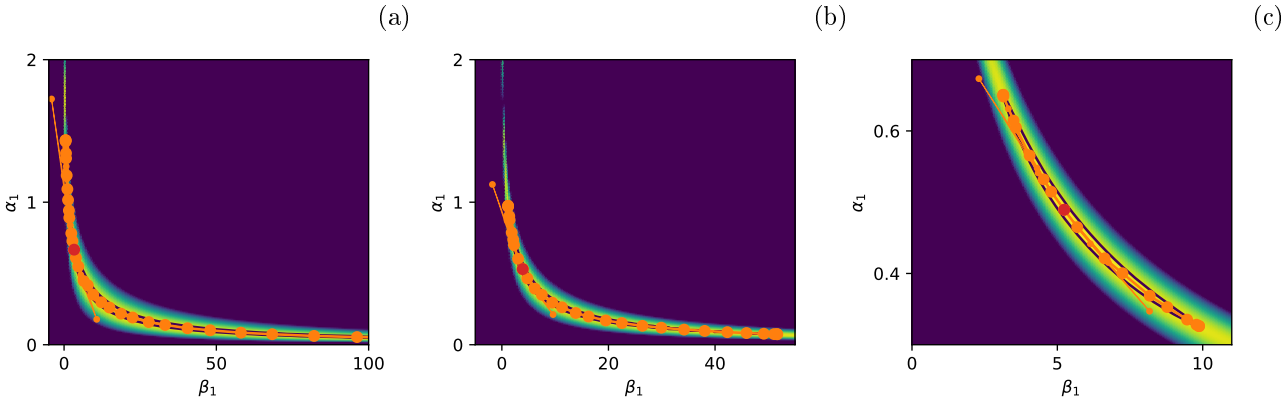


Figure 5.3: Likelihood surface of the 3-parameter benchmark model with different data set sizes N. As N increases, the confidence region becomes smaller and closer to an elliptic shape. The orange dots depict the accepted (large dots) and rejected (small dots) steps of RVM searching for a confidence interval for β_1. RVM follows the ridge of the likelihood surface. The red dot shows the location of the MLE $\hat{\theta}$. The background colour depicts the respective maximal log-likelihood for the given α_1 and β_1 ranging from $\leq \hat{\ell} - 50$ (dark blue) to $\hat{\ell}$ (yellow). The solid blue line denotes the target log-likelihood ℓ^* for a 95% confidence interval. (a) $N = 500$; (b) $N = 1000$; (c) $N = 10000$.

results independent of hardware and implementation details. To display a potential trade-off between robustness (success rate) and speed (number of function evaluations), we did not consider cases in which convergence was not reached. That way, internal stopping criteria did not affect the results.

The specific advantage of some optimization algorithms is in not requiring knowledge of the Hessian matrix. As computing the Hessian is necessary for RVM and may reduce the algorithm's performance compared to other methods, we included the number of function evaluations required to determine the Hessian and the gradient in the recorded count of function evaluations. We computed gradients and Hessian matrices with a complex step method (Lai et al., 2005) implemented in the Python package numdifftools (Brodtkorb and D'Errico, 2019).

5.3.4 Results

To get an impression of how RVM acts in practice, we plotted the trajectory of RVM along with ancillary function evaluations in Figure 5.3. It is visible that the algorithm stays on the

"ridge" of the likelihood surface even if the admissible region is strongly curved. This makes RVM efficient.

In fact, for all considered quality measures, RVM yielded good and often the best results compared to the alternative methods (see Figure A5.1). In all considered scenarios, RVM was the algorithm with the highest success rate, which never fell below 90% (second best: binary search, 52%). In scenarios with small data sets, the success rate of RVM was up to 37 percent points higher than any other method. At the same time, RVM was among the fastest algorithms. In scenarios with large data sets, RVM often converged within three iterations. Furthermore, RVM was quick in the 3 parameter model, in which the Hessian matrix is easy to compute. In the scenario with transformed covariates and 11 parameters, RVM required about three times as many likelihood evaluations as the fastest algorithm but had a more than 56% higher success rate. The error in the results returned by RVM was consistently low compared to other methods. The proportion of large errors was always below 1%, and the mean error excluding these outliers never exceeded 0.05.

The algorithms that require repeated evaluations of the profile likelihood function performed second best in terms of the success rate. Except for the GLM with 50 data points, the binary search, the grid search, and the bisection method consistently had success rates above 70%, whereby the success rate increased with the size of the considered data set. However, these algorithms also required more function evaluations than other methods. In fact, the grid search was more than 5 times slower than any other algorithm. The binary search was slightly less efficient than the bisection method, which exploits the approximately quadratic shape of the profile likelihood function if many data are available. In scenarios with large data sets, the bisection method was among the most efficient algorithms. The errors of the three root finding methods decreased the more data became available to fit the models. However, while the binary search had a consistently low error, both the grid search and the bisection method were more prone to large errors than all other tested methods.

The algorithms developed from the constrained maximization perspective (the method by Neale and Miller and direct constrained maximization) had success rates ranging between 45% and 85% in problems with transformed covariates. In the GLM scenario, the success rate was smaller in with 50 data points and higher with more data. The constrained maximization procedure was slightly more successful than the method by Neale and Miller (1997). Both methods required relatively few function evaluations, whereby direct constrained maximization performed better. Both methods were less prone to large errors than the grid search and the bisection method. However, the outlier-reduced error was on average more than twice as large than with any other method except RVM (Neale and Miller: 0.16, constrained maximum 0.09, RVM: 0.07).

The success of the algorithm VM depended highly on the properties of the likelihood function. In scenarios with few data and transformed covariates, VM had very low success rates (as low as 10%). When more data were added, VM became as successful as the method by Neale and Miller and direct constrained maximization. Thereby, VM was highly efficient whenever results were obtained successfully. Similar to the success rate, the mean error of VM decreased strongly as more data were considered.

Wald's method had very low success rates and large errors except for the GLM with large data sets. In the models with transformed covariates, Wald's method never had a success rate above 17%.

5.4 Discussion

We presented an algorithm that determines the end points of profile likelihood confidence intervals both of parameters and functions of parameters with high robustness and efficiency. We tested the algorithm in scenarios varying in parameter number, size of the data set, and complexity of the likelihood function. In the tests, our algorithm RVM was more robust than any other considered method. At the same time, RVM was among the fastest algorithms in most scenarios. This is remarkable, because there is typically a trade-off between robustness

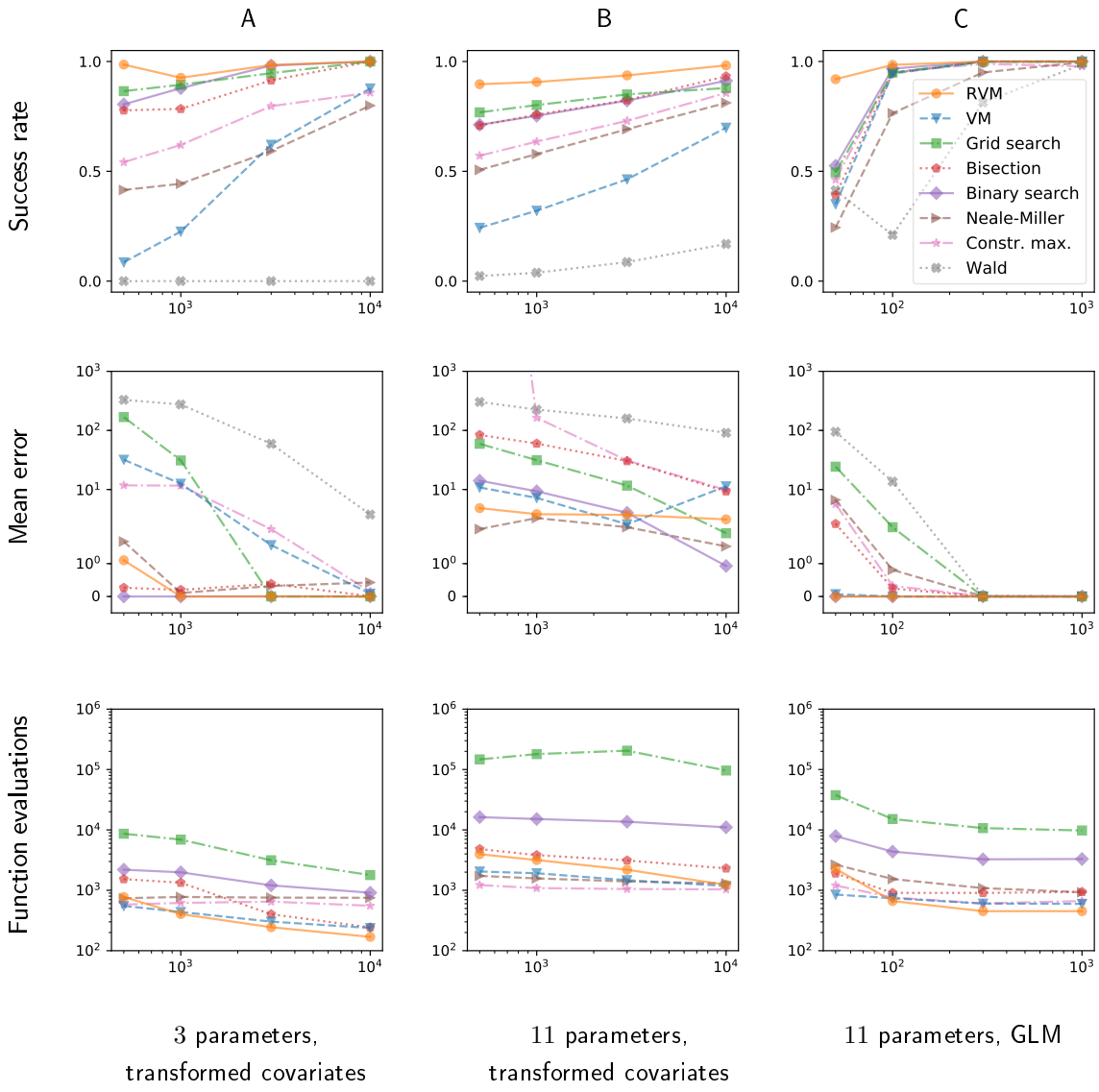


Figure 5.4: Benchmark results. The success rate, the mean error, and the number of function evaluations are plotted for the 3 parameter and the 11 parameter model with transformed covariates and for the 11 parameter GLM. Throughout the simulations, our algorithm RVM had the highest success rate. At the same time, RVM had a low mean error and required only few likelihood function evaluations compared to the considered alternative methods. The parameter values used to generate the Figures are given in Appendix 5.B.

and computational speed of optimization algorithms. RVM achieves this result by exploiting the approximately quadratic form of the log-likelihood surface in "benign" cases while maintaining high robustness with the trust-region approach. Consequently, RVM naturally extends the algorithm VM (Venzon and Moolgavkar, 1988), which appeared to be highly efficient but lacking robustness in our tests.

Surprisingly, RVM turned out to be even more robust than methods based on repeated evaluations of the profile likelihood. For the bisection method and the binary search, this may be due to failures of internal optimization routines, as initial guesses far from the solution can hinder accurate convergence. The grid search method, in turn, was often aborted due to the limited step size, which precluded the method from identifying confidence bounds farther than 40 units away from the respective MLE. This, however, does not explain the comparatively high error in the results of the grid search, as only successful runs were considered. We therefore hypothesize that internal optimization issues were responsible for some failures.

As expected, the algorithms that searched for the confidence interval end points directly were more efficient but less robust than algorithms that repeatedly evaluate the profile likelihood. Remarkably, a "standard" algorithm for constrained optimization performed slightly better than an unconstrained optimizer operating on the modified target function suggested by Neale and Miller (1997). This indicates that the approximation introduced by Neale and Miller (1997) might not be necessary and even of disadvantage.

All methods implemented in this study (except RVM and VM) rely on general optimizers. Consequently, the performance of these methods depends on the chosen optimizers both in terms of computational speed and robustness. Careful adjustment of optimization parameters might make some of the implemented algorithms more efficient and thus more competitive in benchmark tests. Though we attempted to reduce potential bias by applying a variety of different methods, an exhaustive test of optimization routines was beyond the scope of this study. Nonetheless, the consistently good performance of RVM throughout our benchmark tests suggests that RVM is a good choice in many applications.

Though RVM performed well in our tests, there are instances in which the algorithm is not applicable or sufficiently efficient. This are scenarios in which (1) the log-likelihood cannot be computed directly, (2) the Hessian matrix of the log-likelihood function is hard to compute, (3) the dimension of the parameter space is very large, or (4) there are multiple points in the parameter space in which problem (5.4) is solved locally. Below, we briefly discuss each of these limitations.

(1) In hierarchical models, the likelihood function may not be known. As RVM needs to evaluate the log-likelihood, its gradient, and its Hessian matrix, the algorithm is not applicable in these instances. Consequently, sampling based methods, such as parametric bootstrap (Efron, 1981), Monte Carlo methods (Buckland, 1984), or data cloning (Ponciano et al., 2009) may then be the only feasible method to determine confidence intervals.

(2) Especially in problems with a large parameter space, it is computationally expensive to compute the Hessian matrix with finite difference methods, as the number of function calls increases in quadratic order with the length of the parameter vector. Though alternative differentiation methods, such as analytical or automatic differentiation (Griewank, 1989), are often applicable, there may be some instances in which finite difference methods are the only feasible alternative. In these scenarios, optimization routines that do not require knowledge of the Hessian matrix may be faster than RVM. Note, however, that the higher computational speed may come with decreased robustness, and sampling based methods might be the only remaining option if application of RVM is infeasible.

(3) If the parameter space has a very high dimension (exceeding 1000), internal routines, such as inversion of the Hessian matrix, may become the dominant factor determining the speed of RVM. Though it may be possible in future to make RVM more efficient, sampling based methods or algorithms that do not use the Hessian matrix may be better suited in these scenarios.

(4) RVM as well as all other methods implemented in this study are local optimization algorithms. Therefore, the algorithms may converge to wrong results if maximization prob-

lem (5.4) has multiple local solutions. This is in particular the case if the confidence set $\{\theta_0 : \ell_{\mathrm{PL}}(\theta_0) \geq \ell^*\}$ is not connected and thus no interval. RVM reduces the issue of local extreme points by choosing steps carefully and ensuring that the point of convergence is indeed a maximum. This contrasts with VM, which could converge to the wrong confidence interval end point (e.g. maximum instead of minimum) if the initial guesses are not chosen with care. Nonetheless, stochastic optimization routines, such as genetic algorithms (Akrami et al., 2010), and sampling methods may be better suited if a local search is insufficient.

Despite these caveats, RVM is applicable to a broad class of systems. Especially when inestimable parameters are present, commonly used methods such as VM or grid search techniques can break down or be highly inefficient. Furthermore, optimization failures are commonly observed if not enough data are available to reach the asymptotic properties of the MLE (Ren and Xia, 2019). RVM is a particularly valuable tool in these instances.

5.5 Conclusion

We developed and presented an algorithm to determine profile likelihood confidence intervals. In contrast to many earlier methods, our algorithm is robust in scenarios in which lack of data or a complicated likelihood function make it difficult to find the bounds of profile likelihood confidence intervals. In particular, our methods is applicable in instances in which parameters are not estimable and in cases in which the likelihood function has strong nonlinearities. At the same time, our method efficiently exploits the asymptotic properties of the maximum likelihood estimator if enough data are available.

We tested our method on benchmark problems with different difficulty. Throughout our simulations, our method was the most robust while also being amongst the fastest algorithms. We therefore believe that RVM can be helpful to researchers and modellers across disciplines.

Appendices

5.A An alternative way to account for singular matrices

In each iteration, we seek to maximize the approximate likelihood $\hat{\ell}$ with respect to the nuisance parameters. To that end, we solve the equation

$$0 \;=\; \frac{\partial}{\partial \widetilde{\boldsymbol{\delta}}} \hat{\ell}^{\boldsymbol{\delta}} \;=\; \underline{\widetilde{\mathrm{H}}} \widetilde{\boldsymbol{\delta}}^{*} + \widetilde{\mathbf{H}}_0 \delta_0^{*} + \widetilde{\boldsymbol{g}} \tag{A5.1}$$

which has a unique solution if and only if $\underline{\widetilde{\mathrm{H}}}$ is invertible. Otherwise, equation (A5.1) may have infinitely or no solutions. In the main text, we suggested to solve (A5.1) with the Moore-Penrose inverse if $\underline{\widetilde{\mathrm{H}}}$ is singular. However, this procedure appeared to be very sensitive to a threshold parameter in tests, and we obtained better results with an alternative method, which we describe below. We furthermore show test results comparing the two methods.

5.A.1 Description of the method

If $\hat{\ell}$ has infinitely many maxima in the nuisance parameters, we can choose some nuisance parameters freely and consider a reduced system including the remaining independent parameters only. To that end, we check $\underline{\widetilde{\mathrm{H}}}$ for linear dependencies at the beginning of each iteration. We are interested in a minimal set S containing indices of rows and columns whose removal from $\underline{\widetilde{\mathrm{H}}}$ would make the matrix invertible. To compute S, we iteratively determine the ranks of sub-matrices of $\underline{\widetilde{\mathrm{H}}}$ using singular value decompositions (SVMs). SVMs are a well-known tool to identify the rank of a matrix.

The iterative algorithm proceeds as follows: first, we consider one row of $\underline{\widetilde{\mathrm{H}}}$ and determine its rank. Then, we continue by adding a second row, determine the rank of the new matrix and repeat the procedure until all rows, i.e. the full matrix $\underline{\widetilde{\mathrm{H}}}$, are considered. Whenever

the matrix rank increases with addition of a row, this row is linearly independent from the previous rows. Conversely, the rows that do not increase the matrix rank are linearly dependent on other rows of $\widetilde{\underline{\mathrm{H}}}$. The indices of these rows form the set S.

In general, the set of linearly dependent rows is not unique. Therefore, we consider the rows of $\widetilde{\underline{\mathrm{H}}}$ in descending order of the magnitudes of the corresponding gradient entries. This can help the algorithm to converge faster.

After S is determined, we need to check whether there is a parameter vector $\boldsymbol{\theta}^*$ satisfying requirements 1 and 2 from section 5.2.1 for the approximation $\hat{\ell}$. Let $\widetilde{\underline{\mathrm{H}}}_{\mathrm{dd}}$ ("d" for "dependent") be the submatrix of $\underline{\mathrm{H}}$ that remains if all rows and columns corresponding to indices in S are removed from $\widetilde{\underline{\mathrm{H}}}$. Similarly, let $\widetilde{\underline{\mathrm{H}}}_{\mathrm{ff}}$ ("f" for "free") be the submatrix of $\widetilde{\underline{\mathrm{H}}}$ containing only the rows and columns corresponding to indices in S, and let $\widetilde{\underline{\mathrm{H}}}_{\mathrm{df}} = \widetilde{\underline{\mathrm{H}}}_{\mathrm{fd}}^{\top}$ be the matrix containing the rows whose indices are *not* in S and the columns whose indices *are* in S. Let us define $\widetilde{\boldsymbol{g}}_{\mathrm{d}}$, $\widetilde{\boldsymbol{g}}_{\mathrm{f}}$, $\widetilde{\boldsymbol{\delta}}_{\mathrm{d}}$, and $\widetilde{\boldsymbol{\delta}}_{\mathrm{f}}$ accordingly. If $\widetilde{\underline{\mathrm{H}}}_{\mathrm{dd}}$ is not negative definite, $\hat{\ell}$ is unbounded, and requirement 2 cannot be satisfied. Otherwise, we may attempt to solve

$$
\begin{aligned}
0 &= \frac{\partial}{\partial \widetilde{\boldsymbol{\delta}}} \hat{\ell}^{\boldsymbol{\delta}} && \text{(A5.2)} \\
\iff & \\
0 &= \widetilde{\underline{\mathrm{H}}}_{\mathrm{dd}} \widetilde{\boldsymbol{\delta}}_{\mathrm{d}}^{*} + \widetilde{\underline{\mathrm{H}}}_{\mathrm{df}} \widetilde{\boldsymbol{\delta}}_{\mathrm{f}}^{*} + \widetilde{\mathbf{H}}_{0\mathrm{d}} \delta_0^{*} + \widetilde{\boldsymbol{g}}_{\mathrm{d}} && \text{(A5.3)} \\
0 &= \widetilde{\underline{\mathrm{H}}}_{\mathrm{df}}^{\top} \widetilde{\boldsymbol{\delta}}_{\mathrm{d}}^{*} + \widetilde{\underline{\mathrm{H}}}_{\mathrm{ff}} \widetilde{\boldsymbol{\delta}}_{\mathrm{f}}^{*} + \widetilde{\mathbf{H}}_{0\mathrm{f}} \delta_0^{*} + \widetilde{\boldsymbol{g}}_{\mathrm{f}}. && \text{(A5.4)}
\end{aligned}
$$

If equation system (A5.3)-(A5.4) has a solution, we can choose $\widetilde{\boldsymbol{\delta}}_{\mathrm{f}}^{*}$ freely. Setting $\widetilde{\boldsymbol{\delta}}_{\mathrm{f}}^{*} \leftarrow \mathbf{0}$ makes equation (A5.3) equivalent to

$$
\widetilde{\boldsymbol{\delta}}_{d}^{*} = -\widetilde{\underline{\mathrm{H}}}_{\mathrm{dd}}^{-1} \left(\widetilde{\underline{\mathrm{H}}}_{0\mathrm{d}} \delta_0 + \widetilde{\boldsymbol{g}}_{\mathrm{d}} \right). \qquad \text{(A5.5)}
$$

That is, we may set $\widetilde{\underline{\mathrm{H}}} \leftarrow \widetilde{\underline{\mathrm{H}}}_{\mathrm{dd}}$, $\widetilde{g} \leftarrow \widetilde{g}_{\mathrm{d}}$, $\widetilde{\boldsymbol{\delta}}^{*} \leftarrow \widetilde{\boldsymbol{\delta}}_{\mathrm{d}}^{*}$ for the remainder of the current iteration and proceed as usual, whereby the free nuisance parameters are left unchanged: $\widetilde{\boldsymbol{\delta}}_{\mathrm{f}}^{*} = \mathbf{0}$. With

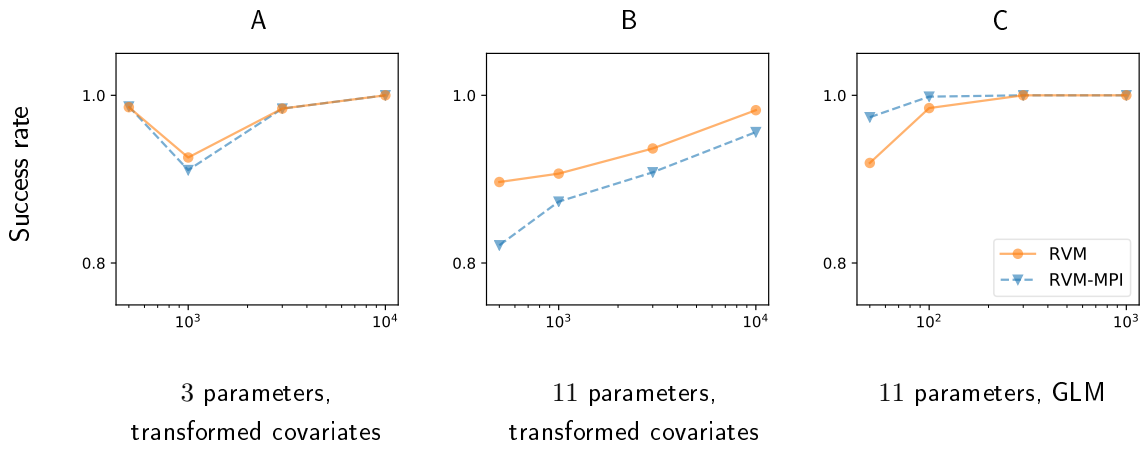


Figure A5.1: Comparison of different methods to handle linearly dependent parameters. The success rate of RVM is plotted for the 3 parameter and the 11 parameter model with transformed covariates and for the 11 parameter GLM. Though the algorithm using the Moore-Penrose inverse (RVM-MPI) performed slightly better for the GLM with little data (panel C), the method had lower success rates when the 11 parameter model with transformed variables was considered (panel B). The parameter values used to generate the Figures are given in Appendix 5.B.

the resulting δ_0^*, we check whether (A5.4) holds approximately. If not, the log-likelihood is unbounded above. We consider this case in section 5.2.4 in the main text.

5.A.2 Tests

We implemented RVM with both suggested methods for treating linearly dependent parameters. To that end, we applied the same testing procedure described in section 5.3 of the main text. The two methods yielded similar results in terms of computational speed (number of required likelihood evaluations) and error in case of reported success (see section 5.3.1 in the main text). However, holding some parameters constant as suggested in this Appendix turned out to be more robust in general and lead to slightly higher success rates (see Figure A5.1). Therefore, we suggest using this method in practice.

5.B Parameters for benchmark tests

Here we provide the parameter values used to generate the data for our benchmark tests. We tested models with transformed covariates with 3 and 11 parameters and a GLM with

11 parameters. For the models with transformed covariates, we considered scenarios with $N = 500$, $N = 1000$, $N = 3000$, and $N = 10000$ data points. The parameters for the two model families are given in Tables A5.1 and A5.2. For the GLM, we considered data sets with sizes $N = 50$, $N = 100$, $N = 300$, and $N = 1000$. We provide the parameter values in Table A5.3.

Parameter	α_1	β_0	β_1
Value	0.5	-10	5

Table A5.1: Parameters for the model with 3 parameters and transformed covariates.

Parameter	α_1	α_2	α_3	α_4	α_5	β_0	β_1	β_2	β_3	β_4	β_5
Value	0.2	1	0.1	0.2	0.5	-1	5	2	-1	-3	-2

Table A5.2: Parameters for the model with 11 parameters and transformed covariates.

Parameter	β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9	β_{10}
Value	0.8	0.2	-0.6	-1	-1	0.2	0.5	0.1	-0.2	0.2	2

Table A5.3: Parameters for the 11 parameter GLM. The covariate powers α_i are all fixed at 1.

Chapter 6

Synthesis and discussion

In this thesis, I have introduced a set of tools suitable to enhance invasive species and disease management. First, I have developed an algorithm to compute locally optimal routes for route choice networks. This method facilitates comprehensive traffic models. Second, I have described a hybrid approach to model the traffic of invasive species and disease vectors. The hybrid model can be fitted with data from road-side traffic surveys and admits a new quality of inference on the spread of infectious diseases and invasive species. Third, I have developed a decision support tool that builds on the traffic estimates from the hybrid model and yields detailed management advice. Fourth, I have presented a new algorithm to compute confidence intervals. The algorithm is robust end efficient even in the presence of inestimable parameters.

I have applied all these methods to a management scenario seeking to prevent the introduction of zebra and quagga mussels to British Columbia. Evaluating the results, I have drawn general conclusions on optimal management. All presented methods are scalable by design and applicable in a broad range of scenarios. Consequently, this thesis contributes to a variety of research areas.

Our understanding of the spread and management of invasive species and diseases is increasing constantly, and the number of scientific studies on the topics have seen an exponential increase within the past decades (Lockwood et al., 2013). Consequently, this thesis is embedded in a continuum of research, building on earlier methods and results, extending

these, and – hopefully – serving as a basis for later studies increasing our understanding further. In this chapter, I discuss the methods and results presented in this thesis in the context of earlier findings. Moreover, I look further down the road at promising possibilities to extend the presented approaches and deepen our understanding of the spread and management of invasive species and diseases.

6.1 Invasive species and disease modelling

6.1.1 Advancements in invasive species and disease modelling

Models play an integral role for understanding the spread of invasive species and diseases (Lewis et al., 2016). Among others, models can be used to extrapolate data from specific locations and times to draw inference on the full state of a system and to predict future developments under scenarios of interest. Consequently, models are an important tool to inform science and management. This motivates the development of ever new, improved, and more accurate models. In this thesis, I presented methods extending the accuracy and range of applicability of gravity models.

Gravity models have been frequently applied to model the spread of invasive species (Bossenbroek et al., 2001, 2007; Potapov et al., 2010; Muirhead and MacIsaac, 2011) and infectious diseases (Stijns, 2003; Xia et al., 2004; Ferrari et al., 2006; Li et al., 2011; Tuite et al., 2011). Though gravity models are often introduced as phenomenological models, they have been justified mechanistically in economic contexts (Anderson, 2011), and further realism can be added by introducing constraints (Wilson, 1970) and accounting for stochastic processes (Flowerdew and Aitkin, 1982). Gravity models can incorporate a variety of covariates and thus model many mechanisms impacting travellers' decisions. This facilitates the accuracy of predictions, makes gravity models adjustable, and also admits scientific inference via hypothesis testing (in nested models) or multiple working hypotheses (Anderson et al., 2000). Consequently, it is likely that gravity models will remain actively used even though

increasing computational resources allow the application of more mechanistic models (e.g. individual based models, Raney et al., 2003; Doniec et al., 2008) and potentially more precise predictors (e.g. machine learning techniques, Humphries et al., 2018).

In this thesis, I have shown how gravity models can be fitted to data obtained via road-side surveys. Incorporating this new data source allows researchers to build models with higher accuracy and to reduce the need for (untested) modelling assumptions. Furthermore, the new approach makes it possible to fit vector traffic models for large-scale systems, which was difficult or even infeasible before. Since long-distance traffic can be the major force moving an invasion or disease front forward (Kot et al., 1996), the introduced extensions to classic gravity models may allow researchers and managers to gain significant new insights on the progression of epidemics and invasions.

The hybrid approach does not only open a new way to fit gravity models but also yields estimates for vector traffic on roads. These estimates can be used to optimize management actions targeting road traffic, as done in this thesis and discussed later. In addition, traffic estimates can also prove useful if roads are entry points or hubs for infectious diseases or invasive species (cf. Trombulak and Frissell, 2000). For example, road travellers stopping at rest areas may spread diseases to locals or other travellers. Consequently, municipalities close to major highways may have increased infection risk even if they are not attractive final destinations. Similar to the described process of disease transmission, intentionally or unintentionally transported animals may be able to escape into the environment at any point of a trip. The escaped animals could be invasive species or carry an infectious diseases. The hybrid model developed in chapter 3 provides scientists and managers with a new tool to model the spread of such infectious diseases and invasive species.

6.1.2 New opportunities for model validation

Due to the potentially high impact of model predictions on both scientific conclusions and policy, scientists need not only to develop new, improved models but also to assess the quality

of models developed earlier. This way, insufficiently justified conclusions can be identified. Furthermore, model assessment facilitates improvements of existing methods, supports researchers in their choice of modelling tools, and increases their awareness of potential pitfalls. Different studies have evaluated the predictive quality of gravity models in the context of invasive species and disease modelling (Li et al., 2011; Muirhead and MacIsaac, 2011; Rothlisberger and Lodge, 2011), at times with surprising conclusions. For example, Muirhead and MacIsaac (2011) found that constraints added to increase the validity of gravity models decreased their accuracy. Since the hybrid model developed in this thesis can make use of more data than traditional gravity models, the hybrid model can be utilized to test and validate different types of models and assumptions.

When models are fitted to data with statistical methods, modellers typically need to make assumptions about the mechanisms causing the discrepancies between model predictions and data. If a model is deterministic, these discrepancies are usually assumed being due to measurement or sampling error. For stochastic models, modelled stochastic processes are a second possible cause for deviations between data and models.

If data on traffic flows are available, deterministic gravity models are often fitted with least squares methods (Flowerdew and Aitkin, 1982; Bossenbroek et al., 2007). In the realm of maximum likelihood, this is equivalent to assuming that residuals are normally distributed with uniform variance. This assumption is often lacking mechanistic justification. Furthermore, it is unlikely that all processes impacting traffic can be modelled deterministically, and it may be more appropriate to account for unknown impacts by assuming that traffic is random to some degree. For these reasons, *stochastic* gravity models have been developed and are commonly regarded better justified than deterministic models (Flowerdew and Aitkin, 1982; Potapov et al., 2010).

Even stochastic gravity models, nonetheless, face the challenge of identifying the most appropriate stochastic model for the random impacts. Typically, several observations of a random process are necessary to identify its underlying stochastic distribution. Collecting

the data necessary to fit a gravity model, however, is often associated with considerable efforts. Therefore, stochastic gravity models are commonly fitted to a *single* realization of the modelled random process, and the "best fitting" distribution is determined via a model selection criterion. This approach makes it hard to distinguish traffic stochasticity from prediction error and can thus lead to wrong conclusions, as will become apparent shortly.

Count data, such as traffic counts, are often modelled with either a Poisson or a negative binomial distribution (Burger et al., 2009). The negative binomial distribution is overdispersed and thus appropriate if correlations between the counted events are supposed. As a consequence, large differences between predicted mean values and observed counts are more likely in the negative binomial model. Now consider a hypothetical scenario in which we desire to model traffic with a stochastic gravity model. Suppose that the traveller counts are approximately Poisson distributed in reality but that the gravity model is not perfectly suited to account for all the mechanism behind travel choices. Consequently, the deviance between observations and model predictions will be high regardless of what distribution is used. However, since large residuals are more likely under a negative binomial distribution, a gravity model with negative binomial distribution will fit the data (statistically significantly) better than a model with the Poisson distribution. This could have two consequences: we may falsely conclude that traveller counts are indeed negatively binomially distributed, which would suggest that travellers' decisions are correlated. Secondly, the Poisson gravity model may yield more accurate predictions and parameter estimates than the negative binomial model even though the latter fits the data better.

Due to the described issue, overdispersion may have been overestimated and the impact of large residuals not sufficiently taken into account in many studies. However, the problem is not specific to the negative binomial and Poisson distribution and could be even more significant if completely different families of distributions, e.g. zero-inflated distributions (Burger et al., 2009; Muirhead and MacIsaac, 2011), are considered.

A potential solution to this problem is to collect additional traffic samples to infer the distribution of the data. This however, becomes difficult if the parameters of the applied stochastic distribution are assumed to depend on the origins and destinations of travellers. Furthermore, the distribution choice could not be easily integrated in a model selection framework. The hybrid model introduced in this thesis makes use of data that can be resampled multiple times with relatively low effort. As these data are used to fit the complete model, traveller count distributions can be fitted origin and destination dependent, and model selection criteria can be applied.

Though the hybrid model may still face the problem of overestimating the traffic variance if the model fits poorly, the hybrid approach is much less prone to this limitation. For example, the residuals in the model fitted in chapter 3 were larger than predicted by the best-fit model (see Appendix 3.G of chapter 3). Though this indicates that the model did not fit the data overly well (which is a bad news), this limitation was directly apparent, reducing the risk of misjudging the model's accuracy (which is a good news). Along with the hybrid model, I introduced several statistical tools to check model assumptions about the stochastic distributions. Thus, the toolset introduced in this thesis can not only be used to construct models that are less error prone but also to assess the impact of the discussed limitations on the accuracy of "classical" stochastic gravity models.

6.1.3 Future directions

The models developed in this thesis were applied to model a specific stage in the invasion process of a particular set of species. Future work could build upon the presented methods by constructing either a more comprehensive model for the zebra and quagga mussel invasion or a model to investigate the transport and dispersal of *multiple* invasive species.

A comprehensive model for the invasion of zebra and quagga mussels would need to incorporate a submodel for the establishment of new populations. To build a joint model, two major challenges would have to be overcome. First, the density of mussel populations in

donor regions would need to be known to predict the expected number of propagules carried per boat. For example, a boat transported from a highly infested area is more likely to be infested than a boat transported from an area with only few invaded lakes. Unless lake-to-lake traffic is modelled on a continental scale (which might be computationally infeasible), the model would need to aggregate the invasion state of many lakes. Thereby, lake usage would have to be taken into account: a jurisdiction with 1 invaded and 99 uninvaded lakes can be a significant propagule donor if the infested lake is heavily frequented by boaters. Knowing the invasion status and the usage of all lakes across North America could be difficult.

The second challenge would be to develop a model linking the number of arriving propagules to establishment probabilities. Though there are models integrating transport and establishment of zebra mussels, these models may be too simplistic to yield a significant benefit over a pure transport model (Bossenbroek et al., 2001; Mari et al., 2011) or are not not easily portable to a new system (Leung et al., 2004). To fit an establishment model, experimental results or data on past invasions would be necessary. A challenge with experimental results is that establishment is likely dependent on local population densities and that zebra mussels are subject to an Allee effect (Leung et al., 2004). That is, there may be a certain minimal number of mussels required to establish a population. However, if a lake is, for example, tens of kilometres long, mussels introduced at one end may not be able to interact with mussels introduced at the other end. The spatial scale at which mussels can benefit from each other could be difficult to determine (Stephens et al., 1999). This challenge could potentially be addressed with new technologies, such as environmental DNA sampling (Rees et al., 2014), yielding data on the spatial distribution of mussels in lakes (Youngbull and Devlin, 2018). Alternatively, historic invasion data could be used to fit an invasion model. This, however, would require a model for historic vector traffic. Constructing such a model may only be feasible if traffic pattern have not changed significantly in recent years.

Adding on to the required submodels discussed above, an improved invasion model could also take into account transport mortality of propagules or environmental similarities be-

tween donor and recipient regions. Zebra mussels are hypothesized to adjust to local habitat conditions over time (Elderkin and Klerks, 2005). Transport mortality could be modelled with a simple distance decay function (Seebens et al., 2013) or by introducing a maximal transport time (Mari et al., 2011). To account for habitat similarity, a general habitat suitability model would be needed (Seebens et al., 2013). Though a truly comprehensive invasion model would account for all the mechanisms discussed above (and more), incorporating any single mechanism could already improve risk assessment and management.

A second direction for future research is to model the spread of multiple invasive species all at once. Often, many invasive species can be transported by means of the same vector. Hence, a comprehensive traffic model extending the presented hybrid model could be used to model the spread of invasive species via road traffic on a general level, such as has been done for invasive species spreading via commercial navigation (Seebens et al., 2013). A general model could increase our understanding of the role of road traffic as a vector for invasive species.

6.2 Invasive species and disease management

6.2.1 The need for detailed management models

Due to the growing volume of traffic and trade all over the world, the task to manage invasive species and infectious diseases effectively will likely have increasing relevance in the foreseeable future. Consequently, there is a high need for new scientific insights about invasive species and infectious diseases, for models predicting their spread, and for tools facilitating management decisions with general and specific advice. The hybrid gravity and route choice model developed in this thesis can serve as the major component of a risk assessment tool. The management support tool presented in the fourth chapter enables managers to optimize control policies, to prepare for potential invasion and infection scenarios, and to identify

general guidelines for successful management. Hence, this thesis makes multiple significant contributions to effective invasive species and disease management.

Researchers have conducted several studies on optimal invasive species and disease management (Joshi et al., 2006; Potapov and Lewis, 2008; Blayneh et al., 2009; Finnoff et al., 2010; Carrasco et al., 2010; Epanchin-Niell and Wilen, 2012). Though efforts have been made to capture both the main mechanisms behind invasions or epidemics and the most important management options, many results from these studies are too general to be directly applied in practice. In the context of invasive species management, quite sophisticated models exist to predict the progression of invasions (Mari et al., 2011). However, when these models are paired with control models, the control actions are often modelled on a rather superficial level, considering management actions in broad categories, such as "early detection", "prevention of introductions", or "eradication".

Despite the undoubtedly significant insights gained with abstract models, management actions depend strongly on specific local conditions, constraints, and cost factors. For example, eradication may be feasible at one location but not at another. Should invasion prevention efforts then be higher at the former location? To what extent? Since managers have to answer these questions on a daily basis, models modelling management options and constraints on a greater level of detail would be needed.

6.2.2 Why are there so few detailed management models?

There are several potential reasons for the low number of detailed management models. First, there is the obviously great challenge to construct a spatially explicit model on a scale both broad and detailed enough to incorporate specific management options. Fitting such models requires, in particular, sufficiently large and detailed data sets. I can deem myself fortunate being given access to watercraft inspection data from the BC Invasive Mussel Defence Program. Without these data, I would not have been able to fit the models developed

in this thesis. Nonetheless, as noted above, detailed invasion models have already existed before this thesis.

In line with the challenge to gather data on the invasion process, the second obstacle for constructing realistic management models is that control options and constraints may not be easily known. In general, a lively dialog and data exchange between modellers and managers will be required to build realistic management models. This thesis benefited strongly from the collaboration with invasive species managers who contributed with their expertise and data. Such collaborations may not always be easy to establish.

Third, scientists may be averse to the difficulties involved with solving high-dimensional optimization problems, which might be NP-hard and thus practically unsolvable in some instances. The expected numbers of invasion events (and hence the invasion costs) are typically non-convex functions of decision variables and therefore difficult to minimize. Consequently, researchers may consider only few management options. There are studies attempting to solve difficult optimization problems with meta-heuristic approaches, such as neurodynamic programming (e.g. Potapov, 2008). Another alternative is to adjust management goals to make optimization problems tractable, as done in this thesis, where propagule transport is minimized instead of invasion risk. The results from simplified management scenarios could serve as useful initial guesses for problems with more complex objective functions.

A fourth factor potentially repelling scientists from considering detailed management models is that the effort required to build such models may be disproportionate to the expected scientific gain. Added realism may not lead to new, "interesting" results, and obtained results may largely coincide with common sense. Considering many scenario-specific details can also limit the general applicability of results, decreasing their relevance for a general audience. Detailed management optimization problems may in addition be unattractive from a methodological perspective. Realistic management problems are often not tractable with analytical techniques, such as Pontryagin's maximum principle (cf. Potapov et al., 2008; Blayneh et al., 2009), and numerical methods may be the only way to optimize strategies.

The arising numerical optimization problems, in turn, may be either already well studied in computing science (such as some of the linear integer problems considered in chapter 4) or so difficult that new solution algorithms or approximation methods are difficult to establish.

Despite these challenges and drawbacks, disease and invasive species managers need to make specific management decisions. They may, however, not have the training and capacities to build decision support tools themselves. Though solving high-detail management optimization problems may appear more like a (software) engineering task, a high expertise in both invasion biology/epidemiology and modelling are necessary. Hence, I believe that science can, and should, make a significant contribution to support policy makers. In this thesis, I have provided both corresponding theory and implementations.

6.2.3 Future directions

The management optimization tool I developed in this thesis targets only one stage of the invasion process: the introduction of propagules. As discussed earlier, a more comprehensive model for the invasion process would be desirable, and minimizing establishment of invasive species, and thus their impact, should be the final objective. This goal could be pursued if a more comprehensive model is available.

Though this thesis includes major advancements towards more detailed management models, the considered management model could be made more realistic with reasonable effort. This might be a worthwhile task for the future. However, a more sustainable solution would be to develop a decision support software that is easy to use for managers, flexibly adjustable to various invasion or epidemic scenarios, and allows managers without modelling expertise to add management options and constraints. In the best case, such a tool would integrate well with existing software used by managers, e.g. for geospatial analysis. The software package developed along with this thesis could serve as the basis for such a tool.

Spread prevention is only one among many ways to control epidemics and biological invasions. Another important part of successful invasive species and infectious disease man-

agement is rapid response. For example, the chances to eradicate an invasive species are much higher if the considered population is small and spatially contained (Pluess et al., 2012). As a consequence, early detection is an important component of successful invasive species and disease management. As for spread prevention, managers have to decide where and to what extend they spend resources on early detection. Developing spatially explicit quantitative management support tools for early detection will be an important task for future research.

6.3 Beyond infectious diseases and invasive species

The models and methods developed in this thesis were all developed with the motivation to facilitate invasive species and disease management. This applies also to the route computation algorithm presented in chapter 2 and to the confidence interval computation method presented in chapter 5. Nonetheless, these methods have applications exceeding the field of invasion and infectious disease modelling. Below I discuss some of these applications.

6.3.1 Traffic models

Traffic models have a broad range of applications including planning of road infra structure (Yang and Bell, 1998) and traffic control (Mahmassani, 2001). The hybrid gravity and route choice model developed in this thesis could also be applied to these problem sets. Though the hybrid model introduced in this thesis does not account for interactions between road travellers, e.g. when individuals avoid congested roads, such details could be included. Route candidates computed with the method developed in chapter 2 could also be applied in equilibrium-based traffic models (Sheffi, 1984) if it is infeasible to consider all roads in a road network. The methods developed in this thesis are particularly useful for large-scale traffic models, especially if analytical and likelihood based methods shall be used for traffic inference. These techniques may be better suited to gain a general understanding of traffic than alternative approaches such as individual based models.

6.3.2 Models for trade

Since gravity models were initially developed in the context of economics to model trade, the new method to fit these models introduced in this thesis may also benefit economic models. Though there may be more easily accessible data sources than road-side surveys to fit trade models, there could be instances where no trade data are readily available. For example, trade in developing countries may not be as well monitored as in industrialized countries. In instances in which data are sparse and large-scale models are needed, the presented hybrid approach could be a promising alternative to existing methods.

6.3.3 Statistical inference

The method to construct profile likelihood confidence intervals developed in chapter 5 has a vast range of potential applications, reaching as far as maximum likelihood model fitting techniques are used. The test results presented in chapter 5 suggest that the introduced algorithm proves particularly useful in situations in which earlier methods fail or are inefficient. Therefore, the new algorithm fills a methodological gap that may have hindered researchers from conducting a careful investigation of parameter uncertainty (Ren and Xia, 2019), and could thus lead to more reliable scientific insights.

6.3.4 Future directions

As tracking and routing applications on smart phones and driving assistance systems are increasingly used, enormous sets of traffic data are generated each day. Using only a small fraction of these data to inform traffic models opens new doors to model, predict, and understand traffic. Researchers have already presented several methods to incorporate these data to answer a variety of traffic related questions (Bierlaire et al., 2010; Tettamanti et al., 2012; Duan and Wei, 2014; Zimmermann et al., 2017). It would be a worthwhile task to develop methods to fit the introduced hybrid traffic model to such data.

In the last decade, we have witnessed a dramatic rise of machine learning techniques. Though machine learning techniques may not be suited to yield information on the mechanisms impacting traffic, they can lead to highly accurate predictions if enough training data are available (Humphries et al., 2018). A potential application of machine learning techniques in the context of the tool set developed in this thesis would be to estimate attractiveness and repulsiveness of traveller origins and destinations. This could lead to more accurate traffic estimates.

The introduced algorithm to construct profile likelihood confidence intervals builds on a number of subroutines and parameters that could potentially be adjusted to increase speed and robustness of the algorithm. Though this would certainly be worthwhile, I believe it is more important to make the algorithm accessible to end users of statistical software. Only if the algorithm is implemented in widely used programming languages, such as R or Python, the algorithm will fulfill its potential to benefit the scientific community. Hence, proper deployment of the algorithm may the most important next step.

6.4 Concluding remarks

In this thesis, I have developed a set of methods facilitating the control of biological invasions and epidemics. The presented methods are suited to assess the traffic of invasive species and disease vectors. The obtained traffic estimates can be used to gain a better scientific understanding of the spread of propagules and pathogens and to inform early detection and rapid response strategies against invasions and epidemics. Furthermore, the results can be used to optimize control measures seeking to prevent the spread of invasive species and diseases. I have introduced a method for this task in this thesis. The developed management support tool can account for local management constraints and provides specific management advice. I have applied the presented methods to the management of zebra and quagga mussels in British Columbia. There, the results are used by invasive species managers to inform

optimal operation of watercraft inspection stations. In conclusion, this thesis makes several important contributions to the control of epidemics and biological invasions.

In my thesis, I have applied, extended, and developed methods from a wide spectrum of scientific fields. I have used graph theoretical approaches to develop an efficient algorithm to identify locally optimal routes in route networks. Thereby, I have connected the research area of route planning with the field of traffic modelling. I have used the resulting paths as input for a hybrid route choice and gravity model, the latter of which originated from economics. I have developed and applied statistical tools to fit the hybrid model to data and to validate the model. Furthermore, I have extended methods from the area of numerical optimization to construct confidence intervals for the hybrid model. Lastly, I have used the estimates from the hybrid model to optimize the management of invasive species and epidemics. Thereby, I applied methods from the field of discrete optimization.

Considering the range of the developed and applied methods, this thesis is a striking example for the usefulness and necessity of interdisciplinary approaches. Since the developed methods can furthermore be applied in various contexts, this thesis can have an impact reaching far beyond the management of epidemics and invasions.

Bibliography

Abraham, I., Delling, D., Goldberg, A. V., and Werneck, R. F. (2013). Alternative routes in road networks. *Journal of Experimental Algorithmics*, 18:1.3:1–17.

Abraham, I., Fiat, A., Goldberg, A. V., and Werneck, R. F. (2010). Highway dimension, shortest paths, and provably efficient algorithms. In *Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pages 782–793, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.

Ackerman, J. D., Sim, B., Nichols, S. J., and Claudi, R. (1994). A review of the early life history of zebra mussels (*Dreissena polymorpha*): comparisons with marine bivalves. *Canadian Journal of Zoology*, 72(7):1169–1179.

Ageev, A. and Sviridenko, M. (2004). Pipage rounding: a new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*, 8(3):307–328.

Ageev, A. A. and Sviridenko, M. I. (1999). Approximation algorithms for maximum coverage and max cut with given sizes of parts. In Cornuéjols, G., Burkard, R. E., and Woeginger, G. J., editors, *Integer Programming and Combinatorial Optimization*, volume 1610, pages 17–30. Springer Berlin Heidelberg, Berlin, Heidelberg.

Aho, K., Derryberry, D., and Peterson, T. (2014). Model selection for ecologists: the worldviews of AIC and BIC. *Ecology*, 95(3):631–636.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723.

Akrami, Y., Scott, P., Edsjö, J., Conrad, J., and Bergström, L. (2010). A profile likelihood analysis of the constrained MSSM with genetic algorithms. *Journal of High Energy Physics*, 2010(4):57.

Albers, H. J., Fischer, C., and Sanchirico, J. N. (2010). Invasive species management in a spatially heterogeneous world: Effects of uniform policies. *Resource and Energy Economics*, 32(4):483–499.

Alberta Environment and Parks Fish and Wildlife Policy (2015). Alberta Aquatic Invasive Species Program 2015 annual report. Technical report, Edmonton, AB.

Alivand, M., Hochmair, H., and Srinivasan, S. (2015). Analyzing how travelers choose scenic routes using route choice models. *Computers, Environment and Urban Systems*, 50:41–52.

Anderson, D. R., Burnham, K. P., and Thompson, W. L. (2000). Null hypothesis testing: problems, prevalence, and an alternative. *The Journal of Wildlife Management*, 64(4):912.

Anderson, J. E. (2011). The gravity model. *Annual Review of Economics*, 3(1):133–160.

Azevedo, J., Santos Costa, M. E. O., Silvestre Madeira, J. J. E., and Vieira Martins, E. Q. (1993). An algorithm for the ranking of shortest paths. *European Journal of Operational Research*, 69(1):97–106.

Barrios, J., Verstraeten, W., Maes, P., Aerts, J.-M., Farifteh, J., and Coppin, P. (2012). Using the gravity model to estimate the spatial spread of vector-borne diseases. *International Journal of Environmental Research and Public Health*, 9(12):4346–4364.

Bast, H., Delling, D., Goldberg, A., Müller-Hannemann, M., Pajor, T., Sanders, P., Wagner, D., and Werneck, R. F. (2016). Route planning in transportation networks. In Kliemann,

L. and Sanders, P., editors, *Algorithm Engineering*, volume 9220, pages 19–80. Springer International Publishing, Cham.

BC Ministry of Environment and Climate Change Strategy (2019). Invasive Mussel Defence Program launches new season. Retrieved from https://news.gov.bc.ca/releases/2019ENV0023-001099.

Bellard, C., Cassey, P., and Blackburn, T. M. (2016). Alien species as a driver of recent extinctions. *Biology Letters*, 12(2):20150623.

Ben-Akiva, M., Bergman, M., Daly, A. J., and Ramaswamy, R. (1984). Modeling inter-urban route choice behaviour. In Volmuller, J. and Hamerslag, R., editors, *Proceedings of the 9th international symposium on transportation and traffic theory*, pages 299–330. VNU Press Utrecht.

Besag, J. (1975). Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 24(3):179–195.

Bierlaire, M., Chen, J., and Newman, J. (2010). Modeling Route Choice Behavior From Smartphone GPS data. Technical Report TRANSP-OR 101016, Transport and Mobility Laboratory, Ecole Polytechnique Fédérale de Lausanne.

Blackwood, J., Hastings, A., and Costello, C. (2010). Cost-effective management of invasive species using linear-quadratic control. *Ecological Economics*, 69(3):519–527.

Blayneh, K., Cao, Y., and Kwon, H.-D. (2009). Optimal control of vector-borne diseases: Treatment and prevention. *Discrete and Continuous Dynamical Systems - Series B*, 11(3):587–611.

Bossenbroek, J. M., Johnson, L. E., Peters, B., and Lodge, D. M. (2007). Forecasting the expansion of zebra mussels in the United States. *Conservation Biology*, 21(3):800–810.

Bossenbroek, J. M., Kraft, C. E., and Nekola, J. C. (2001). Prediction of long-distance dispersal using gravity models: zebra mussel invasion of inland lakes. *Ecological Applications*, 11(6):1778–1788.

Bovy, P. H. L. (2009). On modelling route choice sets in transportation networks: a synthesis. *Transport Reviews*, 29(1):43–68.

Briski, E., Allinger, L. E., Balcer, M., Cangelosi, A., Fanberg, L., Markee, T. P., Mays, N., Polkinghorne, C. N., Prihoda, K. R., Reavie, E. D., Regan, D. H., Reid, D. M., Saillard, H. J., Schwerdt, T., Schaefer, H., TenEyck, M., Wiley, C. J., and Bailey, S. A. (2013). Multidimensional approach to invasive species prevention. *Environmental Science & Technology*, 47(3):1216–1221.

Brodtkorb, P. A. and D'Errico, J. (2019). numdifftools 0.9.39. Retrieved from https://github.com/pbrod/numdifftools.

Buckland, S. T. (1984). Monte Carlo confidence intervals. *Biometrics*, 40(3):811.

Burger, M., van Oort, F., and Linders, G.-J. (2009). On the specification of the gravity model of trade: zeros, excess zeros and zero-inflated estimation. *Spatial Economic Analysis*, 4(2):167–190.

Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208.

Carlton, J. T. and Ruiz, G. M. (2005). Vector science and integrated vector management in bioinvasion ecology: conceptual frameworks. In Mooney, H. A., Mack, R. N., McNeely, J. A., Neville, L. E., Schei, P. J., and Waage, J. K., editors, *Invasive alien species: a new synthesis.*, volume 63, page 36. Island Press, Washington, DC.

Carrasco, L., Mumford, J., MacLeod, A., Knight, J., and Baker, R. (2010). Comprehensive bioeconomic modelling of multiple harmful non-indigenous species. *Ecological Economics*, 69(6):1303–1312.

Cascetta, E., Nuzzolo, A., Russo, F., and Vitetta, A. (1996). A modified logit route choice model overcoming path overlapping problems. Specification and some calibration results for interurban networks. In Lesort, J.-B., editor, *Transportation and Traffic Theory. Proceedings of the 13th International Symposium on Transportation and Traffic Theory*, pages 697–711, Lyon, France.

Casella, G. and Berger, R. L. (2002). *Statistical inference*. Thomson Learning, Pacific Grove, CA, 2nd edition.

Chakraborti, R. K., Madon, S., Kaur, J., and Gabel, D. (2013). General overview of zebra and quagga mussels: what we do and do not know. In Nalepa, T. F. and Schloesser, D. W., editors, *Quagga and Zebra Mussels: Biology, Impacts, and Control*. CRC Press, 2nd edition.

Chew, M. and Carroll, S. P. (2011). The invasive ideology: biologists and conservationists are too eager to demonize non-native species. Retrieved from https://www.the-scientist.com/news-opinion/opinion-the-invasive-ideology-41967.

Chivers, C. and Leung, B. (2012). Predicting invasions: alternative models of human-mediated dispersal and interactions between dispersal network structure and Allee effects. *Journal of Applied Ecology*, 49(5):1113–1123.

Colautti, R. I. and MacIsaac, H. J. (2004). A neutral terminology to define 'invasive' species: Defining invasive species. *Diversity and Distributions*, 10(2):135–141.

Conforti, M., Cornuejols, G., and Zambelli, G. (2014). *Integer programming*, volume 271 of *Graduate Texts in Mathematics*. Springer International Publishing, Cham.

Conn, A. R., Gould, N. I. M., and Toint, P. L. (2000). *Trust-region methods*. MPS-SIAM series on optimization. Society for Industrial and Applied Mathematics, Philadelphia, PA.

Connelly, N. A., O'Neill, C. R., Knuth, B. A., and Brown, T. L. (2007). Economic impacts of zebra mussels on drinking water treatment and electric power generation facilities. *Environmental Management*, 40(1):105–112.

Cook, R. D. and Weisberg, S. (1990). Confidence curves in nonlinear regression. *Journal of the American Statistical Association*, 85(410):544–551.

Cox, D. R. and Snell, E. J. (1989). *Analysis of binary data*. Number 32 in Monographs on Statistics and Applied Probability. Routledge, Boca Raton, FL, 2nd edition.

Daehler, C. C. (2001). Two ways to be an invader, but one is more suitable for ecology. *Bulletin of the Ecological Society of America*, 82(1):101–102.

Dantzig, G. B. (1998). *Linear programming and extensions*. Princeton landmarks in mathematics and physics. Princeton Univ. Press, Princeton, NJ, 11. printing, 1. paperback printing edition. OCLC: 245738716.

Davis, M. A., Chew, M. K., Hobbs, R. J., Lugo, A. E., Ewel, J. J., Vermeij, G. J., Brown, J. H., Rosenzweig, M. L., Gardener, M. R., Carroll, S. P., Thompson, K., Pickett, S. T. A., Stromberg, J. C., Tredici, P. D., Suding, K. N., Ehrenfeld, J. G., Philip Grime, J., Mascaro, J., and Briggs, J. C. (2011). Don't judge species on their origins. *Nature*, 474(7350):153–154.

Davis, M. A., Thompson, K., and Grime, J. P. (2001). Charles S. Elton and the dissociation of invasion ecology from the rest of ecology. *Diversity and Distributions*, 7(1-2):97–102.

De La Barra, T., Perez, B., and Anez, J. (1993). Multi-dimensional path search and assignment. *Transportation planning methods*, (366):307–320.

Delling, D., Goldberg, A. V., Pajor, T., and Werneck, R. F. (2015). Customizable route planning in road networks. *Transportation Science*, 51(2):566–591.

Di, X. and Liu, H. X. (2016). Boundedly rational route choice behavior: A review of models and methodologies. *Transportation Research Part B: Methodological*, 85:142–179.

DiCiccio, T. J. and Tibshirani, R. (1991). On the implementation of profile likelihood methods. Technical report, University of Toronto, Department of Statistics.

Didham, R., Tylianakis, J., Gemmell, N., Rand, T., and Ewers, R. (2007). Interactive effects of habitat modification and species invasion on native species decline. *Trends in Ecology & Evolution*, 22(9):489–496.

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

Doniec, A., Mandiau, R., Piechowiak, S., and Espié, S. (2008). A behavioral multi-agent model for road traffic simulation. *Engineering Applications of Artificial Intelligence*, 21(8):1443–1454.

Drake, D. A. R. and Mandrak, N. E. (2010). Least-cost transportation networks predict spatial interaction of invasion vectors. *Ecological Applications*, 20(8):2286–2299.

Drake, D. A. R. and Mandrak, N. E. (2014). Bycatch, bait, anglers, and roads: quantifying vector activity and propagule introduction risk across lake ecosystems. *Ecological Applications*, 24(4):877–894.

Duan, Z. and Wei, Y. (2014). Revealing taxi driver route choice characteristics based on GPS data. In *CICTP 2014*, pages 565–573, Changsha, China. American Society of Civil Engineers.

Efron, B. (1981). Nonparametric standard errors and confidence intervals. *Canadian Journal of Statistics*, 9(2):139–158.

Elderkin, C. L. and Klerks, P. L. (2005). Variation in thermal tolerance among three Mississippi River populations of the zebra mussel, *Dreissena polymorpha*. *Journal of Shellfish Research*, 24(1):221–226.

Epanchin-Niell, R. S. and Wilen, J. E. (2012). Optimal spatial control of biological invasions. *Journal of Environmental Economics and Management*, 63(2):260–270.

Eubank, R. L. and Webster, J. T. (1985). The singular-value decomposition as a tool for solving estimability problems. *The American Statistician*, 39(1):64.

Famoye, F. (1998). Bootstrap based tests for generalized negative binomial distribution. *Computing*, 61(4):359–369.

Feige, U. (1998). A threshold of ln n for approximating set cover. *Journal of the ACM*, 45(4):634–652.

Ferrari, M. J., Bjørnstad, O. N., Partain, J. L., and Antonovics, J. (2006). A gravity model for the spread of a pollinator-borne plant pathogen. *The American Naturalist*, 168(3):294–303.

Finnoff, D., Potapov, A., and Lewis, M. A. (2010). Control and the management of a spreading invader. *Resource and Energy Economics*, 32(4):534–550.

Flowerdew, R. and Aitkin, M. (1982). A method of fitting the gravity model based on the poisson distribution. *Journal of Regional Science*, 22(2):191–202.

Friedrich, H., Tavasszy, L., and Davydenko, I. (2014). Distribution structures. In Tavasszy, L. and de Jong, G., editors, *Modelling Freight Transport*, pages 65–87. Elsevier, Amsterdam, 1st edition.

Gardner, W., Mulvey, E. P., and Shaw, E. C. (1995). Regression analyses of counts and rates: Poisson, overdispersed Poisson, and negative binomial models. *Psychological Bulletin*, 118(3):392–404.

Ghosh, J. and Samanta, T. (2001). Model selection – an overview. *Current Science*, 80(9):1135.

Gimenez, O., Choquet, R., Lamor, L., Scofield, P., Fletcher, D., Lebreton, J.-D., and Pradel, R. (2005). Efficient profile-likelihood confidence intervals for capture-recapture models. *Journal of Agricultural, Biological, and Environmental Statistics*, 10(2):184–196.

Goldberg, A. V., Kaplan, H., and Werneck, R. F. (2006). Reach for A*: efficient point-to-point shortest path algorithms. In Raman, R. and Stallmann, M. F., editors, *2006 Proceedings of the Eighth Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 129–143. Society for Industrial and Applied Mathematics, Philadelphia, PA.

Golub, G. and Kahan, W. (1965). Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics Series B Numerical Analysis*, 2(2):205–224.

Griewank, A. (1989). On automatic differentiation. *Mathematical Programming: recent developments and applications*, 6(6):83–107.

Haefner, J. W. (2005). *Modeling biological systems: principles and applications*. Springer, New York, 2nd edition.

Hastings, A., Hall, R. J., and Taylor, C. M. (2006). A simple approach to optimal control of invasive species. *Theoretical Population Biology*, 70(4):431–435.

Hellmann, J. J., Byers, J. E., Bierwagen, B. G., and Dukes, J. S. (2008). Five potential consequences of climate change for invasive species. *Conservation Biology*, 22(3):534–543.

Hirzel, A. H. and Le Lay, G. (2008). Habitat suitability modelling and niche theory. *Journal of Applied Ecology*, 45(5):1372–1381.

Hobbs, R. J. and Huenneke, L. F. (1992). Disturbance, diversity, and invasion: implications for conservation. *Conservation Biology*, 6(3):324–337.

Hulme, P. E. (2009). Trade, transport and trouble: managing invasive species pathways in an era of globalization. *Journal of Applied Ecology*, 46(1):10–18.

Humphries, G., Magness, D. R., and Huettmann, F., editors (2018). *Machine learning for ecology and sustainable natural resource management*. Springer International Publishing, Cham.

Hyytiäinen, K., Lehtiniemi, M., Niemi, J. K., and Tikka, K. (2013). An optimization framework for addressing aquatic invasive species. *Ecological Economics*, 91:69–79.

Inter-Ministry Invasive Species Working Group (2015). Zebra and quagga mussel early detection and rapid response plan. Technical report, Victoria, BC.

Jacquez, J. A. and Greif, P. (1985). Numerical parameter identifiability and estimability: Integrating identifiability, estimability, and optimal sampling design. *Mathematical Biosciences*, 77(1-2):201–227.

Johnson, L. E., Ricciardi, A., and Carlton, J. T. (2001). Overland dispersal of aquatic invasive species: a risk assessment of transient recreational boating. *Ecological Applications*, 11(6):1789–1799.

Johnson, R., Crafton, R. E., and Upton, H. F. (2017). Invasive species: major laws and the role of selected federal agencies. Technical report, Congressional Research Service, Washington, DC.

Jones, E., Oliphant, T., and Peterson, P. (2001). SciPy: open source scientific tools for Python. Retrieved from https://scipy.org/.

Jones, H. P., Holmes, N. D., Butchart, S. H. M., Tershy, B. R., Kappes, P. J., Corkery, I., Aguirre-Muñoz, A., Armstrong, D. P., Bonnaud, E., Burbidge, A. A., Campbell, K., Courchamp, F., Cowan, P. E., Cuthbert, R. J., Ebbert, S., Genovesi, P., Howald, G. R., Keitt, B. S., Kress, S. W., Miskelly, C. M., Oppel, S., Poncet, S., Rauzon, M. J., Rocamora,

G., Russell, J. C., Samaniego-Herrera, A., Seddon, P. J., Spatz, D. R., Towns, D. R., and Croll, D. A. (2016). Invasive mammal eradication on islands results in substantial conservation gains. *Proceedings of the National Academy of Sciences*, 113(15):4033–4038.

Jones, L. A. and Ricciardi, A. (2005). Influence of physicochemical factors on the distribution and biomass of invasive mussels (*Dreissena polymorpha* and *Dreissena bugensis*) in the St. Lawrence River. *Canadian Journal of Fisheries and Aquatic Sciences*, 62(9):1953–1962.

Joshi, H. R., Lenhart, S., Li, M. Y., and Wang, L. (2006). Optimal control methods applied to disease models. In Gumel, A. B., Castillo-Chavez, C., Mickens, R. E., and Clemence, D. P., editors, *Mathematical studies on human disease dynamics: emerging paradigms and challenges*, volume 410 of *Contemporary Mathematics*, pages 187–208, Snowbird, UT. American Mathematical Society.

Karatayev, A. Y., Burlakova, L. E., Mastitsky, S. E., and Padilla, D. K. (2015a). Predicting the spread of aquatic invaders: insight from 200 years of invasion by zebra mussels. *Ecological Applications*, 25(2):430–440.

Karatayev, A. Y., Burlakova, L. E., and Padilla, D. K. (2013). General overview of zebra and quagga mussels: what we do and do not know. In Nalepa, T. F. and Schloesser, D. W., editors, *Quagga and Zebra Mussels: Biology, Impacts, and Control*. CRC Press, 2nd edition.

Karatayev, A. Y., Burlakova, L. E., and Padilla, D. K. (2015b). Zebra versus quagga mussels: a review of their spread, population dynamics, and ecosystem impacts. *Hydrobiologia*, 746(1):97–112.

Karesh, W. B., Cook, R. A., Bennett, E. L., and Newcomb, J. (2005). Wildlife trade and global disease emergence. *Emerging Infectious Diseases*, 11(7):1000–1002.

Keane, R. (2002). Exotic plant invasions and the enemy release hypothesis. *Trends in Ecology & Evolution*, 17(4):164–170.

Khuller, S., Moss, A., and Naor, J. S. (1999). The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45.

Kıbış, E. Y. and Büyüktahtakın, İ. E. (2017). Optimizing invasive species management: a mixed-integer linear programming approach. *European Journal of Operational Research*, 259(1):308–321.

Kimball, A. M. (2006). *Risky trade: infectious disease in the era of global trade*. Ashgate, Aldershot, Hants, England ; Burlington, VT. OCLC: ocm62714457.

Knoll, L. B., Sarnelle, O., Hamilton, S. K., Kissman, C. E., Wilson, A. E., Rose, J. B., and Morgan, M. R. (2008). Invasive zebra mussels (*Dreissena polymorpha*) increase cyanobacterial toxin concentrations in low-nutrient lakes. *Canadian Journal of Fisheries and Aquatic Sciences*, 65(3):448–455.

Kobitzsch, M. (2013). An alternative approach to alternative routes: HiDAR. In Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F., Mitchell, J. C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M. Y., Weikum, G., Bodlaender, H. L., and Italiano, G. F., editors, *Algorithms – ESA 2013*, volume 8125, pages 613–624. Springer Berlin Heidelberg, Berlin, Heidelberg.

Koch, F. H., Yemshanov, D., Magarey, R. D., and Smith, W. D. (2012). Dispersal of invasive forest insects via recreational firewood: a quantitative analysis. *Journal of Economic Entomology*, 105(2):438–450.

Kolar, C. S. (2002). Ecological predictions and risk assessment for alien fishes in North America. *Science*, 298(5596):1233–1236.

Kot, M., Lewis, M. A., and van den Driessche, P. (1996). Dispersal data and the spread of invading organisms. *Ecology*, 77(7):2027–2042.

Kraft, D. (1988). A software package for sequential quadratic programming. Technical Report DFVLR-FB 88-28, DLR German Aerospace Center – Institute for Flight Mechanics, Köln, Germany.

Lai, K.-L., Crassidis, J., Cheng, Y., and Kim, J. (2005). New complex-step derivative approximations with application to second-order kalman filtering. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, California. American Institute of Aeronautics and Astronautics.

Lalee, M., Nocedal, J., and Plantenga, T. (1998). On the implementation of an algorithm for large-scale equality constrained optimization. *SIAM Journal on Optimization*, 8(3):682–706.

Lee, A. (2010). Circular data. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):477–486.

Lele, S. R., Nadeem, K., and Schmuland, B. (2010). Estimability and likelihood inference for generalized linear mixed models using data cloning. *Journal of the American Statistical Association*, 105(492):1617–1625.

Leung, B., Drake, J. M., and Lodge, D. M. (2004). Predicting invasions: Propagule pressure and the gravity of Allee effects. *Ecology*, 85(6):1651–1660.

Leung, B., Lodge, D. M., Finnoff, D., Shogren, J. F., Lewis, M. A., and Lamberti, G. (2002). An ounce of prevention or a pound of cure: bioeconomic risk analysis of invasive species. *Proceedings of the Royal Society B: Biological Sciences*, 269(1508):2407–2413.

Leung, B. and Mandrak, N. E. (2007). The risk of establishment of aquatic invasive species: joining invasibility and propagule pressure. *Proceedings of the Royal Society B: Biological Sciences*, 274(1625):2603–2609.

Levine, J. M. and D'Antonio, C. M. (1999). Elton revisited: a review of evidence linking diversity and invasibility. *Oikos*, 87(1):15–26.

Lewis, M., Petrovskii, S. V., and Potts, J. R. (2016). *The mathematics behind biological invasions*. Number 44 in Interdisciplinary applied mathematics. Springer, Cham. OCLC: 957633921.

Li, X., Tian, H., Lai, D., and Zhang, Z. (2011). Validation of the gravity model in predicting the global spread of influenza. *International Journal of Environmental Research and Public Health*, 8(8):3134–3143.

Lindsay, B. G. (1988). Composite likelihood methods. In Prabhu, N. U., editor, *Statistical Inference from Stochastic Processes*, volume 80 of *Contemporary Mathematics*, pages 221–239. American Mathematical Society, Providence, RI.

Lockwood, J. L. (1999). Using taxonomy to predict success among introduced avifauna: relative importance of transport and establishment. *Conservation Biology*, 13(3):560–567.

Lockwood, J. L., Hoopes, M. F., and Marchetti, M. P. (2013). *Invasion ecology*. Wiley-Blackwell, Chichester, West Sussex, UK, 2nd edition.

Lodge, D. M., Williams, S., MacIsaac, H. J., Hayes, K. R., Leung, B., Reichard, S., Mack, R. N., Moyle, P. B., Smith, M., Andow, D. A., Carlton, J. T., and McMichael, A. (2006). Biological invasions: recommendations for US policy and management. *Ecological Applications*, 16(6):2035–2054.

Lund, K., Cattoor, K. B., Fieldseth, E., Sweet, J., and McCartney, M. A. (2018). Zebra mussel (*Dreissena polymorpha*) eradication efforts in Christmas Lake, Minnesota. *Lake and Reservoir Management*, 34(1):7–20.

Luxen, D. and Schieferdecker, D. (2015). Candidate sets for alternative routes in road networks. *Journal of Experimental Algorithmics*, 19:1.1–1.28.

Mack, R. N. (2003). Plant naturalizations and invasions in the eastern United States: 1634-1860. *Annals of the Missouri Botanical Garden*, 90(1):77.

Mackie, G. (1991). Biology of the exotic zebra mussel, *Dreissena polymorpha*, in relation to native bivalves and its potential impact in Lake St. Clair. *Hydrobiologia*, 219(1):251–268.

Mahmassani, H. S. (2001). Dynamic network traffic assignment and simulation methodology for advanced system management applications. *Networks and Spatial Economics*, 1(3/4):267–292.

Mangin, S. (2011). The 100th Meridian Initiative: a strategic approach to prevent the westward spread of zebra mussels and other aquatic nuisance species. Technical Report 152, U.S. Fish and Wildlife Service, Arlington, VA.

Marbuah, G., Gren, I.-M., and McKie, B. (2014). Economics of harmful invasive species: a review. *Diversity*, 6(3):500–523.

Mari, L., Bertuzzo, E., Casagrandi, R., Gatto, M., Levin, S. A., Rodriguez-Iturbe, I., and Rinaldo, A. (2011). Hydrologic controls and anthropogenic drivers of the zebra mussel invasion of the Mississippi-Missouri river system. *Water Resources Research*, 47(3):1–16.

McDermott, S. M., Finnoff, D. C., and Shogren, J. F. (2013). The welfare impacts of an invasive species: Endogenous vs. exogenous price models. *Ecological Economics*, 85:43–49.

Mills, E. L., Rosenberg, G., Spidle, A. P., Ludyanskiy, M., Pligin, Y., and May, B. (1996). A review of the biology and ecology of the quagga mussel (*Dreissena bugensis*), a second species of freshwater dreissenid introduced to North America. *American Zoologist*, 36(3):271–286.

Moerbeek, M., Piersma, A. H., and Slob, W. (2004). A comparison of three methods for calculating confidence intervals for the benchmark dose. *Risk Analysis*, 24(1):31–40.

Muirhead, J. R., Lewis, M. A., and MacIsaac, H. J. (2011). Prediction and error in multi-stage models for spread of aquatic non-indigenous species: Prediction and error in multi-stage models. *Diversity and Distributions*, 17(2):323–337.

Muirhead, J. R. and MacIsaac, H. J. (2011). Evaluation of stochastic gravity model selection for use in estimating non-indigenous species dispersal and establishment. *Biological Invasions*, 13(11):2445–2458.

Neale, M. C. and Miller, M. B. (1997). The use of likelihood-based confidence intervals in genetic models. *Behavior Genetics*, 27(2):113–120.

Nocedal, J. and Wright, S. J. (2006). *Numerical optimization*. Springer series in operations research. Springer, New York, 2nd edition.

Padilla, D. K., Chotkowski, M. A., and Buchan, L. A. J. (1996). Predicting the spread of zebra mussels (*Dreissena polymorpha*) to inland waters using boater movement patterns. *Global Ecology and Biogeography Letters*, 5(6):353.

Pejchar, L. and Mooney, H. A. (2009). Invasive species, ecosystem services and human well-being. *Trends in Ecology & Evolution*, 24(9):497–504.

Penrose, R. (1955). A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(3):406–413.

Pfeiffer, J. M. and Voeks, R. A. (2008). Biological invasions and biocultural diversity: linking ecological and cultural systems. *Environmental Conservation*, 35(04):281.

Pick, F. R. (2016). Blooming algae: a Canadian perspective on the rise of toxic cyanobacteria. *Canadian Journal of Fisheries and Aquatic Sciences*, 73(7):1149–1158.

Pimentel, D., Lach, L., Zuniga, R., and Morrison, D. (2000). Environmental and Economic Costs of Nonindigenous Species in the United States. *BioScience*, 50(1):53.

Pimentel, D., Zuniga, R., and Morrison, D. (2005). Update on the environmental and economic costs associated with alien-invasive species in the United States. *Ecological Economics*, 52(3):273–288.

Pluess, T., Cannon, R., Jarošík, V., Pergl, J., Pyšek, P., and Bacher, S. (2012). When are eradication campaigns successful? A test of common assumptions. *Biological Invasions*, 14(7):1365–1378.

Ponciano, J. M., Taper, M. L., Dennis, B., and Lele, S. R. (2009). Hierarchical models in ecology: confidence intervals, hypothesis testing, and model selection using data cloning. *Ecology*, 90(2):356–362.

Potapov, A. (2008). Stochastic model of lake system invasion and its optimal control: neuro-dynamic programming as a solution method. *Natural Resource Modeling*, 22(2):257–288.

Potapov, A., Muirhead, J., Yan, N., Lele, S., and Lewis, M. (2011). Models of lake invasibility by Bythotrephes longimanus, a non-indigenous zooplankton. *Biological Invasions*, 13(11):2459–2476.

Potapov, A., Muirhead, J. R., Lele, S. R., and Lewis, M. A. (2010). Stochastic gravity models for modeling lake invasions. *Ecological Modelling*, 222(4):964–972.

Potapov, A. B. and Lewis, M. A. (2008). Allee effect and control of lake system invasion. *Bulletin of Mathematical Biology*, 70(5):1371–1397.

Potapov, A. B., Lewis, M. A., and Finnoff, D. C. (2008). Optimal control of biological invasions in lake networks. *Natural Resource Modeling*, 20(3):351–379.

Prasad, A. M., Iverson, L. R., Peters, M. P., Bossenbroek, J. M., Matthews, S. N., Davis Sydnor, T., and Schwartz, M. W. (2010). Modeling the invasive emerald ash borer risk of spread using a spatially explicit cellular model. *Landscape Ecology*, 25(3):353–369.

Prato, C. G. (2009). Route choice modeling: past, present and future research directions. *Journal of Choice Modelling*, 2(1):65–100.

Prato, C. G. and Bekhor, S. (2006). Applying branch-and-bound technique to route choice set generation. *Transportation Research Record: Journal of the Transportation Research Board*, 1985(1):19–28.

Ramcharan, C. W., Padilla, D. K., and Dodson, S. I. (1992). Models to predict potential occurrence and density of the zebra mussel, *Dreissena polymorpha*. *Canadian Journal of Fisheries and Aquatic Sciences*, 49(12):2611–2620.

Raney, B., Cetin, N., Völlmy, A., Vrtic, M., Axhausen, K., and Nagel, K. (2003). An agent-based microsimulation model of swiss travel: first results. *Networks and Spatial Economics*, 3(1):23–41.

Rao, C. R. (1967). Calculus of generalized inverse of matrices. Part 1: general theory. *Sankhya Ser. A*, 29:317–342.

Raue, A., Kreutz, C., Maiwald, T., Bachmann, J., Schilling, M., Klingmüller, U., and Timmer, J. (2009). Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. *Bioinformatics*, 25(15):1923–1929.

Rees, H. C., Maddison, B. C., Middleditch, D. J., Patmore, J. R., and Gough, K. C. (2014). Review: the detection of aquatic animal species using environmental DNA - a review of eDNA as a survey tool in ecology. *Journal of Applied Ecology*, 51(5):1450–1459.

Ren, X. and Xia, J. (2019). An algorithm for computing profile likelihood based pointwise confidence intervals for nonlinear dose-response models. *PLOS ONE*, 14(1):e0210953.

Richardson, D. M., Pysek, P., Rejmanek, M., Barbour, M. G., Panetta, F. D., and West, C. J. (2000). Naturalization and invasion of alien plants: concepts and definitions. *Diversity and Distributions*, 6(2):93–107.

Rosaen, A. L., Grover, E. A., and Spencer, C. W. (2012). The costs of aquatic invasive species to Great Lakes states. Technical report, Anderson Economical Group, East Lansing, MI.

Rothlisberger, J. D. and Lodge, D. M. (2011). Limitations of gravity models in predicting the spread of Eurasian watermilfoil: assessment of gravity models. *Conservation Biology*, 25(1):64–72.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.

Seebens, H., Gastner, M. T., and Blasius, B. (2013). The risk of marine bioinvasion caused by global shipping. *Ecology Letters*, 16(6):782–790.

Sheffi, Y. (1984). *Urban transportation networks: equilibrium analysis with mathematical programming methods*. Prentice-Hall, Englewood Cliffs, NJ.

Shine, C., Kettunen, M., Genovesi, P., Essl, F., Gollasch, S., Rabitsch, W., Scalera, R., Starfinger, U., and ten Brink, P. (2010). Assessment to support continued development of the EU Strategy to combat invasive alien species. Final Report for the European Commission, Institute for European Environmental Policy (IEEP), Brussels, Belgium.

Siderelis, C. and Moore, R. L. (1998). Recreation demand and the influence of site preference variables. *Journal of Leisure Research*, 30(3):301–318.

Simberloff, D. (2003). Confronting introduced species: a form of xenophobia? *Biological Invasions*, 5(3):179–192.

Simberloff, D. and Von Holle, B. (1999). Positive interactions of nonindigenous species: invasional meltdown? *Biological invasions*, 1(1):21–32.

Simon, H. A. (1957). *Models of man; social and rational*. Models of man; social and rational. Wiley, Oxford, England.

Simpson, A., Jarnevich, C., Madsen, J., Westbrooks, R., Fournier, C., Mehrhoff, L., Browne, M., Graham, J., and Sellers, E. (2009). Invasive species information networks: collabora-

tion at multiple scales for prevention, early detection, and rapid response to invasive alien species. *Biodiversity*, 10(2-3):5–13.

Stephens, P. A., Sutherland, W. J., and Freckleton, R. P. (1999). What is the Allee effect? *Oikos*, 87(1):185.

Stijns, J.-P. (2003). An empirical test of the Dutch disease hypothesis using a gravity model of trade. *SSRN Electronic Journal*.

Stoeckel, J. A., Padilla, D. K., Schneider, D. W., and Rehmann, C. R. (2004). Laboratory culture of *Dreissena polymorpha* larvae: spawning success, adult fecundity, and larval mortality patterns. *Canadian Journal of Zoology*, 82(9):1436–1443.

Storn, R. and Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359.

Strayer, D. L. (2009). Twenty years of zebra mussels: lessons from the mollusk that made headlines. *Frontiers in Ecology and the Environment*, 7(3):135–141.

Stryhn, H. and Christensen, J. (2003). Confidence intervals by the profile likelihood method, with applications in veterinary epidemiology. Vina del Mar, Chile.

Surkov, I. V., Oude Lansink, A. G. J. M., van Kooten, O., and van der Werf, W. (2008). A model of optimal import phytosanitary inspection under capacity constraint. *Agricultural Economics*, 38(3):363–373.

Tettamanti, T., Demeter, H., and Varga, I. (2012). Route choice estimation based on cellular signaling data. *Acta Polytechnica Hungarica*, 9(4):207–220.

Ton, D., Duives, D., Cats, O., and Hoogendoorn, S. (2018). Evaluating a data-driven approach for choice set identification using GPS bicycle route choice data from Amsterdam. *Travel Behaviour and Society*, 13:105–117.

Trombulak, S. C. and Frissell, C. A. (2000). Review of ecological effects of roads on terrestrial and aquatic communities. *Conservation Biology*, 14(1):18–30.

Trullols, O., Fiore, M., Casetti, C., Chiasserini, C., and Barcelo Ordinas, J. (2010). Planning roadside infrastructure for information dissemination in intelligent transportation systems. *Computer Communications*, 33(4):432–442.

Tuite, A. R., Tien, J., Eisenberg, M., Earn, D. J., Ma, J., and Fisman, D. N. (2011). Cholera epidemic in Haiti, 2010: using a transmission model to explain spatial spread of disease and identify optimal control interventions. *Annals of Internal Medicine*, 154(9):593.

Turbelin, A. J., Malamud, B. D., and Francis, R. A. (2017). Mapping the global state of invasive alien species: patterns of invasion and policy responses: Mapping the global state of invasive alien species. *Global Ecology and Biogeography*, 26(1):78–92.

USGS (2019). NAS - nonindigenous aquatic species. Retrieved from https://nas.er.usgs.gov/queries/FactSheet.aspx?speciesID=5.

Valéry, L., Fritz, H., Lefeuvre, J.-C., and Simberloff, D. (2009). Invasive species can also be native.... *Trends in Ecology & Evolution*, 24(11):585–585.

Van Den Berg, B. T. (2009). The role of the legal and illegal trade of live birds and avian products in the spread of avian influenza. *Revue Scientifique et Technique de l'OIE*, 28(1):93–111.

Vander Zanden, M. J. and Olden, J. D. (2008). A management framework for preventing the secondary spread of aquatic invasive species. *Canadian Journal of Fisheries and Aquatic Sciences*, 65(7):1512–1522.

Varin, C. (2008). On composite marginal likelihoods. *AStA Advances in Statistical Analysis*, 92(1):1–28.

Varin, C. and Vidoni, P. (2005). A note on composite likelihood inference and model selection. *Biometrika*, 92(3):519–528.

Venzon, D. J. and Moolgavkar, S. H. (1988). A method for computing profile-likelihood-based confidence intervals. *Applied Statistics*, 37(1):87.

Viallefont, A., Lebreton, J.-D., Reboulet, A.-M., and Gory, G. (1998). Parameter identifiability and model selection in capture-recapture models: a numerical approach. *Biometrical Journal: Journal of Mathematical Methods in Biosciences*, 40(3):313–325.

Villa, E. R. and Escobar, L. A. (2006). Using moment generating functions to derive mixture distributions. *The American Statistician*, 60(1):75–80.

Von der Lippe, M. and Kowarik, I. (2007). Long-distance dispersal of plants by vehicles as a driver of plant invasions. *Conservation Biology*, 21(4):986–996.

Waajen, G. W., Van Bruggen, N. C., Pires, L. M. D., Lengkeek, W., and Lürling, M. (2016). Biomanipulation with quagga mussels (*Dreissena rostriformis bugensis*) to control harmful algal blooms in eutrophic urban ponds. *Ecological Engineering*, 90:141–150.

Whittier, T. R., Ringold, P. L., Herlihy, A. T., and Pierson, S. M. (2008). A calcium-based invasion risk assessment for zebra and quagga mussels (*Dreissena spp.*). *Frontiers in Ecology and the Environment*, 6(4):180–184.

Wilson, A. G. (1970). *Entropy in urban and regional modelling*. Number 1 in Monographs in spatial and environmental systems analysis. Pion, London.

Wilson, J. R. U., Ivey, P., Manyama, P., and Nänni, I. (2013). A new national unit for invasive species detection, assessment and eradication planning. *South African Journal of Science*, 109(5/6):1–13.

Wimbush, J., Frischer, M. E., Zarzynski, J. W., and Nierzwicki-Bauer, S. A. (2009). Eradication of colonizing populations of zebra mussels (*Dreissena polymorpha*) by early detection

and SCUBA removal: Lake George, NY. *Aquatic Conservation: Marine and Freshwater Ecosystems*, 19(6):703–713.

Wu, H. and Neale, M. C. (2012). Adjusted confidence intervals for a bounded parameter. *Behavior Genetics*, 42(6):886–898.

Xia, Y., Bjørnstad, O. N., and Grenfell, B. T. (2004). Measles metapopulation dynamics: a gravity model for epidemiological coupling and dynamics. *The American Naturalist*, 164(2):267–281.

Yang, H. and Bell, M. G. H. (1998). Models and algorithms for road network design: a review and some new developments. *Transport Reviews*, 18(3):257–278.

Yokomizo, H., Possingham, H. P., Thomas, M. B., and Buckley, Y. M. (2009). Managing the impact of invasive species: the value of knowing the density–impact curve. *Ecological Applications*, 19(2):376–386.

Youngbull, C. and Devlin, S. (2018). Advances in extraction-free rapid detection of invasive mussel continuous-flow digital droplet PCR in the field - preliminary results from QZAP. Presentation presented at the Columbia River Basin Team Meeting, Portland, OR.

Zimmermann, M., Mai, T., and Frejinger, E. (2017). Bike route choice modeling using GPS data without choice sets of paths. *Transportation Research Part C: Emerging Technologies*, 75:183–196.