Thesis for the degree of Doctor of Philosophy

Combinatorial Semi-Bandit Methods for Navigation of Electric Vehicles

Niklas Åkerblom

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024

Combinatorial Semi-Bandit Methods for Navigation of Electric Vehicles
Niklas Åkerblom
ISBN 978-91-8103-006-8

Acknowledgements, dedications, and similar personal statements in this thesis, reflect the author's own views.

Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie nr 5464
ISSN 0346-718X

Department of Computer Science and Engineering
Chalmers University of Technology
SE–412 96 Gothenburg, Sweden
Telephone + 46 (0) 31 – 772 1000

Typeset by the author using LaTeX.

Printed by Chalmers Digitaltryck
Gothenburg, Sweden 2024

Combinatorial Semi-Bandit Methods for Navigation of Electric Vehicles

Niklas Åkerblom

Department of Computer Science and Engineering
Chalmers University of Technology

Abstract

Climate change is one of the most urgent global challenges humanity is currently facing. As major contributors of greenhouse gas emissions, the transport and automotive sectors have crucial roles to play in solving the problem. To reduce the usage of fossil fuels, electric vehicles need to become more attractive as alternatives to conventional vehicles. Concerns like range anxiety can be mitigated with more accurate navigation systems, especially if such systems are able to sequentially and adaptively collect data to improve their knowledge of the environment.

Hence, this thesis explores a number of different perspectives, settings and methods relating to navigation problems for electric vehicles in uncertain traffic environments. In particular, we focus on a combinatorial multi-armed bandit perspective, since it allows us to adapt and utilize efficient methods for targeted data collection within the navigation setting. Such methods include Bayesian bandit algorithms like Thompson sampling and BayesUCB, which can be used together with prior beliefs informed by domain-specific knowledge to efficiently explore the traffic environment while simultaneously solving the navigation problem.

Throughout the thesis, we apply these kinds of perspectives and methods to various problem settings, including both city-sized and country-sized road networks, relating to online versions of combinatorial optimization problems connected to navigation tasks. Within the appended works, we study the minimization of both expected energy consumption and travel time (including the time required for charging sessions). To show the efficiency of our proposed methods, we perform multiple thorough empirical studies with simulation experiments on realistic problem instances. We also analyze the methods by deriving theoretical upper bounds on their expected regret. With these performance guarantees and results, we aim to demonstrate the utility of the methods for real-world problems and applications.

Keywords: Energy-efficient navigation, online learning, multi-armed bandit problem, Thompson sampling, combinatorial semi-bandit problem.

List of Publications

This thesis is based on the following appended papers:

Paper 1. Niklas Åkerblom, Yuxin Chen, and Morteza Haghir Chehreghani. *Online learning of energy consumption for navigation of electric vehicles.* Artificial Intelligence, 317: 103879, 2023.

Paper 2. Niklas Åkerblom, Fazeleh Sadat Hoseini, and Morteza Haghir Chehreghani. *Online learning of network bottlenecks via minimax paths.* Machine Learning, 112(1), pp. 131-150, 2023.

Paper 3. Niklas Åkerblom, and Morteza Haghir Chehreghani. *A Combinatorial Semi-Bandit Approach to Charging Station Selection for Electric Vehicles.* Transactions on Machine Learning Research, 2023.

Paper 4. Arman Rahbar, **Niklas Åkerblom**, and Morteza Haghir Chehreghani. *Cost-Efficient Online Decision Making: A Combinatorial Multi-Armed Bandit Approach.* Technical report, arXiv:2308.10699. Gothenburg: Chalmers University of Technology, 2024. Submitted.

For both **Paper 2** and **Paper 4**, the first two authors of each paper contributed equally to the works.
The following papers are also co-authored by **Niklas Åkerblom**, but are not appended to the thesis, e.g., since they contain significant overlap with the appended papers (**Paper 1** is a significantly extended version of **Paper 5**) or do not fit with the topic of the thesis:

Paper 5. Niklas Åkerblom, Yuxin Chen, and Morteza Haghir Chehreghani. *An online learning framework for energy-efficient navigation of electric vehicles.* Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, pp. 2051-2057, 2020.

Paper 6. Federica Comuni, Christopher Mészáros, **Niklas Åkerblom**, and Morteza Haghir Chehreghani. *Passive and active learning of driver behavior from electric vehicles.* IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), 2022.

Paper 7. Fazeleh Sadat Hoseini, **Niklas Åkerblom**, and Morteza Haghir Chehreghani. *A contextual combinatorial semi-bandit approach to network bottleneck identification.* Technical report, arXiv:2206.08144. Gothenburg: Chalmers University of Technology, 2022.

Paper 8. Jack Sandberg, **Niklas Åkerblom**, and Morteza Haghir Chehreghani. *Combinatorial Gaussian Process Bandits in Bayesian Settings: Theory and Application for Energy-Efficient Navigation.* Technical report, arXiv:2312.12676. Gothenburg: Chalmers University of Technology, 2023. Submitted.

Paper 9. Tobias Lindroth, Axel Svensson, **Niklas Åkerblom**, Mitra Pourabdollah, and Morteza Haghir Chehreghani. *Online Learning Models for Vehicle Usage Prediction During COVID-19.* IEEE Transactions on Intelligent Transportation Systems, 2024. Accepted for publication.

Paper 10. Hannes Nilsson, Rikard Johansson, **Niklas Åkerblom**, and Morteza Haghir Chehreghani. *Tree Ensembles for Contextual Bandits.* Technical report, arXiv:2402.06963. Gothenburg: Chalmers University of Technology, 2024. Submitted.

Acknowledgments

During this long journey, I have received invaluable help from many people, for which I am incredibly grateful. First, I want to thank my advisor, Morteza Haghir Chehreghani. This thesis would not have been possible without your unwavering support, extensive knowledge and positive attitude. I am also grateful to my co-advisor, Dag Wedelin, and my examiner, Devdatt Dubhashi, for their assistance over the years. Additionally, I would like to thank both of my industrial advisors at Volvo Cars, Viktor Larsson and Rickard Arvidsson, for their mentorship and guidance throughout my doctoral studies. I also want to thank my manager, Ole-Fredrik Dunderberg, as well as my previous managers, Johan, Johan, Eva and Martin.

I have had the privilege of meeting and working with many great new friends and colleagues during this time. It would not have been nearly as rewarding without the intense discussions and fun times with Tobias Johansson, Emil Carlsson and Emilio Jorge. I am also fortunate to have joined the WASP graduate school at the same time as Jonas Nordlöf, whose friendship has made each course, project and study trip something to look forward to. Other people I am grateful to have met or worked with through my PhD studies are: Arman, Fazeleh, Jack, Yuxin, Hampus, Alexandra, Rafael, Shirin, Newton, Hannes, Lovisa, Lena, Linus, Adam, George, Amanda, Karl, Jimmy, Juliette, Russ, Shuangshuang, Angel, Anand, Kolbjörn, Ashkan, Fredrik, Richard, Peter, Birgit, Alexander, and all others I might have missed mentioning.

At Volvo Cars, the support shown by the teams I have belonged to has been very important to me. In particular, I want to thank Anette Westerlund for the help and advice she has given me over the years. Additionally, I am thankful to have had the opportunity to work with, among others, Markus, Erik, Sudhir, Darshan, David, Andreas, Anders, Ellen, Björn, Tobias, Axel, Andreas, Therese, Peter, Eduardo, Ben, Jonas, Jonas, Petter, Mikael, Mikael, Krister, Göran, Sören, Dhananjay, Ahad, Victor, Mitra, Roger, Patrik, Lars-Olof, Yuchu, Ghazaleh, Martin, and Marcus.

I am fortunate to have had the help of many old and good friends, including Jonas, Tobias, Thomas, Tobias and Christoffer. I would also not be where I am today without the love and support of my parents, Inga-Lill and Tommy, as well as my sister, Desirée, and her family, Alice, Wilhelm, and Daniel. I also want to thank Yvonne, Christina and Folke, for their help and encouragement. Finally, I would like to thank Volvo Cars, for giving me the opportunity to pursue a PhD, and the Strategic Vehicle Research and Innovation program (FFI) of Sweden, for funding my research (through the Vinnova / FFI project EENE, reference number: 2018-01937).

Niklas Åkerblom
Gothenburg, February 2024

List of Acronyms

BEV	–	Battery-Electric Vehicle
BUCB	–	Bayesian Upper Confidence Bound (BayesUCB)
CMAB	–	Combinatorial Multi-Armed Bandit / Combinatorial Semi-Bandit
CUCB	–	Combinatorial Upper Confidence Bound
EC^2	–	Equivalent Class Edge Cutting
EV	–	Electric Vehicle
IG	–	Information Gain
LCB	–	Lower Confidence Bound
MAB	–	Multi-Armed Bandit
MDP	–	Markov Decision Process
MST	–	Minimum Spanning Tree
ODM	–	Online Decision Making
PSRL	–	Posterior Sampling for Reinforcement Learning
RL	–	Reinforcement Learning
RCSPP	–	Resource-Constrained Shortest Path Problem
SPP	–	Shortest Path Problem
TS	–	Thompson Sampling
UCB	–	Upper Confidence Bound

Contents

Abstract iii

List of Publications v

Acknowledgments vii

List of Acronyms ix

I Introductory Chapters 1

1 Introduction 3

2 Background 7
2.1 Road network graph model 7
2.2 Shortest path problems 8
2.3 Minimax path problems 10
2.4 Vehicle energy consumption model 12
2.5 Sequential decision-making problems 13
2.6 Multi-armed bandit problems 14
2.7 Multi-armed bandit algorithms 15
2.7.1 Epsilon-greedy 16
2.7.2 Upper confidence bound 16
2.7.3 Thompson sampling 17
2.8 Combinatorial multi-armed bandit problems 18

3 Summary of Included Papers 21
3.1 Paper 1 21
3.2 Paper 2 22
3.3 Paper 3 24
3.4 Paper 4 25

4 Concluding Remarks and Future Work 27

Bibliography 29

II Appended Papers 33

1 Online Learning of Energy Consumption for Navigation of Electric Vehicles 35

1 Introduction . . . 37
1.1 Related work . . . 39
1.2 Our contributions . . . 40
2 Energy consumption model . . . 41
2.1 Setup of the energy consumption model . . . 41
2.2 Rectified Gaussian model of energy consumption . . . 43
2.3 Log-Gaussian model of energy consumption . . . 43
3 Online learning and exploration of the energy model . . . 44
3.1 Shortest path problem as multi-armed bandit . . . 45
3.2 Thompson Sampling . . . 46
3.3 Bayesian Upper Confidence Bound . . . 48
4 Multi-agent learning and exploration . . . 49
4.1 Thompson Sampling with queued delayed feedback . . . 50
4.2 Thompson Sampling with batched feedback . . . 51
5 Experimental results . . . 61
5.1 Real-world experiments . . . 61
5.2 Synthetic networks . . . 69
6 Conclusion . . . 70
A Notation . . . 70
References . . . 73

2 Online Learning of Network Bottlenecks via Minimax Paths 79

1 Introduction . . . 81
2 Bottleneck identification model . . . 83
2.1 Bottleneck identification over a network . . . 83
2.2 Probabilistic model for bottleneck identification . . . 84
3 Online bottleneck learning framework . . . 85
3.1 Thompson Sampling with exact objective . . . 86
3.2 Regret analysis of Thompson Sampling for minimax paths . . . 87
3.3 Thompson Sampling with approximate objective . . . 90
4 Experimental results . . . 93
4.1 Road networks . . . 93
4.2 Collaboration network . . . 96
4.3 Exact objective toy example . . . 97
5 Conclusion . . . 97
A Technical details of regret analysis . . . 98
References . . . 104

3 A Combinatorial Semi-Bandit Approach to Charging Station Selection for Electric Vehicles 109
1 Introduction . . . 111
2 Related work . . . 112
3 Model . . . 113
3.1 Road network graph . . . 113
3.2 Construction of feasibility graph . . . 114
3.3 Probabilistic queue and charging times . . . 116
4 CMAB formulation . . . 118
5 CMAB methods . . . 119
5.1 Epsilon-greedy . . . 119
5.2 Thompson Sampling . . . 120
5.3 BayesUCB . . . 121
6 Experiments . . . 121
6.1 Energy consumption and travel time . . . 122
6.2 Experimental setup . . . 122
6.3 Results . . . 124
7 Discussion and conclusion . . . 126
A Appendix . . . 127
References . . . 129

4 Cost-Efficient Online Decision Making: A Combinatorial Multi-Armed Bandit Approach 135
1 Introduction . . . 137
2 Related work . . . 139
3 Problem formulation . . . 139
4 CMAB methods for ODM . . . 141
4.1 Cost-efficient approximate oracles . . . 142
5 Theoretical analysis . . . 144
6 Experiments . . . 148
6.1 Experimental results . . . 149
6.2 Extension to real-valued test outcomes . . . 149
6.3 Application to online troubleshooting . . . 150
7 Conclusion . . . 150
A Additional proofs . . . 151
A.1 Proof of Lemma 2 . . . 151
A.2 Proof of Lemma 3 . . . 152
A.3 Proof of Lemma 4 . . . 152
B Additional experimental results . . . 153
References . . . 155

Part I

Introductory Chapters

Chapter 1

Introduction

The ongoing climate change resulting from historical and current emissions of greenhouse gases is one of the most urgent global challenges for humanity to overcome. To reduce the rate of emissions, international organizations like the European Union (EU) have set up short-term and long-term targets for their member states to fulfill. The EU, in particular, has specified a goal of decreasing the net emissions of greenhouse gases within the union to zero until 2050 (European Commission 2019). The transportation sector has an important role to play in reaching these targets, since it is, both historically and presently, a major source of greenhouse gas emissions. Recent technological advances within vehicle electrification and connectivity, as well as increasingly efficient machine learning and artificial intelligence methods, can help the industry to address this issue.

Many vehicle manufacturers currently strive to transition from using internal combustion engines for propulsion to instead use electric machines. Among other technologies, battery-electric vehicles (BEVs) utilizing lithium-ion batteries as their sole energy storage medium (in contrast to hybrid vehicles with combinations of energy sources) are popular alternatives to conventional vehicles. However, potential customers often hesitate to buy BEVs since they are concerned about the maximum driving range of such vehicles, a phenomenon called *range anxiety* (Rauh et al. 2015). Several different disturbance factors can increase the risk of being stranded with a depleted battery before reaching an intended destination, such as certain weather conditions, unexpected detours due to road works, low availability of charging stations, and traffic congestion. Moreover, poorly chosen charging stops during a long-distance trip might significantly increase the total travel time.

In the long-term, these problems may be alleviated with, e.g., batteries with higher energy capacity, or expansion of the charging infrastructure. In the near future, however, a relatively low-cost way of increasing the trust of drivers is to provide them with sufficient information to plan their trips so that risks are minimized. An in-vehicle navigation system for a BEV may take both energy consumption and travel time into account, in addition to finding appropriate charging stations when necessary. This kind of system requires not only computationally efficient and robust algorithms and energy consumption models, but also sufficient data about the road network and traffic environment to ensure high accuracy.

Well-known shortest path algorithms, e.g., Dijkstra's algorithm (Dijkstra 1959), A*-search (Hart et al. 1968) and the Bellman-Ford algorithm (Shimbel 1954; Ford Jr 1956; Bellman 1958), may be adapted to minimize vehicle energy consumption instead of travel time. For example, Sachenbacher et al. (2011) develop a heuristic function which estimates the energy consumed between nodes in a road network, and use it together with a modified variant of A*-search which also respects battery capacity constraints. The main focus of their work, as well as that similar works (e.g., Artmeier et al. 2010; Baum et al. 2017), is to minimize the run-time and computational resources required to find feasible paths through the road network.

The aforementioned and most other existing methods assume complete knowledge of the traffic environment, but with internet connections being increasingly ubiquitous in vehicles, data can be continuously collected to improve the quality of the planned paths. By taking a probabilistic view, it is possible to not only update estimates of travel time and energy consumption distribution parameters using observations, but also to consider the uncertainties involved to find paths which are more robust to unexpected external disturbances. For example, risk-averse drivers may want to find a path which minimizes the total travel time, while simultaneously ensuring that the remaining battery energy exceeds a safety threshold with high probability (B. Y. Chen et al. 2013).

Balancing exploration of an unknown environment to gain new knowledge and exploitation of that knowledge to achieve a more long-term objective is a classical problem in the field of machine learning. It is no less present when the objective is to learn enough information to select optimal paths, since learning more information about a particular path does not, in itself, provide any guarantees of an expected improvement in the quality of the found paths. Furthermore, vehicle-specific characteristics and parameters may affect both energy consumption and charging, which can limit the number of vehicles viable for data collection. Hence, to guarantee such an improvement, it is necessary to explicitly combine the learning method with a sufficient degree of efficient exploration.

The *combinatorial multi-armed bandit* (CMAB) problem (Nicolò Cesa-Bianchi and Lugosi 2012), an extension of the classical *multi-armed bandit* (MAB) problem, provides a way of modeling this trade-off *specifically* for combinatorial optimization problems where the underlying parameters are unknown and need to be learned. In this thesis, we introduce various ways of approaching BEV navigation problems using CMAB methods. In particular, we focus on the *Bayesian* CMAB setting, since it allows us to utilize prior knowledge to explore the environment more efficiently.

The structure of the first part of the thesis is as follows. In order to provide relevant background knowledge, Chapter 2 introduces and explains several topics and concepts which are beneficial for understanding the appended papers. Chapter 3 gives a summary of the contents and most important contributions of each appended paper. Finally, Chapter 4 includes concluding remarks on the research project, as well as a few suggestions of possible directions for future research.

The second part of the thesis contains four appended papers (further described in Chapter 3), which together attempt to address the question of *how to, with limited data collection resources available (e.g., vehicles with appropriate sensors), efficiently*

collect information (e.g., energy consumption, charging time) through interactions with the environment, for improved navigation of battery-electric vehicles:

- Paper 1 (Åkerblom, Y. Chen, and Haghir Chehreghani 2023), which is an extended journal version of an earlier conference paper (Åkerblom, Y. Chen, and Haghir Chehreghani 2020), introduces an *online learning framework* utilizing CMAB methods for energy-efficient navigation of battery-electric vehicles.
- Paper 2 (Åkerblom, Hoseini, et al. 2023) extends the framework and methods from Paper 1 to address the problem of network bottleneck identification and avoidance in stochastic transport networks.
- Whereas the methods in Paper 1 and Paper 2 are only applied to navigation problems in city-sized road networks, Paper 3 (Åkerblom and Haghir Chehreghani 2023) focuses on the larger problem of long-distance BEV navigation in country-sized road networks where charging is necessary, whether once or multiple times.
- Finally, Paper 4 (Rahbar et al. 2023) introduces a novel framework which utilizes Bayesian CMAB methods for cost-efficient online decision-making and adaptive information acquisition. This may be used for addressing several different problem domains, including interactive identification of driver charging station preferences.

Chapter 2

Background

In this chapter, we introduce some important notions and background beneficial for understanding the rest of the thesis.

2.1 Road network graph model

A road network typically consists of a vast number of roads connected through intersections. These roads are of various types (e.g., highways, arterial roads and residential streets) which coexist with parallel networks for other modes of transportation, including walkways, bikeways, railways and waterways. Various rules and restrictions apply to vehicles traveling through the network, such as speed limits and turn restrictions. To be able to address navigation problems, a *graph structure* is a useful and common way to model the road network.

A graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ consists of a set of vertices (or nodes) $\mathcal{V}$ and a set of edges (or links) $\mathcal{E}$ connecting them. We let $\mathcal{V}_\mathcal{G}$ and $\mathcal{E}_\mathcal{G}$ denote the vertex set and edge set, respectively, associated with the graph $\mathcal{G}$, but omit the subscripts whenever this is obvious from the context. Each edge $e \in \mathcal{E}$ is characterized by a pair of vertices $(u_1, u_2) \in \mathcal{V} \times \mathcal{V}$. A graph can be either *directed* or *undirected*, indicating whether the pair of vertices for each edge $(u_1, u_2) \in \mathcal{E}$ is ordered, i.e., $(u_1, u_2) \neq (u_2, u_1)$. For the road network, $\mathcal{V}$ corresponds to the set of intersections and $\mathcal{E}$ represents the roads between them. If we also want to model, e.g., turn restrictions, we can split intersections into multiple vertices and edges for all maneuvers allowed. Since roads generally have specified driving directions, we mainly consider directed graphs in this thesis. An example of a graph is shown in Figure 2.1.

A *path* $\boldsymbol{p} = \langle u_1, \ldots, u_n \rangle$ of length n is a sequence vertices $u_1, \ldots, u_n \in \mathcal{V}$ connected by an unbroken sequence of edges $(u_1, u_2), (u_2, u_3), \ldots, (u_{n-1}, u_n) \in \mathcal{E}$. A *cycle* is a path which starts and ends in the same vertex. A graph $\mathcal{G}'(\mathcal{V}', \mathcal{E}')$ is a *subgraph* of a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ if $\mathcal{V}' \subseteq \mathcal{V}$ and $\mathcal{E}' \subseteq \mathcal{E}$. Hence, a path through $\mathcal{G}$ can also be viewed as a subgraph of $\mathcal{G}$. A graph $\mathcal{G}$ is *connected* if, for every pair of vertices $u, u' \in \mathcal{V}$, u' can be reached from u using a path through $\mathcal{G}$. A *tree* is a connected graph which contains no cycles. A *spanning tree* of an undirected graph $\mathcal{G}$ is a connected subgraph of $\mathcal{G}$ which is both a tree and contains every vertex of $\mathcal{G}$.

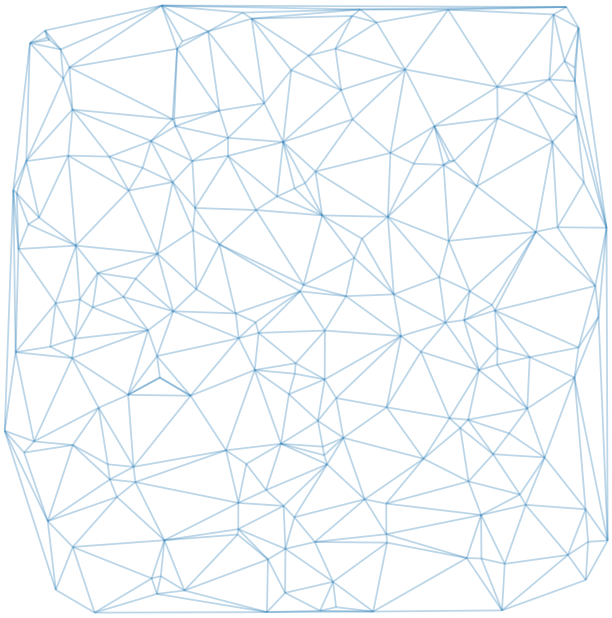


Figure 2.1: An example of a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, with lines representing edges between vertices.

Each vertex $u \in \mathcal{V}$ and edge $e \in \mathcal{E}$ may have attributes associated with them, with the edge *weight* w_e being the most important edge attribute associated with navigation problems (and general shortest path problems). Hence, we indicate a *weighted* graph, with a vector of weights $\boldsymbol{w}$ consisting of the weights w_e for all edges $e \in \mathcal{E}$, by $\mathcal{G}(\mathcal{V}, \mathcal{E}, \boldsymbol{w})$. Examples of relevant edge attributes in a road network graph can be the length, the slope and the speed limit. Depending on the problem setting, either the length or the travel time is often selected as the weight attribute of the edges.

2.2 Shortest path problems

Given the current position of a vehicle and the position of an intended destination, an in-vehicle navigation system attempts to solve the problem of providing the driver with a set of instructions for how to reach the destination, while minimizing the total travel time. A more general form of this problem, which may be formally defined using the notions introduced in Section 2.1, is the *shortest path problem*. Given a directed and weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \boldsymbol{w})$, a source vertex $u_{\text{src}} \in \mathcal{V}$ and a target vertex $u_{\text{trg}} \in \mathcal{V}$, we denote the set of all paths through $\mathcal{G}$ from u_{src} to u_{trg} as $\mathcal{P}_{(u_{\text{src}}, u_{\text{trg}})}$. Then, interpreting each path $\boldsymbol{p} \in \mathcal{P}_{(u_{\text{src}}, u_{\text{trg}})}$ as a set of edges, the shortest path problem is to find a path $\boldsymbol{p}^*$ such that

$$\boldsymbol{p}^* = \arg \min_{\boldsymbol{p} \in \mathcal{P}_{(u_{\text{src}}, u_{\text{trg}})}} \sum_{e \in \boldsymbol{p}} w_e \, . \tag{2.2.1}$$

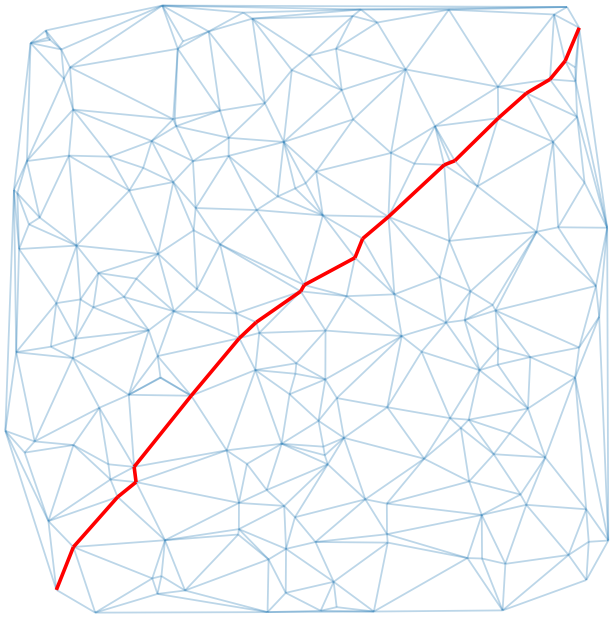


Figure 2.2: Shortest path (in red) through a weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \boldsymbol{w})$, with the lengths of the edges (i.e., the Euclidean distance between the incident vertices of each edge) used as edge weights.

The problem defined in Eq. 2.2.1 has the property of *optimal substructure*, in the sense that for any pair of vertices v_i and v_j in any shortest path $\boldsymbol{p}^*$ through a graph $\mathcal{G}$, the *subpath* in $\boldsymbol{p}^*$ between v_i and v_j is also the shortest path in $\mathcal{G}$ between v_i and v_j. This property enables the shortest path problem to be solved efficiently by combining solutions for subproblems, with a technique commonly called *dynamic programming* (DP). The solution for a pair of vertices in the graph of Figure 2.1 is shown in Figure 2.2.

One well-known classical algorithm for the shortest path problem which utilizes a DP approach is Dijkstra's algorithm (Dijkstra 1959). For each vertex $u \in \mathcal{V}$, starting with a given source vertex $u_{\text{src}} \in \mathcal{V}$, the algorithm computes and stores the total weight of the shortest path from u_{src} to u. For each $u \in \mathcal{V}$, it also stores the immediately preceding vertex on the shortest path to u. Given the set of predecessors to all vertices, finding the shortest path from u_{src} to u_{trg} is as simple as traversing the graph backwards through predecessors, starting with the predecessor of u_{trg}.

While pre-computing all shortest paths (for a single source vertex) is useful for some applications, especially if the weights are fixed, in other cases it might be more interesting to stop the computation earlier, when the shortest path to the target has been found. Dijkstra's algorithm performs iterative computations on vertices selected uniformly in an expanding "circle" around the the starting vertex. If early stopping is the intention, there are more efficient ways of doing this, such as the A* algorithm (Hart et al. 1968). It essentially follows the same approach as Dijkstra's algorithm, with the difference that for each vertex, in addition to the weight of the shortest path to that vertex from the source vertex, it also computes an estimate of

the remaining distance (total weight of the shortest path) to the target vertex. It uses the sum of these two quantities to determine which vertex to try next, resulting in an approach which can be described as a *best-first search* (see e.g., Dechter and Pearl 1985). The efficiency of A* is closely tied to which *heuristic function* is used to estimate the distance remaining to the target.

In contrast to A* and Dijkstra's algorithm, the Bellman-Ford algorithm (Shimbel 1954; Ford Jr 1956; Bellman 1958) is able to handle the presence of negative edge weights in the graph. For some applications, including in-vehicle navigation functions, these algorithms may be insufficiently fast, especially when large-scale graphs or more complicated variants of the problem are considered. It is possible to transform large graphs in various ways to decrease the time required to solve the problem in real-time, e.g., by creating *shortcut edges* between distant vertices. One such approach is called *contraction hierarchies* (Geisberger et al. 2012), which has also been used for navigation of electric vehicles, with limited battery capacity, between charging stations (Baum et al. 2017).

2.3 Minimax path problems

Another interesting problem, closely related to the shortest path problem, is the *minimax path problem*. Again, given a weighted (though not necessarily directed) graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \boldsymbol{w})$, a source vertex $u_{\text{src}} \in \mathcal{V}$ and a target vertex $u_{\text{trg}} \in \mathcal{V}$, we denote the set of all paths through $\mathcal{G}$ between u_{src} to u_{trg} and $\mathcal{P}_{(u_{\text{src}}, u_{\text{trg}})}$. Then the minimax path problem is defined as

$$\boldsymbol{p}^* = \arg \min_{\boldsymbol{p} \in \mathcal{P}_{(u_{\text{src}}, u_{\text{trg}})}} \max_{e \in \boldsymbol{p}} w_e \,. \tag{2.3.1}$$

An example of a minimax path between a pair of vertices in the graph of Figure 2.1 is shown in Figure 2.3. The edge attaining the maximum weight in $\boldsymbol{p}^*$ can be called a *bottleneck*, since any alternative path through the graph is guaranteed to have a maximum edge weight which is at least as high as the weight of the bottleneck. Hence, the problem is also referred to as the *bottleneck shortest path problem*, which is equivalent with the *maximin path problem* if the weights are negated. The maximin path problem, where the path minimizing the maximum edge weight should be found, is also called the *widest path problem* when the weights are interpreted as capacities.

The objective of identifying bottlenecks (or minimax paths) has important applications in several different domains, including computer communication network routing (Shacham 1992) and planning of service vehicles (e.g., ambulances, fire trucks or police cars) (Berman and Handler 1987). The authors of the latter work, as an example, study the problem of routing service vehicles to non-service locations (e.g., depots) in a way which maximizes their availability. To address the problem, their approach is to use a modified variant of Dijkstra's algorithm to find a path through the road network which minimizes the maximum travel time to any vertex with a potential service demand (where the travel time is weighted with an estimate of the demand).

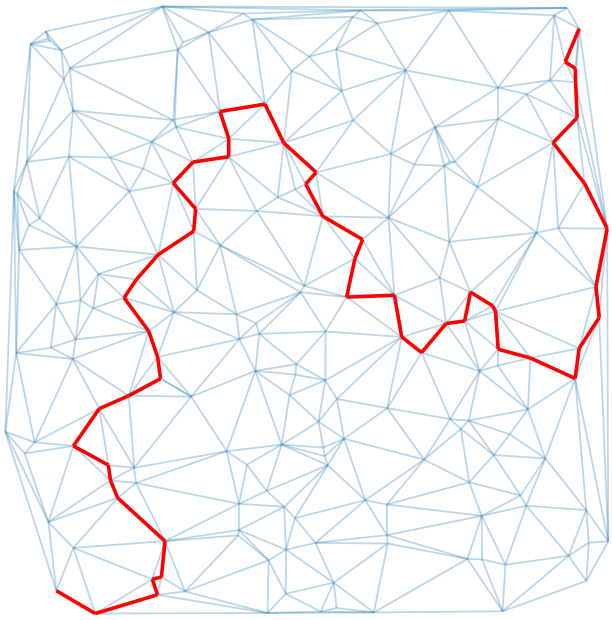


Figure 2.3: Minimax path (in red) through a weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \boldsymbol{w})$, with the lengths of the edges (i.e., the Euclidean distance between the incident vertices of each edge) used as edge weights.

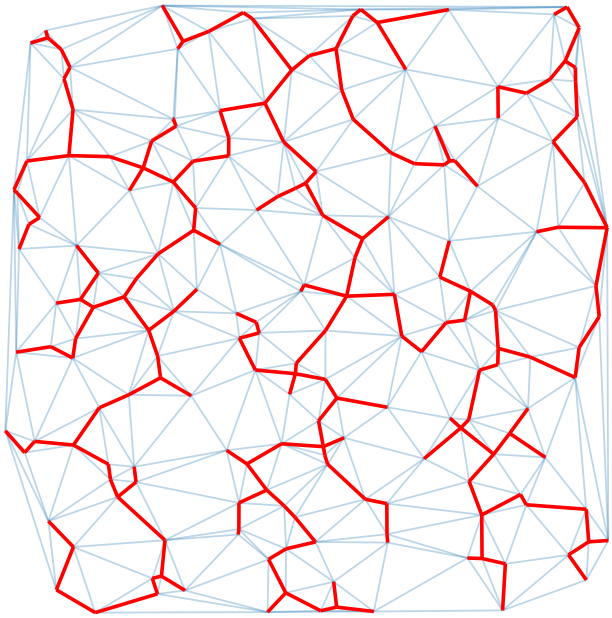


Figure 2.4: Minimum spanning tree (in red) of a weighted undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \boldsymbol{w})$, with the lengths of the edges (i.e., the Euclidean distance between the incident vertices of each edge) used as edge weights.

While the modified version of Dijkstra's algorithm by Berman and Handler (1987) may be used whether or not a graph is directed, there are advantages to considering alternative methods for undirected graphs. Given an undirected weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \boldsymbol{w})$ and the set $\mathcal{T}_\mathcal{G}$ of all spanning trees of $\mathcal{G}$, a *minimum spanning tree* (MST) $\mathcal{T}_{\min}$ of $\mathcal{G}$ is defined as $\mathcal{T}_{\min} \triangleq \arg\min_{\mathcal{T}\in\mathcal{T}_\mathcal{G}} \sum_{e\in\mathcal{E}_\mathcal{T}} w_e$ (an example of which is shown in Figure 2.4). Then, a classical result by Hu (1961) states that any path through $\mathcal{T}_{\min}$ is also a minimax path through $\mathcal{G}$. To find $\mathcal{T}_{\min}$, we can use well-known methods like Kruskal's algorithm (Kruskal 1956) or Prim's algorithm (Prim 1957), where the latter achieves a worst-case time complexity of $\mathcal{O}(|\mathcal{E}| + |\mathcal{V}| \log |\mathcal{V}|)$ when implemented using Fibonacci heaps (Fredman and Tarjan 1987). Though Dijkstra's algorithm can also be implemented using a priority queue based on Fibonacci heaps for a time complexity of $\mathcal{O}(|\mathcal{E}| + |\mathcal{V}| \log |\mathcal{V}|)$, the algorithm needs to be rerun every time the source vertex changes (or, like in Johnson's algorithm (Johnson 1977), be run once for each possible source vertex). In contrast, for any pair of source and target vertices, traversal of the MST $\mathcal{T}_{\min}$ has time complexity $\mathcal{O}(|\mathcal{V}_{\mathcal{T}_{\min}}| + |\mathcal{E}_{\mathcal{T}_{\min}}|) \leq \mathcal{O}(|\mathcal{V}|)$ since $|\mathcal{V}_{\mathcal{T}_{\min}}| = |\mathcal{V}|$ and $|\mathcal{E}_{\mathcal{T}_{\min}}| \leq |\mathcal{V}|$, making it a good alternative to Dijkstra's algorithm in undirected graphs containing many more edges than vertices.

2.4 Vehicle energy consumption model

For navigation problems involving electric vehicles, the energy consumed during motion can be used as either a weight (instead of, or together with, travel time) or a constraint (i.e., the energy consumed should not exceed the available battery capacity). Depending on the intended use and other requirements, energy consumption tied to specific road segments in a transportation network can also be modeled in various ways. Certain frameworks for high fidelity vehicle simulations may use very detailed and accurate models, which can be unsuitable for assigning weights to edges in a road network graph due to the long computation times required. Moreover, such models often depend on the availability of fine-grained elevation and speed data, which might not be feasible to provide in a navigation scenario.

Consequently, we use a simpler (but well-established, see e.g., Guzzella, Sciarretta, et al. 2007) point mass model for vehicle energy consumption throughout this thesis. Figure 2.5 shows a vehicle in motion and the main longitudinal forces acting on it. In order for the vehicle to accelerate (uphill, in this case), the traction force F_t has to overcome the resistive forces of gravity F_g, rolling resistance F_r and aerodynamic drag F_a. In other words, with vehicle mass m and acceleration $\dot{v}$, the following equation holds:

$$m\dot{v} = F_t - F_g - F_r - F_a. \tag{2.4.1}$$

The data sets used in this thesis do not contain detailed road-specific acceleration and deceleration information. Hence, we let the left-hand side ($m\dot{v}$) in Eq. 2.4.1 be zero and assume that the speed is constant (which might result in an underestimation of the energy consumption, in the worst case). The second term in the right-hand side is the gravitational force (specifically, the longitudinal component), where

$$F_g = mg \sin\alpha, \tag{2.4.2}$$

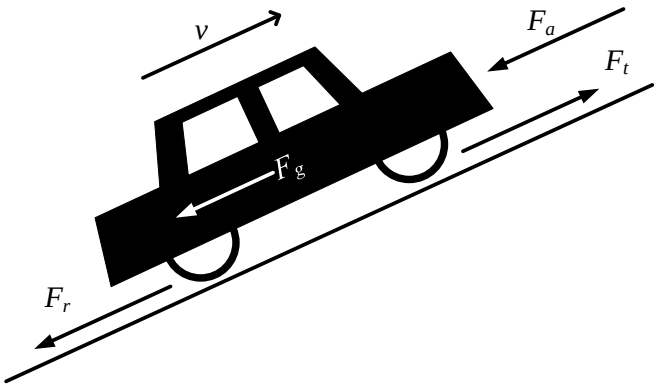


Figure 2.5: Longitudinal forces acting on a vehicle during motion.

with gravitational acceleration g and road slope angle α. The third right-hand side term of Eq. 2.4.1 is the rolling friction force

$$F_r = C_r mg \cos \alpha, \tag{2.4.3}$$

where the rolling resistance coefficient C_r depends on properties of the tires and road surface. The fourth and final right-hand side term of Eq. 2.4.1 is the aerodynamic friction force

$$F_a = \frac{1}{2} \rho A C_d v^2, \tag{2.4.4}$$

with the air density ρ, vehicle frontal surface area A, air drag coefficient C_d (depending on the shape of the vehicle) and squared speed v^2.

To obtain the energy consumption E, we start by rearranging and expanding all terms in Eq. 2.4.1 and multiplying them with the vehicle speed v to get the mechanical power, which we can then integrate with respect to time so that we get

$$E = \frac{l}{\eta} \left(mg \sin \alpha + C_r mg \cos \alpha + \frac{1}{2} \rho A C_d v^2 \right), \tag{2.4.5}$$

where l is the road segment length and η is the efficiency of the powertrain for conversion from electrical to mechanical energy.

2.5 Sequential decision-making problems

In this thesis, we study electric vehicle navigation problems from the perspective of *sequential decision-making under uncertainty*. Specifically, the task of repeatedly selecting paths through a road network (either as a single vehicle, over time, or multiple vehicles, in aggregate) can be seen as sequential interactions of an agent with an uncertain environment. Let Π be a set of available *policies* specifying the way in which an agent can interact with the environment. When addressing a problem like

this, the objective is to select a policy $\pi \in \Pi$ which maximizes some sort of long-term reward granted to the agent by the environment as a result of the performed actions. The policy may select actions both according to the current state of the environment and considering the results of earlier interactions, and can be either stochastic or deterministic in nature.

Algorithm 1 General sequential decision-making problem

Input: Interaction policy $\pi \in \Pi$

1: **for** time step $t \leftarrow 1, \ldots, T$ **do**
2: $S_t \leftarrow$ Environment reveals the current state $S_t \in \mathcal{S}$ to agent.
3: $A_t \leftarrow$ Agent performs an action $A_t \in \mathcal{A}(S_t)$ selected according to policy π.
4: $R_t \leftarrow$ Environment grants a reward $R_t \in \mathcal{R}(S_t, A_t)$ to agent.
5: Environment enters next state $S_{t+1} \in \mathcal{S}$.

In Algorithm 1, we show an outline of the protocol of interactions between the agent and the environment in a general sequential decision-making problem. For each time step t until a specified time horizon T, the interaction occurs in the following way. First, information on the current state $S_t \in \mathcal{S}$ of the environment may be revealed to the agent. The agent then uses the policy π to select and perform an action $A_t \in \mathcal{A}(S_t)$ (where the set of available actions may depend on time step and the current state of the environment). As a result of the action, the environment may then give the agent a reward $R_t \in \mathcal{R}(S_t, A_t)$ (and possibly other types of feedback or information relating to the action, which can be utilized by the policy). Finally, the environment transitions to the next state $S_{t+1} \in \mathcal{S}$, potentially as a consequence of the performed action.

This interaction protocol is sufficiently general to cover many different types of problems, and depending on the specific problem considered, the state space $\mathcal{S}$, action space $\mathcal{A}$ and reward space $\mathcal{R}$ can be either continuous or discrete. An important problem, for any agent acting according to a policy which adapts to earlier observations revealed by the environment, is how to balance between *exploring* the environment and *exploiting* previously acquired knowledge to increase the rewards received.

2.6 Multi-armed bandit problems

The class of problems introduced and outlined in Section 2.5 can be modeled in many different ways and with varying levels of detail. If we are interested in how the actions of the agent affect and change the state of the environment, we can model the environment and interactions using *Markov decision processes* (MDPs). This allows us to utilize *reinforcement learning* (RL) methods to learn and improve policies, by interacting with the environment to observe state transitions and rewards (see e.g., Sutton and Barto 2018). However, in settings where significant state transitions directly due to the actions of the agent are either unlikely to occur or not interesting to consider, there may be benefits to using a simplified model of the decision-making problem.

A classical way of modeling (exclusively) the aspect of the problem relating to the trade-off between exploration and exploitation is the *multi-armed bandit* (MAB) problem. In a MAB problem, state transitions are generally ignored since they are assumed to not be caused by the actions of the agent (to any significant degree), though *contextual* information relating to the state may sometimes be available to the agent. Hence, the main objective for a MAB agent is to learn the relationship between actions (often referred to as *arms*) and rewards, as outlined in Algorithm 2.

Algorithm 2 Multi-armed bandit (MAB) problem

Input: MAB algorithm π

1: **for** time step $t \leftarrow 1, \ldots, T$ **do**
2: $a_t \leftarrow$ Agent selects ("plays") arm $a_t \in \mathcal{A}$ in accordance with algorithm π.
3: $r_t(a_t) \leftarrow$ Environment gives agent reward $r_t(a_t) \in \mathcal{R}$.
4: Algorithm π updates knowledge using observed reward $r_t(a_t)$.

The MAB problem was likely first studied by Thompson (1933), though the name (referring to the example of a gambler trying to find the best slot machine or "one-armed bandit" in a casino through repeated trials) was not coined until years later. Throughout this thesis, we are exclusively concerned with *stochastic* bandit problems, where the environment is assumed to draw the reward $r_t(a_t) \in \mathcal{R}$ (for the arm $a_t \in \mathcal{A}$ played by the agent at time step t) from a *fixed* and *unknown* (by the agent) probability distribution associated with a_t. Alternatively, one may consider *adversarial* bandit problems (see e.g., Auer, Nicolo Cesa-Bianchi, Freund, et al. 1995) with weaker assumptions on how the rewards are generated.

The goal of a MAB algorithm is to select and play arms which maximize the expected sum of rewards received until a considered (though not necessarily known in advance by the agent) time horizon T. The reward maximization problem is usually reformulated as an equivalent *regret* minimization objective, with the regret defined as

$$\text{Regret}(T) \triangleq \mathbb{E}\left[\sum_{t\in[T]} (\mu_{a^*} - \mu_{a_t})\right], \tag{2.6.1}$$

where μ_a denotes the expected value of the reward distribution of arm $a \in A$, and $a^* \triangleq \arg\max_{a\in\mathcal{A}} \mu_a$. In other words, the regret is the expected sum, over time, of the difference in mean reward between the best arm and the played arm. The outer expectation in Eq. 2.6.1 is taken over anything random in how the MAB algorithm selects arms (which might also depend on the random rewards of previously played arms).

2.7 Multi-armed bandit algorithms

The naïve approach to address the stochastic MAB problem is the *greedy* method, which plays the arm with the highest estimated mean reward in each time step, with the estimates being continuously updated with observed rewards from played arms. It is easy to see that if an arm which is played early exhibits high reward

observations compared to the initial estimates of the other arms, this method will commit to that (likely sub-optimal) arm.

2.7.1 Epsilon-greedy

One way to improve the greedy method is to carefully introduce some random exploration of arms other than the one with the highest estimated mean reward. One such method is *ϵ-greedy* exploration (see Sutton and Barto 2018), where in each time step, with probability ϵ, a uniformly random arm is played. Otherwise (i.e., with probability $1 - \epsilon$), the method plays the arm with the highest mean reward estimate. While the regret received during the "greedy" time steps will decrease over time as more observations are collected from the environment, having a constant ϵ means that regret will continue to be incurred during the "exploration" time steps.

Algorithm 3 ϵ_t-greedy

Input: Parameters $c > 0$ and $0 < d \leq \min_{a \in \mathcal{A}: a \neq a^*} \mu^* - \mu_a$

for time step $t \leftarrow 1, \ldots, T$ **do**
  $\epsilon_t \leftarrow \min\left\{1, \frac{c|\mathcal{A}|}{td^2}\right\}$.
  $x \leftarrow$ Sample $x \sim$ Bernoulli(0.5).
  if $x \leq \epsilon_t$ **then**
    $a_t \leftarrow$ Sample arm $a_t \sim \mathcal{A}$ uniformly at random.
  else
    $a_t \leftarrow \arg\max_{a \in \mathcal{A}} \hat{\mu}_{a,t-1}$ (where $\hat{\mu}_{a,t-1}$ is the current average reward of a).
  Play arm a_t and receive reward $r_t(a_t)$.
  Update average reward $\hat{\mu}_{a_t,t}$ of arm a_t with reward $r_t(a_t)$.

A slight modification of the approach, called *ϵ_t-greedy*, is to decrease the exploration probability ϵ_t with each time step t. Auer, Nicolo Cesa-Bianchi, and Fischer (2002) show a sub-linear upper bound on the regret (w.r.t. the horizon T) of the method (outlined in Algorithm 3) if ϵ_t decreases with a rate of $1/t$. Intuitively, since the reward distributions in the stochastic MAB setting are assumed to be fixed, it is efficient to explore less when enough information has been collected by the agent.

2.7.2 Upper confidence bound

There is a principle for sequential decision-making called *optimism in the face of uncertainty*, which states that it is beneficial to take *optimistic* (though plausibly good) decisions in an uncertain environment. A class of algorithms based on this principle, collectively referred to as *upper confidence bound* (UCB) methods (see Lai 1987; R. Agrawal 1995; Auer, Nicolo Cesa-Bianchi, and Fischer 2002; Auer 2002), induces exploration by adding an exploration term (confidence width) to the mean reward estimate of each arm. In each time step, the algorithm then finds and plays the arm maximizing this sum (which should be higher than the expected reward of each arm with high probability).

To avoid excessive exploration, the confidence width is usually selected as a decreasing function w.r.t. the number of plays of each arm. To ensure that it is an upper bound for the unknown mean reward with high probability, it is often derived using concentration inequalities for the assumed reward distributions (see e.g., Auer, Nicolo Cesa-Bianchi, and Fischer 2002). Here, the intuition is that if an arm is selected by the algorithm and played, it is either due to it having been played few times so far or that it has a high average reward. Since the average will concentrate around the unknown mean with more reward observations, a sub-optimal arm which has initially been selected optimistically will eventually be discarded by the algorithm. The UCB method (specifically, UCB1 for reward distributions with bounded support, as defined by Auer, Nicolo Cesa-Bianchi, and Fischer 2002) is outlined in Algorithm 4.

Algorithm 4 UCB1

1: Play each arm $a \in \mathcal{A}$ once and observe the rewards.
2: **for** time step $t \leftarrow 1, \ldots, T$ **do**
3: **for** $a \in \mathcal{A}$ **do**
4: $U_t(a) \leftarrow \hat{\mu}_{a,t-1} + \sqrt{\frac{2t}{N_{t-1}(a)}}$ (where $N_{t-1}(a)$ is the current number of plays of a).
5: $a_t \leftarrow \arg\max_{a \in \mathcal{A}} U_t(a)$.
6: Play arm a_t and receive reward $r_t(a_t)$.
7: Update average reward $\hat{\mu}_{a_t,t}$ of arm a_t with reward $r_t(a_t)$.

2.7.3 Thompson sampling

Thompson sampling (TS) (Thompson 1933), is likely the oldest method for MAB problems, but has mostly been forgotten until the last few decades. It has also been called *posterior sampling* and *probability matching*. The latter name gives an intuition on one of the central ideas behind TS. Like with ϵ-greedy, played arms are randomly sampled, but the probability that each arm will be selected is *matched* with the probability that the arm is the best arm. The impressive performance of Thompson sampling on many problems has been evaluated extensively through experimental studies (e.g., Chapelle and Li 2011; Graepel et al. 2010) and theoretical analyses (e.g., S. Agrawal and Goyal 2012; Kaufmann, Korda, et al. 2012; Russo and Van Roy 2014), with the method often outperforming other methods (like UCB).

In practice, this is accomplished through a Bayesian assumption of a prior distribution over reward distribution parameters. The prior distribution is used together with the observed rewards of played arms to compute posterior distributions. In each time step, as outlined in Algorithm 5, samples are drawn from the posterior distributions associated with all arms. Then, the arm with maximum expected reward with respect to the sampled parameters is selected and played.

Algorithm 5 Thompson sampling

Input: Prior distribution $\lambda_{a,0}$ over the reward mean μ_a of each arm $a \in \mathcal{A}$.

1: **for** time step $t \leftarrow 1, \ldots, T$ **do**
2: **for** $a \in \mathcal{A}$ **do**
3: $\tilde{\mu}_a \leftarrow$ Sample $\tilde{\mu}_a$ from posterior distribution $\lambda_{a,t-1}$.
4: $a_t \leftarrow \arg\max_{a \in \mathcal{A}} \tilde{\mu}_a$.
5: Play arm a_t and receive reward $r_t(a_t)$.
6: Update posterior distribution $\lambda_{a_t,t}$ of arm a_t with reward $r_t(a_t)$.

2.8 Combinatorial multi-armed bandit problems

When the size of the set of arms is very large, it can be infeasible to use the methods described in Section 2.7 directly to address the MAB problem, since they depend on collecting observations and computing estimates for each individual arm. However, sometimes there is an underlying structure to the problem that we can utilize to learn about more than a single arm during each time step. For example, in a *linear bandit problem* (see e.g., Auer 2002; Dani et al. 2008), a linear relationship is assumed to exist between expected rewards and feature vectors associated with each arm. Hence, it is sufficient to select arms in order to estimate the coefficients of the linear function rather than the individual mean rewards of all arms.

The *combinatorial multi-armed bandit* (CMAB) problem (Nicolò Cesa-Bianchi and Lugosi 2012) is another example of such an extension of the MAB problem with assumptions of an underlying relationship between arms, which may be utilized for more efficient exploration. In a CMAB problem, each arm consists of a set of elements where the mean reward of the arm is assumed to be a function of the elements (and associated parameters) in the set. Then, when we play an arm, we can hopefully also learn something about different arms with overlapping elements.

More concretely, we call each such element a *base arm*, and denote the set of all base arms $\mathcal{A}$ (relating to the notation used for the standard MAB problem in Section 2.6). To more clearly distinguish the individual base arms from the played arms (i.e., sets of base arms), we refer to each playable set of base arms $\boldsymbol{a} \subseteq \mathcal{A}$ as a *super-arm*. If some kind of feedback can be observed for each of the base arms in a played super-arm, we say that CMAB has *semi-bandit feedback*, in contrast to the *bandit feedback* case where only a single reward value is revealed for each played super-arm (similar to the standard MAB setting).

The CMAB problem formulation (especially with semi-bandit feedback) may be used for combinatorial optimization problems with uncertain environments where repeated trials are possible, e.g., shortest path problems with stochastic edge weights (and unknown parameters). In such problems, the set of feasible solutions is often constrained, such as the set of all paths $\mathcal{P}_{(u_{\text{src}},u_{\text{trg}})}$ between a source vertex u_{src} and a target vertex u_{trg}, as described in Section 2.1. In the corresponding CMAB problem, super-arms are selected from a set of *feasible* super-arms $\mathcal{I} \subseteq 2^{\mathcal{A}}$ (representing the set of feasible solutions to the particular combinatorial optimization problem, e.g., $\mathcal{P}_{(u_{\text{src}},u_{\text{trg}})}$).

At time step t, the reward for the played super-arm $\boldsymbol{a}_t \in \mathcal{I}$ is typically a function of the feedback of each component base arm $a \in \boldsymbol{a}_t$, with (for stochastic CMAB problems) the expected reward of $\boldsymbol{a}_t$ being a function of the parameters associated with the base arms. We denote the *expected reward function* (given a base arm $\boldsymbol{a}$ and the vector $\boldsymbol{\theta}$ of parameters for all base arms) $f_{\boldsymbol{\theta}}(\boldsymbol{a})$. As an example, let θ_a denote the expected reward (or, more generally, the expected feedback) of each base arm $a \in \mathcal{A}$, in a CMAB problem where the reward of each super-arm $\boldsymbol{a} \in \mathcal{I}$ is the sum of the rewards of all base arms $a \in \boldsymbol{a}$ (e.g., the shortest path problem). Then, the expected reward for each super-arm $\boldsymbol{a} \in \mathcal{I}$ can be defined as $f_{\boldsymbol{\theta}}(\boldsymbol{a}) \triangleq \sum_{a \in \boldsymbol{a}} \theta_a$.

Like for the standard MAB problem, regret is a useful metric for comparing and evaluating CMAB methods. The regret for a CMAB algorithm applied to a specific CMAB problem instance characterized by a parameter vector $\boldsymbol{\theta}^*$ (unknown to the algorithm) is defined as

$$\text{Regret}(\boldsymbol{\theta}^*, T) \triangleq \mathbb{E}\left[\sum_{t \in [T]} \left(f_{\boldsymbol{\theta}^*}(\boldsymbol{a}^*) - f_{\boldsymbol{\theta}^*}(\boldsymbol{a}_t)\right) \,\middle|\, \boldsymbol{\theta}^*\right], \tag{2.8.1}$$

where, analogous to the standard MAB problem, $\boldsymbol{a}^* \triangleq \arg\max_{\boldsymbol{a} \in \mathcal{I}} f_{\boldsymbol{\theta}^*}(\boldsymbol{a})$. Since the set of feasible super-arms $\mathcal{I}$ (for many CMAB problems) is of exponential size with respect to the number of base arms, it is generally not practical for CMAB algorithms to enumerate over $\mathcal{I}$ to select a super-arm $\boldsymbol{a}_t$ to play. Instead, several algorithms (e.g., CUCB by W. Chen et al. 2013) utilize offline optimization *oracles* (i.e., combinatorial optimization algorithms, such as Prim's algorithm for finding minimum spanning trees) together with an estimate $\hat{\boldsymbol{\theta}}$ (of the underlying parameter vector $\boldsymbol{\theta}^*$) acquired during earlier time steps.

Throughout this thesis, we often utilize Bayesian CMAB algorithms (e.g., Thompson sampling) together with assumptions regarding prior distributions over possible environments (i.e., CMAB problem instances). Since the regret defined in Eq. 2.8.1 is defined for a specific problem instance $\boldsymbol{\theta}^*$, it can also be useful to consider a Bayesian notion of regret taking a prior distribution $\boldsymbol{\lambda}_0$ over problem instances into account, such that

$$\text{BayesRegret}(T) \triangleq \mathbb{E}_{\boldsymbol{\theta}^* \sim \boldsymbol{\lambda}_0}\left[\text{Regret}(\boldsymbol{\theta}^*, T)\right], \tag{2.8.2}$$

where $\text{Regret}(\boldsymbol{\theta}^*, T)$ is defined as in Eq. 2.8.1 and the outer expected value is taken over the prior distribution $\boldsymbol{\lambda}_0$.

Chapter 3

Summary of Included Papers

In this chapter, we provide a brief summary of the contents included in each of the appended papers.

3.1 Paper 1

In this thesis, we study two main perspectives on (long-distance) navigation of battery-electric vehicles. To limit the time spent for charging during long trips, one may either choose paths between charging stations minimizing energy consumption (possibly resulting in longer time spent driving) or minimizing travel time (while ensuring that the energy consumption does not exceed the energy available). The focus of Paper 1 is on the former perspective.

Concretely, we consider the task of how a single vehicle or a fleet of vehicles may, by collecting energy consumption data for roads traversed during multiple trips through a road network, learn enough to reduce the expected energy consumption over time. In this setting, we view the energy consumption of road segments (represented by edges in a graph structure) as stochastic with *a priori* unknown distribution parameters. To learn these parameters efficiently, we propose a novel online learning framework utilizing Bayesian combinatorial semi-bandit methods for BEV navigation, where the prior distribution parameters are assigned using a model (i.e., Eq. 2.4.5 in Section 2.4) of road segment-specific vehicle energy consumption.

We adapt *Thompson sampling* and *BayesUCB* (a Bayesian variant of UCB proposed by Kaufmann, Cappe, et al. 2012) for our framework to induce sufficient (according to the prior beliefs assigned to the parameters) exploration of the road network, where the latter method is novel to the online shortest path problem. We consider both single-agent (i.e., a single vehicle collecting data over time) and multi-agent (with several vehicles synchronously sharing data) versions of the problem setting and framework.

To evaluate the methods, we perform a thorough experimental study using multiple real-world city road networks (consisting of actual map data combined with simulated traffic environment data) and synthetic networks with varied density and size. We compare the CMAB methods with more naïve baselines (combinatorial versions of greedy and ϵ_t-greedy strategies) for three different scenarios: **(i)** the edge-specific

(a) Luxembourg (b) Turin

Figure 3.1: Exploration of energy-efficient paths in Paper 1, using Thompson sampling (explored paths in red, with higher opacity indicating more visits).

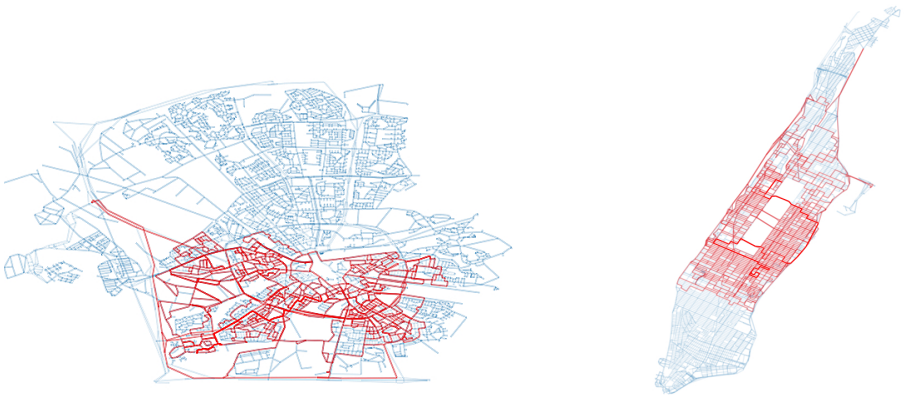
energy consumption distributions and prior distributions are misspecified by the agent (i.e., the energy consumption is generated from a different, albeit realistic, distribution), **(ii)** the energy consumption and prior distributions are correctly specified by the agent (i.e., the assumptions are correct), and **(iii)** there exists correlation between the energy consumption of different edges. In these experiments, Thompson sampling consistently outperforms the other exploration strategies. Examples of how Thompson sampling explores the cities of Luxembourg and Turin are shown in Figure 3.1.

Beyond the aforementioned empirical study, we also perform a theoretical analysis and derive a general upper bound of $\tilde{\mathcal{O}}\left(|\mathcal{A}|K + |\mathcal{A}|\sqrt{T}\right)$[1] on the Bayesian regret of combinatorial Thompson sampling under semi-bandit feedback received in batches of size K (i.e., the agent receives feedback only every K time steps). This result is then extended to regret bounds for our framework (with Thompson sampling as exploration strategy), both in the single-agent (i.e., $K = 1$, non-delayed feedback) and multi-agent (i.e., K-agent) settings.

3.2 Paper 2

Whereas the *offline* combinatorial optimization problem of Paper 1 is the shortest path problem (as defined in Section 2.2), the corresponding offline problem studied in Paper 2 is instead the *minimax path problem* (see Section 2.3). In other words, this work outlines an approach for the online version of a bottleneck identification problem, where the objective is to find a path (through a graph) which minimizes the expected maximum edge weight (i.e., the bottleneck). As an example (relating to the overall theme of this thesis), instead of finding a path which minimizes the total travel time, we may wish to find a path which is expected to avoid *any* traffic congestion (e.g., to reduce driver and passenger irritation). Finding road network

[1]Here, $\tilde{\mathcal{O}}(\cdot)$ hides a polylogarithmic factor w.r.t. T.



(a) Eindhoven (b) Manhattan

Figure 3.2: Exploration of minimax paths in Paper 2, using Thompson sampling (explored paths in red, with higher opacity indicating more visits).

bottlenecks can also be important for local road authorities wishing to know where infrastructure improvements are necessary.

To address this problem, we formulate an online learning framework similar to the one introduced in Paper 1, but with a different (non-linear) expected reward function and offline optimization oracles (i.e., the methods described in Section 2.3 for both undirected and directed graphs). Again, we take a Bayesian combinatorial semi-bandit approach and derive a Bayesian regret bound of $\tilde{\mathcal{O}}\left(|\mathcal{A}|\sqrt{T}\right)$ for Thompson sampling applied to the objective characterized by this expected reward function (the expected minimum base arm reward, or equivalently, the expected maximum when the function is instead interpreted as an expected cost function to be minimized).

The primary issue with the objective described above is that no exact expression is known to exist for the expected reward function (except for a very low number of base arms). To address this, we formulate an alternative *approximate objective* (with the expected reward function defined by the minimum expected feedback, rather than the expected minimum). Then, with CMAB methods (including Thompson sampling and BayesUCB) utilizing the approximate objective, we perform simulation experiments on several undirected and directed graphs to evaluate the performance of the framework. Two of these networks (explored using Thompson sampling) are shown in Figure 3.2.

With the exception of one experiment on a problem instance of small size, where both *exact* and *approximate* methods are compared using regret computed with the exact expected reward function, all (large-scale) experiments are evaluated using an approximate notion of regret. However, we note that both Thompson sampling and BayesUCB still exhibit sub-linear regret (w.r.t. T) on all problem instances considered. Finally, to connect both objectives, we derive an upper bound on the difference in (exact) expected reward between the best paths under the exact and approximate objectives.

3.3 Paper 3

Paper 1 (and Paper 2 to some extent, if the objective is to avoid edges with high energy consumption) may be used to avoid (or delay) the problem of selecting charging stations when the distance between the source and target vertices is relatively short. However, for long-distance trips, charging might be unavoidable. Furthermore, many drivers (and passengers) likely prefer paths which minimize travel time rather than energy consumption, especially considering that paths selected according to the latter objective may avoid high-speed roads.

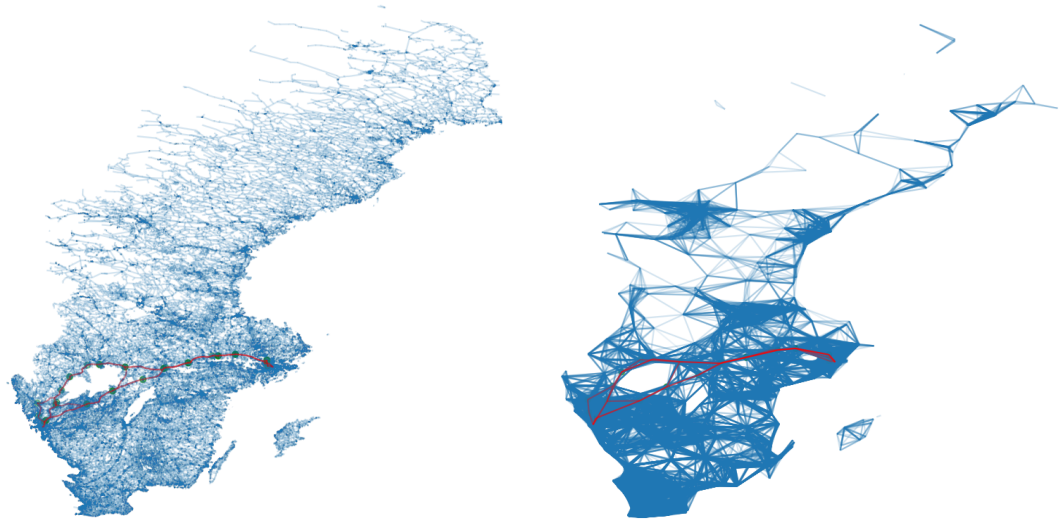
In Paper 3, we study the problem of BEV navigation where at least one charging session is necessary, e.g., due to a limited energy storage capacity of the vehicle battery or a long distance between the source and target vertices. Similar to the previous works, we consider the environment (in this case, the time required for each charging session) to be stochastic and initially unknown. However, in contrast to Paper 1 and Paper 2, wherein only city-sized road networks are used to evaluate the methods, the focus in this work is on methods viable for large-scale networks of (at least) country-size. Furthermore, there are additional complicating factors to consider. Firstly, the battery energy must not be depleted between charging stations. Secondly, the time required for a charging session is influenced by the amount of energy discharged during the preceding trip between charging stations.

To deal with this, while still allowing efficient combinatorial semi-bandit methods to be employed (with performance retained from the aforementioned smaller problem settings), we transform country-wide road network graphs into graphs with pre-computed feasible paths (of least travel time) between charging stations. Then, we model the time of each charging session as a function of the energy to be charged (dependent on the preceding path selected by the algorithm), the waiting time for the charging station and the available charging power.

Through this approach, only parameters associated with vertices need to be explored and estimated by the applied CMAB methods. Since the number of vertices (charging stations) is significantly lower than the number of edges (feasible shortest paths), and the regret incurred by many CMAB algorithms (e.g., Thompson sampling) scales with the number of base arms, this results in a combination of good CMAB performance and computational efficiency.

We demonstrate this by performing simulation experiments on problem instances characterized by the real-world country-wide road networks[2] of Sweden (with road and feasibility graphs shown in Figure 3.3), Norway and Finland, combined with data on the specifications of the actual charging locations in each of the countries. We focus on source and target vertex pairs corresponding to major cities, and evaluate the performance of various CMAB methods applied to the problem instances. Despite the assumption of sub-exponential base arm feedback distributions (relaxed relative to the more common sub-Gaussian assumption for MAB problems), the performance of Thompson sampling and BayesUCB on these larger problem instances is comparable to the performance exhibited by the methods when applied to the earlier problem

[2]Road networks are based on map data copyrighted OpenStreetMap contributors and available from `https://openstreetmap.org`.



(a) Road graph: Sweden (b) Feasibility graph: Sweden

Figure 3.3: Road and feasibility graphs for Sweden, with Thompson sampling exploration of paths in red and charging stations in green, where the opacity indicates degree of exploration for both.

settings of Paper 1 and Paper 2. Moreover, the measured run-time performance shows the viability of the methods for addressing real-world problems.

3.4 Paper 4

Finally, in Paper 4, we consider a problem which is separate from (but still relevant for) the navigation setting considered in Papers 1-3, i.e., the problem of cost-efficient *online decision-making* (not to to be confused with sequential decision-making, though we utilize a CMAB perspective in this work to address and analyze the problem). It is closely related to the problems of *adaptive information acquisition* and *online feature selection*, and concerns a setting where a decision-maker can perform a number of tests to collect enough information before making a single decision based on the observed test results. Since these tests can be costly, the objective is to reduce the number of tests performed while still ensuring good decisions. Methods for this problem setting may be used for applications including e.g., medical diagnosis, interactive troubleshooting or adaptive surveys for driver charging preferences (loosely relating to the theme of the thesis).

In this work, we cast the online decision-making problem as a sequential decision-making problem under uncertainty, where a decision-maker sequentially performs series of tests followed by decisions. The objective is to improve the way in which tests are selected over time, to eventually reduce the expected cost of necessary testing. We address the task by formulating it as a combinatorial semi-bandit problem, enabling combination of CMAB methods (Thompson sampling and BayesUCB) with efficient adaptive information acquisition strategies like *equivalent class edge cutting* (EC^2, see Golovin et al. 2010) and *information gain* (IG, see Dasgupta 2005). We also

develop variants of the latter methods, adapted to handle stochastic test costs, which we call *weighted EC*^{2} (W-EC^2) and *weighted IG* (W-IG), respectively.

To evaluate the performance of the methods, we perform an experimental study where we apply them to data sets from different domains. We also analyze the approach by deriving a Bayesian regret bound of $\tilde{\mathcal{O}}\left(mn\sqrt{T}\right)$ for a hypothetical perfect CMAB oracle (where both W-EC^2 and W-IG are approximate oracles) used together with Thompson sampling.

Chapter 4

Concluding Remarks and Future Work

In this thesis, we study a variety of different problems and perspectives related to the navigation of electric vehicles in uncertain environments, and propose strategies based on combinatorial semi-bandits for targeted collection of data to improve navigation performance. The problems considered are particularly amenable to Bayesian CMAB methods, since we often have access to domain-specific knowledge which can be used for assigning prior beliefs to probabilistic model parameters.

Whereas navigation systems for conventional vehicles (i.e., with internal combustion engines) only need to consider travel time, charging sessions for BEVs can significantly impact the total duration of a trip. In addition, while expected travel time may be estimated using data from many different sources (cellular devices, connected vehicles, road sensors, etc.), parameters associated with vehicle energy consumption or charging stations might be specific to certain vehicle models, road segments or charging locations, considerably reducing the number of agents available for data collection and increasing the relative benefit of being economical with agent resource utilization.

With the goal of achieving this, we adapt combinatorial versions of Thompson sampling and BayesUCB to various kinds of base arm feedback (e.g., energy consumption, traffic congestion or charging time), different expected reward functions with associated offline optimization oracles (e.g., shortest paths or minimax paths), as well as types of graphs (e.g., small, large, dense or sparse). We also demonstrate the performance and potential of the methods through both empirical and theoretical analyses, where Thompson sampling, in particular, consistently performs well.

With the exception of Paper 3, we almost exclusively consider fixed and independent base arm feedback distributions with sub-Gaussian noise. Assumptions like these are very common in the MAB literature, and allow us to focus on specific aspects in our theoretical analyses (e.g., batched feedback or non-linear expected reward functions) with clean proofs which may be extended by later works for more complex settings (e.g., Sandberg et al. 2023). Another benefit of using (relatively) simple models is that they are computationally efficient, and consequently viable for implementation in real navigation systems.

Nonetheless, the aforementioned assumptions are unlikely to hold in realistic traffic environment settings, where traffic conditions differ significantly between rush hours and other times of the day, as well as between weekdays and weekends. Furthermore, traffic patterns might also vary during the year due to changing weather conditions.

Hence, for future work, it may be worthwhile to consider ways in which the methods can be extended to sufficiently capture any time-dependency associated with the feedback. One potential way forward is to model this aspect as a (combinatorial) *contextual multi-armed bandit problem*, where contextual information (possibly the time itself, or some useful proxy for the time-dependency) is revealed to the agent before it has to select an arm. Then, the objective is to learn the way in which the feedback depends on the context (see, e.g., Sandberg et al. 2023, for contextual CMAB methods applying Gaussian processes to problems similar to the ones studied in Paper 1).

One central issue with using such an approach, for large-scale online shortest path problems in particular, is that the contextual information may be outdated when locations far from the source vertex are reached by the agent (since the arrival time of a location depends on the path traversed to reach it). In fact, it is shown by Dean (2004) that even in a full-knowledge setting, in general, finding the path of least time in a time-dependent graph is an NP-hard problem. This gives an indication of the difficulty of the problem, though a possible approach might be to develop sufficiently good approximation oracles (see, e.g., W. Chen et al. 2013).

Relating to the electric vehicle navigation problem described in Paper 3, where the main problem considered is the selection of charging stations to explore, another interesting aspect to consider is that a human agent may be less likely to accept a specific charging *connector* (e.g., due to personal preferences, distance to amenities, etc.) than the general charging *location* (e.g., a larger parking lot). A possible future research direction connected to this can be to investigate how this affects the performance of the methods and if it is beneficial to explicitly model this aspect.

Bibliography

Agrawal, Rajeev (1995). "Sample mean based index policies by O(log n) regret for the multi-armed bandit problem". In: *Advances in Applied Probability* 27.4, pp. 1054–1078. DOI: 10.2307/1427934 (cit. on p. 16).

Agrawal, Shipra and Navin Goyal (June 25–27, 2012). "Analysis of Thompson Sampling for the Multi-armed Bandit Problem". In: *Proceedings of the 25th Annual Conference on Learning Theory*. Ed. by Shie Mannor, Nathan Srebro, and Robert C. Williamson. Vol. 23. Proceedings of Machine Learning Research. Edinburgh, Scotland: PMLR, pp. 39.1–39.26 (cit. on p. 17).

Åkerblom, Niklas, Yuxin Chen, and Morteza Haghir Chehreghani (July 2020). "An Online Learning Framework for Energy-Efficient Navigation of Electric Vehicles". In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. Ed. by Christian Bessiere. Main track. International Joint Conferences on Artificial Intelligence Organization, pp. 2051–2057. DOI: 10.24963/ijcai.2020/284 (cit. on p. 5).

Åkerblom, Niklas, Yuxin Chen, and Morteza Haghir Chehreghani (2023). "Online learning of energy consumption for navigation of electric vehicles". In: *Artificial Intelligence* 317. DOI: 10.1016/j.artint.2023.103879 (cit. on p. 5).

Åkerblom, Niklas and Morteza Haghir Chehreghani (2023). "A Combinatorial Semi-Bandit Approach to Charging Station Selection for Electric Vehicles". In: *Transactions on Machine Learning Research*. ISSN: 2835-8856 (cit. on p. 5).

Åkerblom, Niklas, Fazeleh Sadat Hoseini, and Morteza Haghir Chehreghani (2023). "Online learning of network bottlenecks via minimax paths". In: *Machine Learning* 112.1, pp. 131–150. DOI: 10.1007/s10994-022-06270-0 (cit. on p. 5).

Artmeier, Andreas, Julian Haselmayr, Martin Leucker, and Martin Sachenbacher (2010). "The Shortest Path Problem Revisited: Optimal Routing for Electric Vehicles". In: *KI 2010: Advances in Artificial Intelligence*. Ed. by Rüdiger Dillmann, Jürgen Beyerer, Uwe D. Hanebeck, and Tanja Schultz. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 309–316. ISBN: 978-3-642-16111-7. DOI: 10.1007/978-3-642-16111-7_35 (cit. on p. 4).

Auer, Peter (Mar. 2002). "Using Confidence Bounds for Exploitation-Exploration Trade-Offs". In: *J. Mach. Learn. Res.* 3, pp. 397–422. ISSN: 1532-4435 (cit. on pp. 16, 18).

Auer, Peter, Nicolo Cesa-Bianchi, and Paul Fischer (2002). "Finite-time analysis of the multiarmed bandit problem". In: *Machine learning* 47, pp. 235–256. DOI: 10.1023/A:1013689704352 (cit. on pp. 16, 17).

Auer, Peter, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire (1995). "Gambling in a rigged casino: The adversarial multi-armed bandit problem". In: *Proceedings of IEEE 36th Annual Foundations of Computer Science*. IEEE, pp. 322–331. DOI: 10.1109/SFCS.1995.492488 (cit. on p. 15).

Baum, Moritz, Jonas Sauer, Dorothea Wagner, and Tobias Zündorf (2017). "Consumption Profiles in Route Planning for Electric Vehicles: Theory and Applications". In: *16th International Symposium on Experimental Algorithms (SEA 2017)*. Ed. by Costas S. Iliopoulos, Solon P. Pissis, Simon J. Puglisi, and Rajeev Raman. Vol. 75. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 19:1–19:18. ISBN: 978-3-95977-036-1. DOI: 10.4230/LIPIcs.SEA.2017.19 (cit. on pp. 4, 10).

Bellman, Richard (1958). "On a routing problem". In: *Quarterly of applied mathematics* 16, pp. 87–90. DOI: 10.1090/qam/102435 (cit. on pp. 4, 10).

Berman, Oded and Gabriel Y Handler (1987). "Optimal minimax path of a single service unit on a network to nonservice destinations". In: *Transportation Science* 21.2, pp. 115–122. DOI: 10.1287/trsc.21.2.115 (cit. on pp. 10, 12).

Cesa-Bianchi, Nicolò and Gábor Lugosi (2012). "Combinatorial bandits". In: *Journal of Computer and System Sciences* 78.5. JCSS Special Issue: Cloud Computing 2011, pp. 1404–1422. ISSN: 0022-0000. DOI: 10.1016/j.jcss.2012.01.001 (cit. on pp. 4, 18).

Chapelle, Olivier and Lihong Li (2011). "An Empirical Evaluation of Thompson Sampling". In: *Proceedings of the 24th International Conference on Neural Information Processing Systems*. NIPS'11. Granada, Spain: Curran Associates Inc., pp. 2249–2257. ISBN: 9781618395993 (cit. on p. 17).

Chen, Bi Yu, William HK Lam, Agachai Sumalee, Qingquan Li, Hu Shao, and Zhixiang Fang (2013). "Finding reliable shortest paths in road networks under uncertainty". In: *Networks and spatial economics* 13.2, pp. 123–148 (cit. on p. 4).

Chen, Wei, Yajun Wang, and Yang Yuan (June 17–19, 2013). "Combinatorial Multi-Armed Bandit: General Framework and Applications". In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. by Sanjoy Dasgupta and David McAllester. Vol. 28. Proceedings of Machine Learning Research. Atlanta, Georgia, USA: PMLR, pp. 151–159 (cit. on pp. 19, 28).

Dani, Varsha, Thomas P Hayes, and Sham M Kakade (2008). "Stochastic linear optimization under bandit feedback". In: *21st Annual Conference on Learning Theory*, pp. 355–366 (cit. on p. 18).

Dasgupta, Sanjoy (2005). "Analysis of a greedy active learning strategy". In: *Advances in neural information processing systems* 17, pp. 337–344 (cit. on p. 25).

Dean, Brian C (2004). "Shortest paths in FIFO time-dependent networks: Theory and algorithms". In: *Rapport technique, Massachusetts Institute of Technology* 13 (cit. on p. 28).

Dechter, Rina and Judea Pearl (July 1985). "Generalized Best-First Search Strategies and the Optimality of A*". In: *J. ACM* 32.3, pp. 505–536. ISSN: 0004-5411. DOI: 10.1145/3828.3830 (cit. on p. 10).

Dijkstra, Edsger W (1959). "A note on two problems in connexion with graphs". In: *Numerische mathematik* 1, pp. 269–271. DOI: 10.1007/BF01386390 (cit. on pp. 4, 9).

European Commission (2019). *Communication from the Commission to the European Parliament, the European Council, the Council, the European Economic and Social Committee and the Committee of the Regions - The European Green Deal*. COM(2019) 640 final. Publications Office of the European Union (cit. on p. 3).

Ford Jr, Lester R (1956). *Network flow theory*. Tech. rep. P-932. Rand Corporation, Santa Monica, CA (cit. on pp. 4, 10).

Fredman, Michael L. and Robert Endre Tarjan (July 1987). "Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms". In: *J. ACM* 34.3, pp. 596–615. ISSN: 0004-5411. DOI: 10.1145/28869.28874 (cit. on p. 12).

Geisberger, Robert, Peter Sanders, Dominik Schultes, and Christian Vetter (2012). "Exact routing in large road networks using contraction hierarchies". In: *Transportation Science* 46.3, pp. 388–404. DOI: 10.1287/trsc.1110.0401 (cit. on p. 10).

Golovin, Daniel, Andreas Krause, and Debajyoti Ray (2010). "Near-optimal bayesian active learning with noisy observations". In: *Advances in Neural Information Processing Systems* 23 (cit. on p. 25).

Graepel, Thore, Joaquin Quiñonero Candela, Thomas Borchert, and Ralf Herbrich (2010). "Web-Scale Bayesian Click-through Rate Prediction for Sponsored Search Advertising in Microsoft's Bing Search Engine". In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. ICML'10. Haifa, Israel: Omnipress, pp. 13–20. ISBN: 9781605589077 (cit. on p. 17).

Guzzella, Lino, Antonio Sciarretta, et al. (2007). *Vehicle propulsion systems*. Springer. DOI: 10.1007/978-3-540-74692-8 (cit. on p. 12).

Hart, Peter E, Nils J Nilsson, and Bertram Raphael (1968). "A formal basis for the heuristic determination of minimum cost paths". In: *IEEE transactions on Systems Science and Cybernetics* 4.2, pp. 100–107. DOI: 10.1109/TSSC.1968.300136 (cit. on pp. 4, 9).

Hu, T.C. (1961). "The Maximum Capacity Route Problem". In: *Operations Research* 9, pp. 898–900. DOI: 10.1287/opre.9.6.898 (cit. on p. 12).

Johnson, Donald B. (Jan. 1977). "Efficient Algorithms for Shortest Paths in Sparse Networks". In: *J. ACM* 24.1, pp. 1–13. ISSN: 0004-5411. DOI: 10.1145/321992.321993 (cit. on p. 12).

Kaufmann, Emilie, Olivier Cappe, and Aurelien Garivier (Apr. 21–23, 2012). "On Bayesian Upper Confidence Bounds for Bandit Problems". In: *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Neil D. Lawrence and Mark Girolami. Vol. 22. Proceedings of Machine Learning Research. La Palma, Canary Islands: PMLR, pp. 592–600 (cit. on p. 21).

Kaufmann, Emilie, Nathaniel Korda, and Rémi Munos (2012). "Thompson Sampling: An Asymptotically Optimal Finite-Time Analysis". In: *Algorithmic Learning Theory*. Ed. by Nader H. Bshouty, Gilles Stoltz, Nicolas Vayatis, and Thomas

Zeugmann. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 199–213. ISBN: 978-3-642-34106-9. DOI: 10.1007/978-3-642-34106-9_18 (cit. on p. 17).

Kruskal, Joseph B (1956). "On the shortest spanning subtree of a graph and the traveling salesman problem". In: *Proceedings of the American Mathematical society* 7.1, pp. 48–50 (cit. on p. 12).

Lai, Tze Leung (1987). "Adaptive Treatment Allocation and the Multi-Armed Bandit Problem". In: *The Annals of Statistics* 15.3, pp. 1091–1114 (cit. on p. 16).

OpenStreetMap contributors (2017). *Planet dump retrieved from https://planet.osm.org*. https://www.openstreetmap.org. Accessed: 2021-09-08 (cit. on p. 24).

Prim, R. C. (1957). "Shortest connection networks and some generalizations". In: *The Bell System Technical Journal* 36.6, pp. 1389–1401. DOI: 10.1002/j.1538-7305.1957.tb01515.x (cit. on p. 12).

Rahbar, Arman, Niklas Åkerblom, and Morteza Haghir Chehreghani (2023). "Cost-Efficient Online Decision Making: A Combinatorial Multi-Armed Bandit Approach". In: *arXiv preprint arXiv:2308.10699* (cit. on p. 5).

Rauh, Nadine, Thomas Franke, and Josef F Krems (2015). "Understanding the impact of electric vehicle driving experience on range anxiety". In: *Human factors* 57.1, pp. 177–187 (cit. on p. 3).

Russo, Daniel and Benjamin Van Roy (2014). "Learning to optimize via posterior sampling". In: *Mathematics of Operations Research* 39.4, pp. 1221–1243 (cit. on p. 17).

Sachenbacher, Martin, Martin Leucker, Andreas Artmeier, and Julian Haselmayr (2011). "Efficient Energy-Optimal Routing for Electric Vehicles". In: *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*. AAAI'11. San Francisco, California: AAAI Press, pp. 1402–1407 (cit. on p. 4).

Sandberg, Jack, Niklas Åkerblom, and Morteza Haghir Chehreghani (2023). "Combinatorial Gaussian Process Bandits in Bayesian Settings: Theory and Application for Energy-Efficient Navigation". In: *arXiv preprint arXiv:2312.12676* (cit. on pp. 27, 28).

Shacham, N. (1992). "Multicast routing of hierarchical data". In: *[Conference Record] SUPERCOMM/ICC '92 Discovering a New World of Communications*, 1217–1221 vol.3. DOI: 10.1109/ICC.1992.268047 (cit. on p. 10).

Shimbel, Alfonso (1954). "Structure in communication nets". In: *Proceedings of the symposium on information networks*. Polytechnic Institute of Brooklyn, pp. 199–203 (cit. on pp. 4, 10).

Sutton, Richard S and Andrew G Barto (2018). *Reinforcement learning: An introduction*. MIT press (cit. on pp. 14, 16).

Thompson, W.R. (1933). "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples". In: *Biometrika* 25.3–4, pp. 285–294. DOI: 10.2307/2332286 (cit. on pp. 15, 17).