Obtaining, Improving and Evaluating Spatio-Temporal Information for Efficient Solutions of Mobility Problems

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

vorgelegt von

Lukas Rottkamp
aus Herford

München, den 15.01.2025

Erstgutachter/in:	Prof. Dr. Matthias Schubert
Zweitgutachter/in:	Prof. Dr. Kristian Torp
Drittgutachter/in:	Prof. Dr. Dimitris Sacharidis
Tag der Einreichung:	15.01.2025
Tag der Disputation:	11.07.2025

Eidesstattliche Versicherung

Hiermit erkläre ich, Lukas Rottkamp, an Eides statt, dass die vorliegende Dissertation ohne unerlaubte Hilfe gemäß Promotionsordnung vom 12. Juli 2011 mit 1. Änderungssatzung vom 6. Juni 2012 und 2. Änderungssatzung vom 29. September 2016, § 8 Abs. 2 Pkt. 5, angefertigt worden ist.

München, 15. Januar 2025

Lukas Rottkamp

Contents

Zusammenfassung **vii**

Abstract **ix**

1 Introduction **1**

2 Research Area **5**

2.1 Mobility Problems 6

2.1.1 Parking Search Problem 8

2.1.2 Ambulance Location Problem 9

2.2 Solving Mobility Problems 12

2.2.1 Parking Search Problem 13

2.2.2 Ambulance Location Problem 15

2.3 Obtaining real-time data 17

2.3.1 Measuring attributes of stationary entities 17

2.3.2 Measuring attributes of mobile entities 19

2.4 Improving real-time data 20

2.4.1 Filling spatial gaps 20

2.4.2 Nowcast and Forecast given recent measurements 21

2.4.3 Nowcast and Forecast without recent measurements 22

2.5 Evaluating real-time data 23

2.5.1 Evaluating obtainment methods 23

2.5.2 Evaluating improvement methods 24

2.5.3 Evaluating application-specific data usage methods 24

3 Overview of Contributions **27**

3.1 Solving mobility problems with real-time data 29

3.1.1 Sharing data between competing parking search agents 29

3.1.2 Redeployment of electric ambulances 29

3.2 Obtaining real-time data 30

3.2.1 Placing a limited number of sensors 30

3.3 Improving real-time data 30

3.3.1 Exploiting daily cycles for short term predictions 30

3.4 Evaluating real-time data 31
3.4.1 Effects of data quality 31

4 Efficient Parking Search using Shared Fleet Data 33

5 DEAR: Dynamic Electric Ambulance Redeployment 41

6 Efficient On-Street Parking Sensor Placement 53

7 A Time-Inhomogeneous Markov Model for Resource Availability under Sparse Observations 63

8 Quantifying the Potential of Data-Driven Mobility Support Systems 69

9 Concluding Remarks 81
9.1 Summary 81
9.2 Outlook 82

Bibliography 83

Acknowledgements 93

Zusammenfassung

Die Optimierung des Straßenverkehrs senkt Kosten und schont die Umwelt. Um effiziente Routen zu finden, sind Mobilitätsdienste wie automatisierte Routenplaner auf Basis digitaler Straßenkarten mittlerweile allgegenwärtig. Zusätzliche Echtzeitinformationen über den sich ständig ändernden Zustand der Umgebung bringen weitere Verbesserungen. Diese räumlich-zeitlichen Informationen sind aufgrund der komplexen Umgebung, insbesondere der Vielzahl unabhängiger Akteure, nicht deterministisch und müssen daher durch Sensoren erfasst werden. Die erfassten Daten sind jedoch oft nicht perfekt. Beispielsweise können durch unvollständige Sensorabdeckung Beobachtungslücken entstehen. Eine reduzierte Anzahl von Sensoren kann sogar gewollt sein, denn es muss ein Kompromiss zwischen der Qualität von Mobilitätsdiensten und damit verbundener Kosten gefunden werden. Dieser Zielkonflikt besteht auch im Hinblick auf andere Eigenschaften von Mobilitätsdiensten, etwa der Anzahl dafür benötigter Fahrzeuge. Die Lösung dieser Zielkonflikte ist ein wichtiger Teil der Optimierung.

Um den Straßenverkehr zu optimieren, untersucht diese Dissertation, inwiefern Mobilitätsprobleme effizient unter Verwendung von Echtzeitdaten gelöst werden können. Da Installation und Betrieb von Sensoren mit Kosten verbunden sind, werden Methoden zur Auswahl von Standorten für stationäre Sensoren behandelt. Mobile Sensoren können eine Alternative zu stationären Sensoren sein. Da deren Abdeckungsbereich dynamisch ist, werden Methoden zur Abschätzung unbeobachteter Zeiten und Orte vorgestellt. Weiterhin sind die Auswirkungen verschiedener Datenqualitäten auf die Lösungen von Mobilitätsproblemen nicht offensichtlich. Daher wird betrachtet, wie die Güte verschiedener Methoden und Datenqualitäten bewertet und verglichen werden kann.

Die genannten Beiträge zu den Themenfeldern Beschaffung, Verbesserung und Evaluation von Daten werden mit realen Mobilitätsproblemen illustriert: Die Suche eines freien Straßenparkplatzes kann mittels Methoden, die Echtzeitdaten verwenden, effizient gelöst werden. Weiterhin werden Planungsmethoden für kooperative Multiagentensysteme, die Flottendaten gemeinsam nutzen, vorgestellt. Eine weitere Anwendung ist die Positionierung von Rettungswagen: Betreiber von Notfalldiensten möchten verletzte Personen schnellstmöglich erreichen. Eine Anzahl von Rettungswagen muss daher optimal im Einsatzgebiet verteilt werden. Bisherige Methoden dafür wurden auf Fahrzeuge mit Verbrennungsmotoren ausgelegt. Elektrische Rettungswagen benötigen allerdings signifikante Ladezeiten. Daher wird in dieser Arbeit eine neue Problemdefinition samt Lösungsmethode für die Positionierung elektrischer Rettungswagen vorgestellt.

Abstract

Optimization of street traffic lowers costs and reduces environmental harm. In order to obtain efficient routes, mobility services, such as automated route planners using digital street maps, are now ubiquitous. Adding real-time information regarding the constantly changing state of the environment brings further improvements. Due to the environment's complexity, especially the large number of independent actors, this spatio-temporal data is non-deterministic and thus must be obtained through sensors. However, obtained information is often not perfect. For example, observation gaps may result from incomplete sensor coverage. A reduced sensor set may even be intended, as a trade-off between mobility service quality and associated cost must be made. This trade-off also extends to other characteristics of mobility services, e.g., the amount of vehicles necessary for providing the service. Solving such trade-offs is an important part of optimizing street traffic.

In order to optimize traffic, this thesis investigates how mobility problems can be efficiently solved using real-time information. As the installation and maintenance of sensors to obtain such information comes with a cost, methods for selecting locations of stationary sensors are compared. A method with higher cost efficiency than previously existing solutions is introduced. Mobile sensors can be an alternative to stationary sensors. As their coverage area is dynamic, imputation methods to obtain probabilities for unseen times and locations are presented. Further, the effects of different data qualities on solutions to mobility problems are not obvious. Thus, methods for evaluating performance of different methods and data qualities are considered.

The stated contributions to the topics of obtaining, improving and evaluating data are illustrated using real-world mobility problems: The on-street parking search problem can be solved by methods that make use of real-time data. Further, cooperative multi-agent planning methods using shared fleet data are presented in this thesis. Another application is the ambulance redeployment problem: Emergency medical service providers aim to reach injured persons as fast as possible. Therefore, a number of ambulances must be optimally distributed over the operational area. Previous distribution methods were designed for combustion engine vehicles. Electric ambulances, however, need significant charging times. Thus, this thesis presents a novel problem formulation suitable for electric ambulances, as well as a method for solving this new problem.

Chapter 1

Introduction

Mobility is a central aspect of life. Attending a workplace, visiting a service provider, and joining social gatherings are pillars of modern society and economy. In many places all over the world, individual transportation by private car is a major concern. High load on street networks causes traffic congestion, especially during rush hour. Scarcity of street-network resources such as parking bays or charging stations for electric vehicles further contributes to congestion: Drivers searching for such resources are slowing down traffic and continue to use street space after they have reached their destination. This is especially problematic in case available resources are rare, e.g., most encountered parking bays are occupied [29]. Traffic congestion diverts time from more valuable activities: Not only the quality of life of individuals is affected, but workforce productivity significantly suffers from time loss [4, 42]. Further costs are incurred by increased fuel use, maintenance and fleet sizes. Finally, congestion directly increases pollution due to the widespread use of fossil fuels. CO_2 emissions are the main driver of the current climate change that increases the number of hot temperature extremes, heavy precipitation events and drought events in many regions all over the world [23]. Therefore, optimizing transport is an important endeavor in order to reduce social, economic and environmental cost.

This optimization can be done by two approaches: Optimizing vehicles and optimizing their routes. Electric vehicles can be used to reduce environmental harm. However, optimization of trips is still necessary, as electric vehicles still use energy and are affected by congestion and inefficient routes. More efficient trips not only reduce the negative impact of the optimized trips, but decrease pressure on the street network as a whole. This improves traffic flow for others, which brings additional indirect benefits. One way to optimize trips is to prevent unnecessary detours. For example, the on-street parking search problem is prone to detours: If a vehicle has to be parked on a street near the destination and parking space is rare, the resulting search phase adds further driving until the vehicle is parked [88]. Depending on the search strategy, some of the paths driven during search may be not necessary and could be avoided in order to reach the goal earlier. Optimization of transport also pertains to ambulances of Emergency Medical Service (EMS) providers. One of their core responsibilities is a quick response to health emergencies in order to assess the situation, perform health-stabilizing actions, and transport patients swiftly to hospitals where further measures can be provided [48]. A low response time is critical because survival and recovery rates decline quickly over time in severe health conditions such as cardiac arrest [71, 16]. This time is significantly impacted by the distribution of waiting ambulances and their crews over the covered area [40], which is an interesting spatio-temporal problem. Existing problem formulations and their solutions, however, do not consider the limited battery capacity of current electric vehicles. In summary, both the optimization of trip times and the increased use of electric vehicles are proper ways to reduce the social, economic and environmental cost of transportation.

Optimization of trips especially benefits from the growing availability of data and computational resources that enable tackling such problems using computer science approaches [65]. For example, calculating the fastest (or otherwise optimal) route between a determined origin and destination is now a standard feature of every modern car and mobile phone. The corresponding routing problem has been subject to much research and engineering due to the mentioned benefits of reducing travel times or energy use. In case a vehicle has to be parked at a destination, an extension to this problem surfaces if multiple possible parking opportunities exist. This parking search problem is usually complicated by not knowing the current, or future, occupancy of parking spots. Solving it brings further reduction of travel times. Additionally to the street graph, parking bays are increasingly included in digital maps. Real-time data, such as parking occupancy data, can be obtained by sensors. Sensors can be permanently installed at static locations, e.g., in-ground sensors at parking bays. Data can also be recorded by mobile sensors, i.e., vehicles sensing the availability state while passing by the location. Both variants require significant expenses for implementing and maintaining sensor networks. In the case of electric ambulances, their current positions and energy levels can be determined by the vehicles and then submitted to a central planning instance in real time. Together with expected incident occurrence by area, this is the foundation for real-time ambulance redeployment decisions that optimize ambulances' base stations in order to reach incidents as fast as possible.

Additionally to real-time data, estimates of future states bring further benefits. In the parking use-case, the activity between the current time and future developments until the arrival at parking bays is unknown because competing agents normally do not share information. Thus, their future arrival or departure may change the state of resources. This creates uncertainty about the future state of parking bays even if the real-time data is always fully known. As the agent's trip towards the respective destinations takes time, this can negatively influence the outcome of a parking intention. A suitable solver should therefore not only consider momentary snapshots but also information about the dynamics of parking availability, e.g., estimated inflow and outflow. This dynamic may follow complex patterns. For example, the parking situation in a residential area may be very stable at nighttime, and may change frequently in a busy shopping street. In the ambulance use-case, future energy usage must be considered when making redeployment decisions. For example, an ambulance may be low on energy and therefore required to be stationed at a base station with a charger for a while. Its energy consumption can be estimated based on past behavior, as can the time needed for charging. Finally, time and location of future incidents are unknown. This demand needs to be predicted using appropriate methods, e.g., based on historic data combined with inputs such as population density. No matter the use-case, the real-time state of the environment is often partially unknown: Gaps in the sensor network may create permanent blind spots. Temporary blind spots are caused by mobile sensors being out of an area. In these cases, a now-cast is necessary, i.e., the prediction of the current state given data from other locations at the same time.

This thesis presents contributions to the state of the art of using real-time data to better solve mobility problems. Contributions are clustered into four topics: 1) Problem formulation and methods to find solutions to mobility problems. 2) Obtaining data, e.g., through sensors. 3) Post-processing of data, e.g., if sensor gaps are present. 4) Evaluating the previous steps with regard to the effects of different data qualities in order to arrive at efficient mobility systems. The thesis takes two common mobility problems as examples: The parking search problem and the ambulance redeployment problem. Both benefit from additional data. However, this data needs to be obtained first, e.g., by a set of sensors. As the raw data may not cover all locations and/or times, it may be beneficial to improve it, e.g., through spatio-temporal interpolation. When using data in a real application, it is prudent to compare the expected use of certain data qualities, as improving data quality is usually connected with a cost that may or may not be justified. It should be noted that these contributions do not only concern the stated problems, but can be applied to benefit other mobility problems.

The remainder of this thesis is structured as follows: Chapter 2 describes the research area of this thesis including problem definitions and related work. Chapter 3 gives an overview of contributions of this thesis by identifying and answering a set of core research questions. The subsequent chapters (4, 5, 6, 7, and 8) contain peer-reviewed publications that detail the contributions of this thesis. For each publication, the individual contributions of the respective authors are stated at the beginning of its respective chapter. Chapter 9 then concludes the thesis by giving a summary and an outlook.

Chapter 2

Research Area

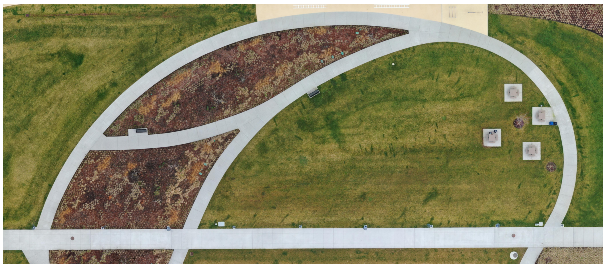
This chapter gives an overview of the research area of this thesis. Key concepts and problems are presented together with approaches published in related work.

First, the term “mobility problem” and other key concepts used throughout this thesis are defined. Two important mobility problems are introduced, namely the parking search problem and the ambulance location problem. The main driver of this thesis is that real-time data enables better solutions to mobility problems. Thus, ways to obtain better solutions by using additional real-time spatio-temporal data regarding the environment are shown. Ways to obtain real-time data are then described. Because obtained data may be incomplete, common approaches for improving real-time data are then presented. Finally, as various data qualities come with different cost, ways to achieve a preferable cost-benefit trade-off between data quality expenses and resulting solution quality are discussed.

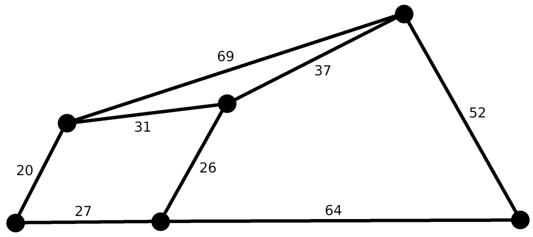
2.1 Mobility Problems

In this thesis, **mobility** means the movement of entities within a spatial environment to more beneficial locations. Entities can be persons, goods or vehicles. The benefit, and therefore value, of a new location may be due to certain actions possible by or with this entity at this location. Beneficial locations are not necessarily static for a given entity. For example, a person may wish to be at home at night but in an office during the day. A **mobility problem** consists of a set of spatial **requirements**, such as a specified entity being at a certain location, a number of locations covered by multiple entities, or a single entity having visited a certain number of locations. It concerns a specified environment, i.e., a street network and all relevant entities. The environment state is affected by a sequence of movement actions. If a sequence of actions transforms the environment in a way that satisfies all requirements, it is called a **solution** to the mobility problem. In its simplest form, a mobility problem asks for a specified entity to be at a certain location, i.e., the route between two given locations, a common problem solved by in-car navigation systems. Its solution is a series of actions, one for each intersection encountered until the destination is reached. More complicated mobility problems require a multitude of requirements to be fulfilled. For example, the Traveling Salesman Problem (TSP) is a well-known problem with multiple locations to be visited by a given entity [51]. The **cost** of a solution is the effort or expense required to execute it. Its metric depends on the underlying use-case. Travel time is often used as the metric because it directly covers economic costs such as wages and vehicle occupation. Further, time spent driving is usually perceived to be less valuable than time spent at the destination. Recent environmental concerns have led to the use of environmental impact metrics and metrics combining environmental impact and travel time [36]. Naturally, in order to use resources efficiently, a good solution should minimize the cost metric.

Environments with many active entities taking actions that change the environment state cause complex dynamics. Actions of one entity can influence the outcomes of an otherwise independent entity. Whether this is considered by a mobility problem depends on the use-case and stance of the involved party. For example, a person looking for a parking opportunity typically aims to find a parking opportunity as fast as possible without considering the goals of other traffic participants. In this case, the solver's optimization horizon ends when a parking opportunity is found. On the other hand, a more global approach optimizes multiple entities over a larger time span: It may, for example, be better (with regard to this global point of view) to send a first user to a parking bay with slightly longer driving time if this allows another user to reach the destination much faster. Optimizing over a larger horizon is especially important for the ambulance location use-case: Letting a patient in serious condition wait for an exceptionally long time poses a severe health risk and thus must be prevented by looking at a larger picture, i.e., moving ambulances to areas with lacking coverage beforehand. Therefore, ambulance placement algorithms optimize a long horizon response time metric [101]. This extended scope typically means that solvers are more complex and must also be evaluated over an extended time span in order to obtain conclusive performance reports.



(a) Paved footpaths seen from above. Cropped aerial photo by Gary Stebbins, CC-BY 4.0



(b) Visualization of graph representation. Edges labeled with actual distance (meters).

$V = \{1, 2, 3, 4, 5, 6\}$
$E = \{(1, 2, 20), (1, 3, 27), (2, 4, 31), (2, 5, 69), (3, 4, 26), (3, 6, 64), (4, 5, 37), (5, 6, 52)\}$

(c) Textual form of graph representation. Nodes in V are indexed left to right. Edges in E are triples, giving indices of two nodes connected by the edge followed by its actual length.

Figure 2.1: Section of the Edmonds Civic Center Playfield footpaths in Washington, USA

A key element of mobility problems is the environment. Mobility is usually constrained to a set of interconnected pathways that enable a safe and speedy locomotion, e.g., a street network. This view is taken throughout this thesis, i.e., all possible travel routes are covered by a graph structure mirroring the real-world pathway network: Pathways are represented by edges E that connect nodes V, which represent real or virtual points in the underlying space, e.g., our three-dimensional Euclidean reality [14]. These nodes include origins and destinations. Each edge connects exactly two nodes, i.e., $E \subseteq N \times N$. As such a graph is defined as tuple $G = (V, E)$. Each vertex $v \in V$ may have additional attributes such as its real world coordinates or type. Each edge $e \in E$ is also usually associated with certain attributes, such as length of the road or the time needed to traverse it. In routing applications, it is common to include the spatial distance and the speed limit of the corresponding road. Node and edge labels are chosen according to the use case. The use case also determines the degree of abstraction and the entities contained or attached to the graph. For example, tunnels and their height may be included in case vehicles may in some configurations not be able to safely pass through them. One example of a mapping from real environment to graph can be seen in Figure 2.1: Figure 2.1a shows an aerial photo of footways. Figure 2.1b shows a visualization of the corresponding graph: Nodes (black circles) indicate the coordinates of intersections. Edges give all possible transfers, labeled with their real-world spatial distances. A textual representation of the graph, similar to a representation inside computer programs, is given in Figure 2.1c. Movement graphs for all parts of the world are prepared and licensed by public and private institutions, as well as community-driven projects, such as the OpenStreetMap initiative.

(a) Parking bays in one-way street (left). (b) Parking bays at intersection.

Figure 2.2: On-street parking bays (rectangles with dark borders) in Melbourne, Australia. Parking bay data by City of Melbourne (`https://data.melbourne.vic.gov.au/explore/dataset/on-street-parking-bays`), CC BY license. Map data by OpenStreetMap contributors, ODbL license. Base map by OpenStreetMap France, CC BY-SA license.

2.1.1 Parking Search Problem

When arriving at a destination by one's own car, it is necessary to park at a suitable parking spot. This need for parking can be fulfilled by three categories of parking types. The most convenient is a reserved, private parking bay. The second is a shared parking facility such as a public parking garage (off-street parking). The third is curbside parking (on-street parking), i.e., parking directly at the side of a street. Especially in residential areas, public on-street parking is often the only choice. A street naturally provides only a limited supply of parking bays. In many cities, this results in a shortness of parking opportunities, creating the need to search for a parking bay within the destination area. This requires additional time for the parking process which contributes to traffic jams, wastes time and resources, and pollutes the environment [88]. In contrast to public on-street parking, private and off-street parking are less complicated: Private parking at an exclusively reserved location does not come with availability difficulties. While off-street parking may be difficult to find in some situations, many solutions for guiding drivers to suitable off-street parking lots exist [5, 20]. This thesis therefore concerns itself with the reduction of parking search time when on-street parking opportunities are rare, i.e., the on-street parking search problem. Figure 2.2 shows two exemplary on-street parking situations: In Figure 2.2a, a number of parking bays is located in a one-way street that are not immediately accessible when arriving from the lower left corner. Figure 2.2b shows an intersection in which more parking bays are located on the left street segment. In both cases, knowing the location of parking bays is helpful when looking for a parking bay in the vicinity.

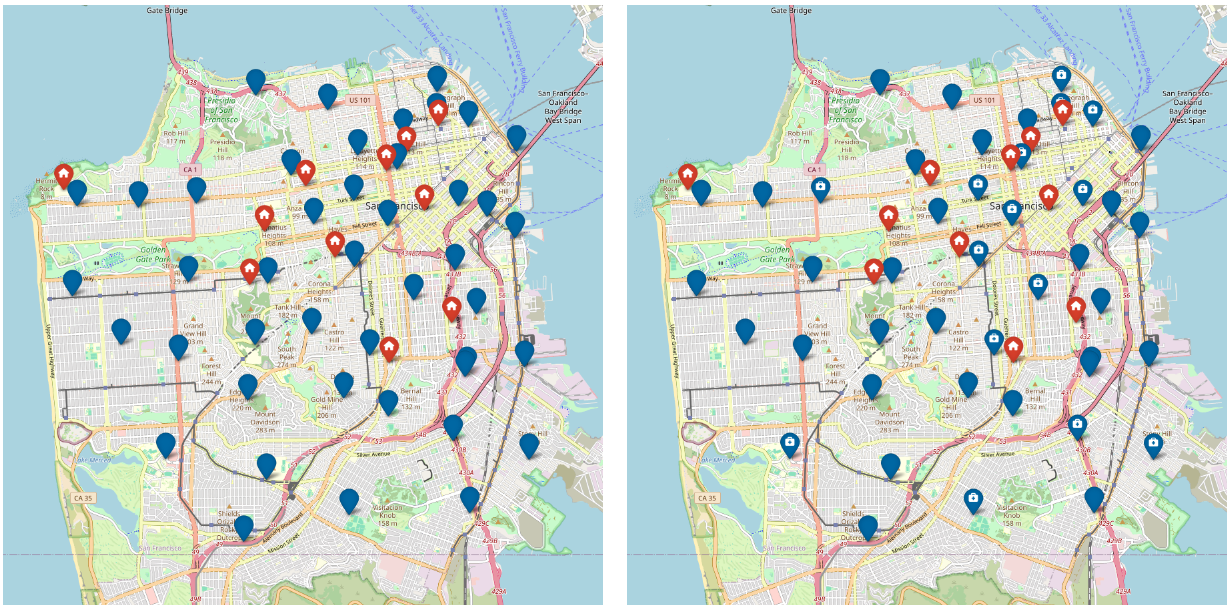
Formally, the parking search problem consists of the following parts: A street graph $G = (V, E)$ with known edge traversal costs. A subset of V are on-street parking bays, each fitting exactly one vehicle and thus having a binary availability state. Third, the origin $o \in E$ and destination $d \in E$ of the agent, i.e., the requirements of the problem. Its solution is a route $r = (e_1, ...e_n), e_i \in E, \forall j \in [1, n-1] : e_j = (u, v) \Rightarrow e_{j+1} = (v, w); u, v, w \in V$ leading from origin to a parking bay available at arrival time. The cost of the route is the sum of costs of traversed edges $c(r) = \sum_{i=1}^{n} c(e_i)$. As a parking bay can be far away from the destination, it is prudent to further add the walking time between parking bay e_n and actual destination to the cost metric: $\hat{c}(r) = c(r) + w(e_n, d)$ with w being the walking cost function. An optimal solution to the parking search problem minimizes the total trip time $\hat{r} = \text{argmin}_r \, \hat{c}(r)$ under the constraint that e_n is available at arriving time.

It should be noted that the parking search problem is trivial if lots of free parking bays are available, i.e., even advanced methods bring only small advantages if parking opportunities are spread out evenly over an area and parking opportunities are not rare [97]. This case is therefore not interesting from a computer scientist's point of view. Further, the disadvantages of parking search, such as additional time requirements, pollution and traffic congestion, are especially large in case free parking opportunities are rare. Therefore, this thesis focuses on the scenario of parking opportunities being rare.

2.1.2 Ambulance Location Problem

Ambulance crews aim to reach emergency sites as fast as safely possible as survival and recovery rates decline quickly over time in severe health conditions such as cardiac arrest [71, 16]. It is therefore crucial to aim for low distances between ambulances on standby and potential accident scenes so that the nearest available ambulance arrives at the scene quickly in case of an emergency [26]. Minimizing distances also reduces the risk for an ambulance to be involved in a traffic accident while driving to the scene and reduces deployment cost [26]. Large cities or metropolitan areas therefore distribute ambulances over the service area in order to minimize emergency response times. Because ambulances need to be cleaned and resupplied between emergency runs, stand-by locations are restricted to a fixed set of base stations, such as hospitals or fire stations [32]. Figure 2.3 illustrates the emergency medical service network of San Francisco, USA: Figure 2.3a shows the actual locations of hospitals and base stations within the city. The distribution of hospitals follows the population density, which is highest in the north-east of the peninsula. Figure 2.3b adds an exemplary assignment of a limited number of ambulances to base stations. This particular assignment also follows population density while at the same time having ambulances in more remote areas to ensure low response times in these areas.

The ambulance distribution problem therefore asks for an assignment of ambulances to base stations that optimizes the emergency response service, limited by a certain number



(a) Locations of hospitals (markers with house icons) and base stations (markers without icon).

(b) Assignment of 15 ambulances (markers with briefcase icons) to base stations.

Figure 2.3: Ambulance scenario in San Francisco, USA. Map data by OpenStreetMap contributors, ODbL license.

of available ambulance vehicles. Ideal solutions optimize a response time metric. In the context of the ambulance distribution problem, response time is defined as the time between an ambulance's departure from its base station and its arrival at the scene [40]. Several response time metrics are possible. A common choice is the average response time (ART) over the number of processed incidents [15]. However, each patient in a life-threatening situation requires a swift response. A single very large delay cannot be equalized by a high number of slightly faster responses. Therefore, ambulance service providers often use the fraction of responses within a certain response time threshold (RTT) as a response time metric [63], e.g., aim to reach 90% of all incidents within 10 minutes.

An early problem formulation for the distribution of ambulances is called the Ambulance Location Problem (ALP) [15, 62, 40]. It asks for a mapping $M : A \rightarrow W$ that assigns each ambulance to a certain base station. This mapping is fixed, i.e., an ambulance always returns to its assigned base station. A static approach has the benefit of low complexity as no further decisions are necessary once the mapping has been decided. A weakness of the ALP formulation's fixed mapping becomes apparent when considering that due to a certain combination of distress calls, two ambulances may end up nearer at the respective other's base station than their own. In this case, it would be preferable for the ambulances to switch base stations in order to reduce travel time. Such situations can be better considered by a dynamic mapping, i.e., an ad-hoc assignment of base stations based on the actual situation. This can be found by solving the Real-Time Ambulance Redeployment Problem or Dynamic Ambulance Redeployment Problem (DAR) [31, 40, 92]. Its underly-

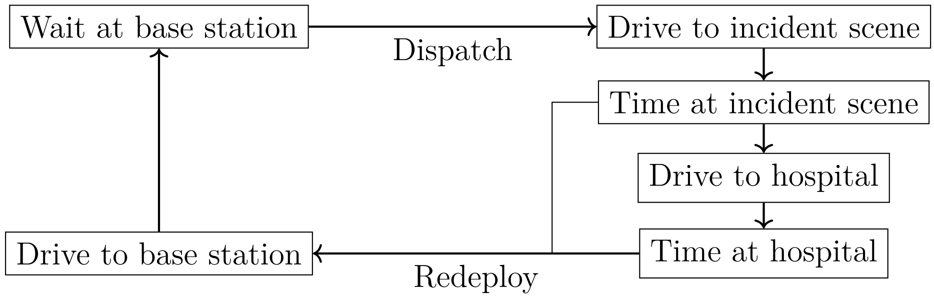


Figure 2.4: Simplified schematic overview of the EMS process with redeployment.

ing ambulance process is shown in Figure 2.4: After an ambulance completes its mission at the incident scene or hospital, it is not forced to return to its previous base station but may return to a different one. This re-assignment of its base station is called redeployment. Resulting solutions can react to the shifting distribution of ambulances and thus generally yield better results than solutions for static formulations due to the volatility of the problem [40]. Formally, the DAR is defined as follows [40]: Its environment state contains a street graph as described above, a set of ambulances A, a set of hospitals $H \subseteq V$, and a set of base stations $W \subseteq V$. A certain ambulance demand d_i is assumed for each node in the street graph. Additionally to the standard driving time, it also includes driving times with siren $\tau_{i,j}$ for each pair of nodes. Each time an incident has to be handled, the nearest available ambulance is dispatched to the incident and follows the process given in Figure 2.4. Determining the next base station to be redeployed to is based on the environment state immediately before the decision is taken.

2.2 Solving Mobility Problems

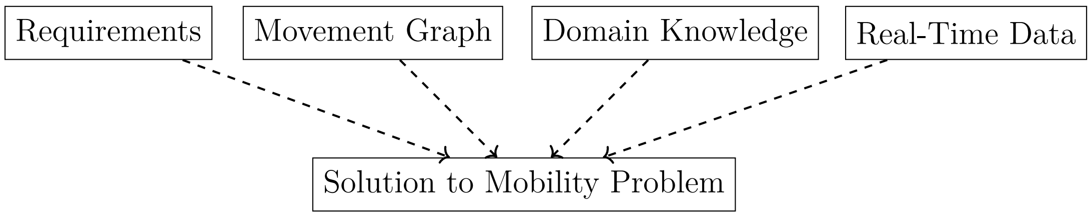


Figure 2.5: Inputs for solving a mobility problem.

Figure 2.5 shows inputs for solving mobility problems: First, the **requirements** to be fulfilled for the mobility problem to be solved and the **movement graph** as both described in Chapter 2.1. Requirements and movement graph are sufficient to obtain solutions. Additional benefit can be achieved by including **domain knowledge**, such as knowing certain rules or characteristics of the underlying processes and involved entities. It includes vehicle characteristics, and characteristics of involved processes such as unloading times, typical traffic patterns, etc. This knowledge can be gained by analyzing the problem and observing environment dynamics, i.e., it represents experience with the respective problem. It may be obtained by mobility researchers or data-driven learning systems. A central point of this thesis is that additional **real-time data** enables better solutions to mobility problems. Real-time data reduces uncertainty about the current environment state and thus enables solutions to be better tailored to current circumstances. One example is a navigation system that is able to produce solutions based on the street graph and traffic rules, but further profits from real-time data, especially in extreme situations like temporary street closures or severe traffic jams. Real-time data is conceptually different from domain knowledge, as it cannot be learned but must be freshly obtained from the environment. It comes in different qualities as exemplified in Figure 2.6 for parking availability data: In the best case, the states of entities are perfectly known, e.g., through direct sensor coverage. In other scenarios, only estimates are available. If future states are concerned, state values are usually probabilistic as mobility problems are typically subject to complex system dynamics, e.g., caused by unknown motivations of individual traffic participants. A probabilistic problem formulation subsumes both types of data and is therefore preferable. In case a method for solving a mobility problem requires non-probabilistic input, it is possible to "convert" probabilistic data to non-probabilistic data by sampling from the respective distribution or following a maximum-likelihood approach.

To make use of the existing work regarding solving problems computationally, it is often beneficial to fit mobility problems into general formal frameworks for which solvers have been established. For example, many mobility problems can be formalized as Markov Decision Processes (MDP). An MDP is a tuple (S, A, C, P) of the following elements [96, 84]: A state space S that contains all possible states of the environment. Each state $s \in S$ contains all relevant information, such as the current location of the entities. Further, a set of actions A that can be taken in order to change the environment state. A state transition

(a) Discrete states (available or occupied).

(b) Probabilistic state knowledge.

Figure 2.6: Availability states of on-street parking bays. Base map license as in Figure 2.2.

probability distribution $P(s'|s,a)$ that gives the probability to transition to state s' after taking action a while in state s. And a cost function $C(s,a)$ giving the cost of taking action a when in state s. The solution to an MDP is a policy $\pi : S \rightarrow A$ that assigns an action to each possible state. Solving the MDP means to find a policy π that maximizes a certain optimization criteria.

Once the problem is formalized, a method to solve it, i.e., to obtain a solution that minimizes the cost metric, has to be found. A multitude of methods for solving mobility problems have been discussed in related work. The naive method is to enumerate all possible solutions and select the best one. However, this complete search is often computationally infeasible due to the large number of solution candidates. Even for conceptually simple problems, such as finding the shortest route between two locations, the complexity of the environment quickly increases the search space thus rendering naive methods such as a brute force search infeasible. Thus, optimized methods have been devised in numerous fields of study. Solvers can be classified into optimal and heuristic solvers: Optimal solvers guarantee an optimal solution. For example, Dijkstra's Algorithm is a fast optimal solver for the Shortest Path Problem [28]. Depending on the problem, optimal solvers may be computationally expensive. For example, the TSP is NP-complete, with a run time rising exponentially with the number of visited cities [37]. Many common mobility problems are instances of the Vehicular Routing Problem (a generalization of the TSP) and also NP-complete [50]. Because this causes the computational demands to be infeasible for problem sizes occurring in the real world, heuristic solvers are widely used. Heuristic solvers do not guarantee optimal solutions, but solutions can be obtained in practical time frames and are often sufficient for practical purposes.

2.2.1 Parking Search Problem

The naive approach to the parking search problem is to drive towards the destination and start looking out for parking bays when near the destination, possibly expanding the search radius until a parking opportunity is found [9]. This strategy is often good enough if many available parking bays are present in the vicinity of the destination. Otherwise, resulting detours will cost time and money. A better strategy is to not target the destination itself but instead known locations of nearby parking opportunities, i.e., to take the number of parking bays per street segment into account [90]. For example, if the destination lies in a residential street that does only provide parking for residents, looking for parking in this street as a non-resident should be avoided. Parking search can be further improved if the typical parking availability is known for the destination area. For example, if parking demand is known to be low, nearer streets with only a few bays may be preferred to slightly farther streets with many bays. On the other hand, streets with low number of bays may be avoided if free parking opportunities are known to be very rare. A simple implementation of parking routing is to look up the nearest group of parking bays in the street graph and set a navigation route to this group instead of the actual destination, speculating that one of those nearest bays might be free at arrival. Alternatively, known parking bays can be displayed to the driver who includes this additional information when deciding on a route to take based on intuition or other preferences [72, 12]. This can be automated by optimizing over all possible routes using a metric that includes both the estimated time spent to find a parking bay, and the (walking) distance to the actual destination [3].

Adding real-time information

Real-time information about availability states of parking areas or individual parking bays enables better solutions to the parking search problem [84]. Figure 2.7 illustrates this: Coming from the south, the western area is initially preferable due to the higher number of parking bays. But considering the current high occupancy there and having high availability estimates for three bays on the right side, it is preferable to instead turn right. Real-time information about availability states can thus be used to obtain more targeted solutions. The corresponding approach has been formalized as a Dynamic Resource Routing Problem (DRRP) [84, 83]: Given a street graph with parking bays as described above, the goal is to minimize the time needed to find an available resource (parking spot) and then walk to the destination. In contrast to formulations in the preceding paragraph, resources are now assumed to have a binary availability state and are not considered if they are known to be not available. The states are probabilistic, i.e., a resource has a probability of being available. Known or assumed availability probability distributions can then be used to find the fastest route. The DRRP has been modeled as a fully-observable Markov Decision Process

Figure 2.7: Coming from the south, it is preferable to turn right given the favorable estimates for three parking bays despite the smaller number of parking bays (note left-hand driving in this picture). Base map license as in Figure 2.2.

(MDP): The state space contains the location of the driver, i.e., the current node in the set of street graph nodes N, and the binary availability state of each parking bay in the set of bays P. Actions move the driver to a neighboring node, i.e., change the state. As availability information is contained in the state, these may also change during the time needed to traverse between the two nodes. This can be done by a continuous time Markov model [84]. Actions are taken based on a policy that can in theory be optimally computed. However, due to the large state space size of $|N|2^{|P|}$, this is computationally infeasible for typical on-street parking areas unless further optimizations are employed [84]. Approximate solvers with drastically lower computational demands have been applied, such as Replanning and Hindsight Planning approaches [82]: Replanning solves the problem for the most likely scenario only, i.e., assumes the most likely availability state for each parking spot. This is fast and has been shown to give good results, but stands in obvious conflict to the problem formulation (and the real world) that does not ignore low-probability bays during parking search. The Hindsight Planner approach includes these by repeatedly sampling possible scenarios according to the probability model and then choosing the best action according to all sampled scenarios [107].

2.2.2 Ambulance Location Problem

The Ambulance Location Problem (ALP) as described in Section 2.1.2 seeks a fixed mapping from ambulances to base stations. Various methods to obtain this mapping have been proposed: The Location Set Covering Model assigns base stations in a way that guarantees the coverage of each possible demand location V. Here, a demand location is said to be covered when it can be reached within a defined driving time [99]. Other definitions of coverage consider the expected demand at demand locations [21, 25, 75]. More sophisticated methods honor the fact that ambulances may be already busy and can not actually cover certain locations during this time, such as the Maximum Expected Covering Location Problem (MEXCLP) approach [24] and its variations [8, 34, 59, 74].

The Dynamic Ambulance Redeployment Problem (DAR), also defined in Section 2.1.2, on the other hand, seeks a dynamic mapping of ambulances to base stations. Methods for solving the DAR often adapt previously existing static redeployment methods and their optimization strategies to the dynamic case. For example, the MEXCLP method described above has been adapted to the dynamic case by sending the ambulance that needs to be redeployed to the base station that maximizes coverage according to the original MEXCLP model at each redeployment step [40]. Another approach uses a reinforcement learning approach that learns redeployment actions with real-world EMS request data: The input for the policy learning algorithm is a vector consisting of a score for each base station. Scores are calculated by a separate neural network based on numerous features of the current environment state, such as the expected number of ambulance requests, the positions of ambulances, travel times of the ambulance to redeploy, and travel times of occupied ambulances [43].

Dynamic redeployment approaches can adapt to changing demand patterns [73]. The demand for ambulances varies significantly because the distribution of the population varies during the day and week. For example, ambulance demand is generally lower at night, especially in industrial areas on weekends, and shifts to residential areas at night [69]. Dynamic redeployment methods can integrate historical patterns into their redeployment decisions.

Adding real-time information

Real-time data can lead to better solutions in various ways. While demand patterns can be inferred from historic data, demand also depends on effects such as the current weather [104]. Real-time weather monitoring therefore can improve redeployment decisions. Other real-time data concerns the situation at hospitals: If hospitals are crowded, the patient and crew may need to wait for several minutes until the patient can be taken over by hospital staff [30]. Incorporating real-time data about waiting times can lower the time until patients can be taken care of, as well as lower the necessary waiting time at hospitals [53].

The introduction of electric ambulances introduces new challenges which can be addressed by real-time data. An electric ambulance running out of energy mid-trip causes a temporary loss of this ambulance until another vehicle arrives to recharge it. Additionally to the extra effort, such an event poses a severe health risk if it happens en-route to an incident or hospital. Therefore, it is crucial to consider the real-time battery state when making dispatch as well as redeployment decisions. Electric ambulances further benefit from the intelligent management of charging stations as the availability of chargers at base stations is restricted due to grid limitations or budget constraints. Sending an ambulance with low energy to a base station with no available chargers should be avoided in order to maximize the number of available ambulances. In conclusion, electric ambulance deployments benefit from real-time data due to their limited energy capacity and non-negligible charging duration requirements. It is therefore important to obtain real-time data regarding charging infrastructure and the battery levels of ambulances. Additional data about active charging processes is beneficial because even though a charger may be occupied by a charging ambulance, this ambulance may clear the charger before another ambulance arrives. On the other hand, if the charging ambulance is still low on battery and another one is already waiting, a detour to a farther away base station with better charging opportunities may be preferable. Related work does not explicitly cover these special considerations for electric ambulances. This thesis fills this gap as described in Chapter 3.

2.3 Obtaining real-time data

As introduced in Section 2.1, entities present in the environment have certain attributes. Attributes can be classified by their volatility: Some attributes change never or very infrequently, such as the presence of a traffic light or a parking bay at a certain street position. They can be obtained by surveying the street network once in a while, or by manual updates initiated by infrastructure providers. On the other hand, high-variability attributes, such as parking bay availability, change frequently. Real-time coverage is therefore required for such attributes. This can be provided by sensor networks. Because real-time data enables better solutions to mobility tasks, the focus of this section lies on high-variability attributes. Attributes can further be classified as categorical or numerical. Categorical attributes take exactly one specific value of a discrete set of values. For example, parking bays are either occupied or available at a certain time. Numerical attributes lie on a scale that can be used for mathematical operations, such as a battery level.

The street environment is a public space that can be observed by an interested party. Local legislation may restrict data collection to protect the privacy of individuals, but it is often possible to record data while maintaining the privacy of individuals by using adequate methods [110]. Data collection can be subdivided into two categories depending on the collection methods: First, data concerning stationary entities and secondly, data concerning mobile entities. Both are described in the remainder of this section.

After data has been recorded locally, it can be sent to a central server for processing [55], or be transmitted between vehicles through a vehicular ad-hoc network (VANET) [19, 27, 52]. A VANET does not require central server infrastructure and may thus be cheaper to maintain. However, the type of information transmission is not further considered in this thesis as it does not affect mobility problems or their solvers.

2.3.1 Measuring attributes of stationary entities

A straight-forward way to obtain data about entities at fixed locations is to manually record them by persons on site. Manual collection is quite flexible and can be a good option to obtain data for static attributes or for short studies concerning dynamic attributes. However, a long term deployment may be prohibitively expensive. Automated approaches using sensors are preferable for long term deployments as their higher initial investment is offset by lower maintenance cost. Such automated sensor setups are described next, starting with stationary sensors followed by mobile sensors.

Stationary sensors

The state of stationary entities can be determined by temporarily or permanently installed stationary sensors which are placed so that their sensor range covers the entity. The type of sensors to deploy depends on the attribute to be measured. For example, the presence of cars at parking bays can be detected using distance sensors (e.g., infrared, radar, ultrasonic sensors), magnetometers and pressures sensors [39] placed in or directly above the street surface parking lot [109, 10, 111]. Another method is to evaluate camera footage overlooking parking bays. This has the advantage that one sensor can cover multiple parking bays. However, processing is more challenging due to natural variability of the scene (visibility, weather, lighting, etc.) [86]. It may also be problematic with regard to privacy considerations.

Depending on the area to be monitored, the amount of sensors needed may be large and thus require considerable installation and maintenance cost. Outfitting only a subset of parking bays with sensors is cheaper, although it obviously comes with the disadvantage of blind spots. In case a sensor can cover multiply entities, one approach is to maximize the *coverage*, i.e., maximize the number of entities covered by sensors [7, 95]. Heuristics such as simulated annealing or genetic algorithms can be used in case the optimal solution is too computationally expensive because of a large set of sensors [70]. Another approach is to place sensors in a data-driven fashion. This problem is known as *sparse sensor placement optimization for reconstruction (SSPOR)* problem [58]. It can be used if data about the spatial phenomenon to be measured is already available. Solutions for it often maximize a non-spatial measure such as entropy or mutual information [49, 67]. Alternatively, the error of a reconstruction method can be minimized [89]. In the case of parking sensors, the data needed for data-driven approaches can be recorded by preliminary studies conducted by persons manually observing the parking bays over a limited time. Another option is the temporary deployment of a full set of sensors. This temporary set can then be replaced with a reduced but permanently installed set after the optimal set has been determined.

For certain types of real-time data, no additional sensors are needed. For example, data regarding ambulance base station chargers can be taken directly from their control units.

Mobile sensors

Data can also be recorded and transmitted by mobile sensors, i.e., sensors attached to objects moving through the environment. These sensors may record a multitude of entities at different times. During the time of movement between entities, they may not be in range of any entity and thus not deliver information. This approach can be further split into two approaches: First, deploying specialized probe vehicles whose primary mission is to record such data. A famous example are Google Street View mapping vehicles that have been driven over large areas and time spans in order to record pictures of streets and other data such as air quality data [2]. However, deploying specialized probe vehicles to obtain data comes with a cost due to maintenance and energy use. A cheaper alternative is to collect data passively during the normal operation of vehicles, e.g., by taxis during their regular service [11]. The disadvantage of this approach is, however, that the areas to be recorded can not be controlled as to not inconvenience the user, causing irregular data gaps which complicate data use. It should be noted that the spatial density of the fleet vehicles' distribution is varying considerably: Highly frequented areas come with a higher number of measurements due to the higher density of vehicles. While this eases the issue as data about highly frequented area tends to be comparatively valuable, gaps still exist and need to be attended.

Modern cars are commonly outfitted with a multitude of sensors to increase safety and comfort. For example, recent cars contain ultrasonic distance sensors for collision prevention while maneuvering near obstacles. Front-facing cameras are installed for automated emergency brake systems. Using these sensors for additional data recording can help to reduce cost of sensing applications. In the case of parking occupancy data, data has been obtained using ultrasonic distance sensors [61] and front cameras [38, 35]. Note that car manufacturers typically don't provide sensor interfaces which complicates data access. Thus, it has been proposed to use sensors of smartphones carried in cars [57] that can be readily accessed by custom apps.

2.3.2 Measuring attributes of mobile entities

Data about mobile entities can also be recorded by stationary and mobile sensors. For the mobility problems covered in this thesis, this is not practical. For example, the destination of a vehicle passing a stationary sensor can generally only be determined with high uncertainty. Similarly, the battery state of a vehicle cannot be determined from outside. Therefore, attributes of mobile entities are usually recorded and transmitted by the entities themselves: Sensors that measure the location, battery state and energy use area already installed in modern vehicles. Measured values can then be transmitted to a central controlling instance via a radio connection such as GSM. In case the vehicle's navigation system is set to a certain destination, such data can be transmitted over the same connection. This also applies to determined routes or certain targeted parking bays. If an external device such as a smartphone is used to provide the mobility service, the relevant information can directly be transmitted from this device. Parking destinations can be chosen by drivers or navigation systems and shared with other traffic participants [19, 27, 52] or a central server [55]. Data regarding ambulance positions and states are usually transmitted by crew through radio on certain events such as arrival at the incident scene, hospital or base station [60]. Computerized systems have been introduced that automatically determine ambulance locations with recording devices such as GPS devices combined with a real-time digital radio connection to the dispatch center [60, 33, 41]. Obtained data can then be considered in ambulance redeployment, e.g., by an emergency dispatch center operator or a computerized system.

2.4 Improving real-time data

Deployed data recording systems often yield data of imperfect quality. This can be due to anomalies such as malfunctions, but can also be systematic due to sensor characteristics or sensor placement. In some cases, the set of available measurements can be sparse. Improving data is therefore often beneficial or even mandatory. Dynamic attributes change depending on the underlying process which makes them probabilistic, i.e., a probability distribution over possible values can be assigned to them. Modeling and estimating these distributions is often a key to improving imperfect data. This section gives an overview of types of data deficits and how they can be mitigated: First, spatial gaps are considered. This is followed by a discussion of temporal missing data that requires nowcast. Data of future points in time can naturally not be observed. Thus, because estimates of future states are beneficial, the topic of predicting future states is covered.

2.4.1 Filling spatial gaps

Recorded spatio-temporal data lacks in the spatial dimension if not every location is covered by sensors, e.g., due to high installation or maintenance cost. Locations of interest within such gaps can be interpolated by spatial interpolation algorithms which take available measurements into account. Interpolation can be formalized as finding a function $A(x) = z$ that assigns a value z to each position x, constrained by the fact that it passes through each point in a given set of N points $R = \{(x_i, z_i)\}$ with $i \in [1, N]$ [64]. In real-world problem settings, x is a two-dimensional location in a city coordinate system. The value z refers to the measured domain, such as availability of a parking bay at location x.

To determine the interpolation function A, several further assumptions have to be taken that depend on the domain [66, 64]. One assumption is that local correlations exist and decrease with growing distance. This is exploited by the Inverse Distance Weighting (IDW) interpolation method that blends measured values weighted by distance [87], shown in Equation 2.1: A value A at location x is the weighted average of all available values A_i. Weights $w_i(x)$ depend only on distance, each being the reciprocal of the distance between the location x of interest and the location x_i of the ith available value raised to the power of α. This parameter α determines the spatial influence: Increasing α shifts weight to nearer sensors, while a small α causes a smoother interpolation as even far away sensors are included with comparatively large weight. IDW is beneficial for interpolating parking availability if only a limited number of sensors is present [13]. This makes sense as the parking pressure of neighboring parking bays should be similar due to the local nature of the parking search process.

$$A(x) = \frac{\sum_{i \in R} w_i(x) A_i}{\sum_{i \in R} w_i(x)}, w_i(x) = \frac{1}{d(x, x_i)^\alpha} \tag{2.1}$$

Note that spatial interpolation often uses the Euclidean distance. In the field of mobility, where travel is conducted on a graph structure, it is instead natural to use graph distance. This distance is the sum of edge lengths of edges needed to travel between two points. In case a certain mode of transport, i.e., transport by car, is known, it further may be prudent to include the driving speed on edges. Driving speed can further be estimated by known speed limits or derived from actual timestamped driving trajectories [44, 108].

2.4.2 Nowcast and Forecast given recent measurements

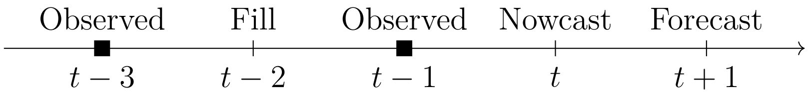


Figure 2.8: Visualization of an entity's measurement time series. Measurements are only available for points marked with black squares. Time t is now.

Similar to the spatial dimension discussed above, the temporal dimension may also suffer from gaps. They can occur on unforeseen, accidental outages. In this case, the latest available data will become increasingly unreliable. Gaps also occur in normal operation due to systematic effects, e.g., the sampling frequency being too low to track the underlying dynamics. One such scenario is a mobile sensor network that is not able to cover all entities within the spatial area continuously, e.g., due to a limited number of sensors or a limited detection ranges: If sensors only visit an entity from time to time, this causes an irregularly spaced, intermittent sampling. A reduced set of sensors may be a conscious choice in order to lower costs and thus is an especially relevant case to consider.

Unknown temporal data falls in three categories given in Figure 2.8: Missing values before the latest observation may be filled to improve solutions. In case the latest measurement is not recent, estimate for the current state (nowcast) is needed. Forecasts may also be used by solvers. For this thesis, gaps after the latest measurement, i.e., nowcast and forecast, are of special importance because mobility problems benefit from real-time data as discussed above in Section 2.2. Note that the age of the most recent measurement varies: Some entities may have been just measured, while others may not have been visited for a rather long time. The older a measurement, the less reliable it is. This *information decay* is an interesting effect that calls for a blend of recent measurements and knowledge about the dynamics that can be determined on historic data: If the last measurement is fresh, it can be taken directly, but the older it is, the more important a model becomes. On the other hand, if the recent measurement is very old (with regard to the attribute's variance), it may have lost its predictive power completely. Continuous-time Markov models have been employed to obtain a nowcast or forecast [18, 45, 82]. They assume exponentially distributed sojourn times in the respective state changes, which allows to determine the current state probabilities based on an existing measurement in the past. While they elegantly solve the problem of determining the current state, their assumption of fixed state change probabilities often clashes with the real world. For example, parking bay availability distributions are usually highly dependent of the time of day, e.g., in residential areas, the time between state changes is much lower at nighttime. Parameter estimation for Markov models is normally done on complete data [76]. It has been done on intermittent data for discrete Markov models using an expectation-maximization algorithm [106] and a Bayesian approach [56]. Various other methods to obtain nowcast or forecast values based on recent measurements exist [98]. They typically are supervised learning methods: Feature vectors include information about the time of day and recent measurements. They are then used to train models such as Regression Trees [109], Wavelet Neural Networks [105] or Support Vector Regression [10] on historic data. These models, however, do not explicitly include the assumption of monotonic information decay discussed above and thus may lead to less accurate results.

2.4.3 Nowcast and Forecast without recent measurements

In contrast to the case discussed above, it may be possible that recent data is never available. In this case, there is no "most recent measurement" to work with and the notion of "information decay" is not useful. One reason may be an abnormal situation, such as a long-term outage. A more common situation is the regular absence of sensors, for example due to budget restrictions. In this case, nowcasts and forecasts must be based on historic data, if available. Expert knowledge may also be incorporated into models. As a wealth of research has been conducted on predicting system states in case no recent data is available, this is not a focus of this thesis: Different types of time series modeling and supervised learning approaches can be used.

Ambulance demand predictions are necessary for determining the number of ambulances, base station locations, hospital capacity and scheduling ambulance crews. Therefore, ambulance demand predictions have been made long before modern deployment methods were introduced. Prediction methods include time series analysis and statistical modeling [1, 46, 68, 91]. Lately, machine learning approaches are being pursued [85, 54, 102]. Some methods are able to predict long term trends, e.g., due to population increase while others focus on the near future by processing weather or event information. Long term forecasts are often used when deciding the construction of new base stations. In contrast, short term predictions are used by DAR solvers. Prediction methods usually do not only predict the absolute demand, but also demand in various areas. For example, higher populated areas typically bring a higher incident density. Accordingly, data about population density, points of interest etc. are valuable input features.

2.5 Evaluating real-time data

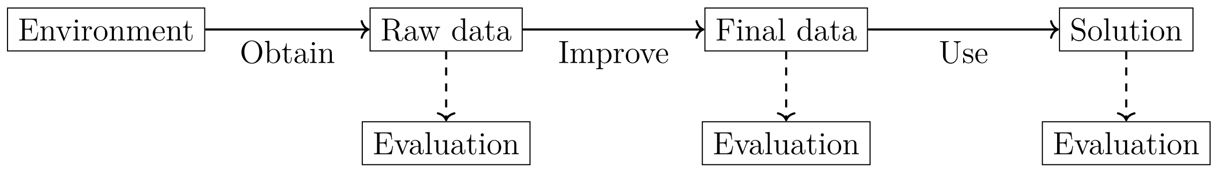


Figure 2.9: Data pipeline from environment observation to using data in mobility service. The three arrows represent classes of data processing methods.

Figure 2.9 summarizes the data flow described in this thesis. Each processing step (*Obtain*, *Improve* and *Use*) can be implemented by a multitude of methods as described in the previous sections. After each step, the result can be evaluated and compared against results of alternative methods. One reason for evaluation is that the final product may be subject to a certain required minimum quality. For example, emergency medical service providers are normally required to satisfy certain minimum response time metrics, often by law [17]. Combinations of methods which cannot reach these minimum targets must therefore be excluded from consideration. While such target figures result from the evaluation of the last step only, it may still make sense to evaluate the intermediate steps, as this evaluation may highlight certain deficiencies. A second reason for evaluation is to ensure a reasonable cost-benefit ratio: Defaulting to the best data quality may be prohibitively expensive, or at least not cost effective. This is due the fact that a higher data quality is usually connected to more expenses, e.g., caused by a larger number of sensors or better sensor capabilities. It is thus advantageous to evaluate different data qualities in order to compare them. The best scenario can then be selected based on efficiency. It should be noted that the effect of different evaluations of the first steps depend on the mobility problem: Some use-cases may depend largely on accurate raw data while others may not be as sensitive. This further depends on evaluated scenarios: For example, it is not at all necessary to obtain real-time data in a parking setting with an abundance of free parking bays. The remainder of this section summarizes evaluation methods of the respective data processing steps.

2.5.1 Evaluating obtainment methods

Performance of methods for obtaining data using sensors depends mainly on the density of a sensor network. As perfect raw data obtained by fully covering the entities in question with sensors is expensive, it may be beneficial to first determine the effects of a reduced sensor set. To obtain test data for evaluation, a temporary field study can be used to obtain complete data. This data can then be statistically evaluated to observe characteristics of the underlying processes. Statistical methods such as time-series cross-correlation analysis can be used to determine the level of sensor redundancy [6, 47]. High redundancy may indicate that the set of sensors can be reduced without harming the quality of obtained data. The perfect data can further be artificially altered by reducing the virtual set of sensors to obtain information about statistical properties of less expensive sensor networks. Another factor is the spatial uniformity of obtained data. For example, the coverage of a mobile sensor deployment depends on the routes of sensors which are determined by the type of vehicles: A fleet of dump trucks can be expected to return a regular covering of a residential area at a regular interval, e.g., visiting each street once a week on a given day. On the other hand, if a fleet of commuters' cars is equipped, the data set relatively over-evaluates areas targeted by commuters at certain locations and times. If the routes of mobile sensors can be controlled, different methods to allocate routes may be simulated to obtain multiple sets of raw data for evaluation. These can then be analyzed as described above.

2.5.2 Evaluating improvement methods

Data improvement methods as described in Section 2.4 transfer an input data set into an output data set. As these methods are usually presented in scientific publications, standard ways of comparing their performance have been established by the scientific community. For interpolation or prediction methods, a test set is normally held back while the remaining data is processed. The method's error can then be determined by comparing generated results against withheld test set values. This enables insights in the suitability of methods. In the case of model-based approaches, evaluations indicate the validity of model assumptions. Methods based on neural networks can be evaluated with regard to parameters such as their network topology and training methods. Note that the choice of improvement methods is limited by the initial data quality. For example, some models are not suitable for sparse data.

2.5.3 Evaluating application-specific data usage methods

The evaluation of actual application performance arguably yields the most valuable results, as it directly measures the end-user value and thus is crucial for strategic decisions. For example, it may turn out that a substantially lower interpolation error may not improve the application performance much. In case a lower error can only be obtained with large investments such as a higher number of installed sensors, it may be wise to accept a deployment with higher errors. Thus, finding a cost-efficient solution requires evaluating the last processing step. Evaluation answers the question of how much data is necessary to reach a certain target. The economic efficiency of smart city solutions can be evaluated before its actual implementation using initial cost benefit analysis (CBA) [100]. Here, costs include investment costs, maintenance costs, etc. Direct benefits include income from rented devices, apps or service subscriptions. Secondary effects such as the reduction of accidents or the increase in life quality are often harder to determine, but still a significant part of the positive effects. To quantify such benefits, it may be suitable to determine time savings.

Simulation methods are especially valuable because they allow the evaluation of different scenarios. For example, a parking service may respond differently when parking availability distributions are modified in the simulation. A simulation is a highly controlled evaluation environment that allows to change individual variables while keeping others unchanged. Computing different scenarios yields insights into complex system behavior. These allow better informed implementation choices. One example is a simulation study regarding the helpfulness of Parking Guidance and Information (PGI) signs that direct drivers to free parking garages, which determined that such signs did barely reduce travel times [103]. To get insight into human parking search process, a simulated human driver searching for on-street parking opportunities has been introduced [9]. Another strategy is the assumption of unrealistically extreme scenarios to evaluate the maximum impact of a smart city method. For example, if data about homes and workplaces of the population is known, the reduction of cars can be determined under the assumption that everyone will use a central car pooling service [22]. While this scenario is unrealistic, it is nevertheless valuable to determine an upper bound on the effectiveness of ride-sharing. For example, if the expected benefits are large, such behavior could be encouraged by the government. In addition to the best-case scenario, it may be interesting to determine mixed scenarios, i.e., 10% of the population using a certain service.

Chapter 3

Overview of Contributions

This thesis brings several contributions to the research area described in Chapter 2. The research area is structured into four fields that enable a categorization of contributions, namely: 1) Solving mobility problems with real-time data; 2) Obtaining real-time data; 3) Improving real-time data; 4) Evaluating real-time data.

Regarding the first field, the thesis contributes to two topics: The parking search problem given multiple agents sharing data (Section 3.1.1; Chapter 4), and the dynamic ambulance redeployment problem for electric vehicle fleets (Section 3.1.2; Chapter 5). In the second field, the thesis considers the problem of distributing a limited number of stationary sensors over spatially distributed entities (Section 3.2.1; Chapter 6). The field of improving data is addressed by contributions regarding the processing of temporally intermittent data, e.g., data obtained from mobile sensors (Section 3.3.1; Chapter 7). Finally, the field of evaluating data is considered by presenting a framework for evaluating and comparing different scenarios regarding their data qualities in order to quantify the potential of mobility services (Section 3.4.1; Chapter 8).

These contributions are contained within five peer-reviewed publications, which are included in the named chapters. To give an overview, each publication is summarized into a research question and its answer, presented in Table 3.1. The remainder of this chapter further expands on these research questions and details the respective context and motivation.

Research question	Summary of contribution
How can the parking search problem be formulated and efficiently be solved for a fleet of selfish agents that share data? [Section 3.1.1]	The parking search problem for multiple selfish agents which share destination and observation data is formulated as a Markov Decision Process. Efficient approximate solvers are presented and evaluated in a realistic setting. [Chapter 4]
Do existing solutions for the ambulance redeployment problem work well for electric ambulances? If not, how can better solutions be obtained? [Section 3.1.2]	The new Dynamic Electric Ambulance Redeployment problem is formulated, considering battery levels and other data relevant for an electric ambulance fleet. A novel solver based on minimizing spatial energy deficits is presented. [Chapter 5]
How do data-driven methods compare to data-agnostic methods when distributing a limited number of stationary sensors over a set of spatially distributed entities? [Section 3.2.1]	Several methods for distributing sensors are presented, some only needing location data, some being data-driven. They are evaluated in a realistic simulation environment for the parking search problem. [Chapter 6]
How can the state of an entity with a temporally non-stationary but cyclic probability distribution be estimated when recent observations are sparse? [Section 3.3.1]	The cyclic behavior of state changes is modeled by a novel Time-Inhomogeneous Markov Model. Training methods for complete data and sparse data are presented and evaluated. [Chapter 7]
How to quantify the potential of data-driven mobility services in relation to data quality? [Section 3.4.1]	A simulation-based framework for determining benefits of mobility services is presented. General scenarios are introduced and exemplified using the parking search mobility service. [Chapter 8]

Table 3.1: Overview of research questions and summarized contributions of this thesis.

3.1 Solving mobility problems with real-time data

3.1.1 Sharing data between competing parking search agents

Agents searching for a free parking opportunity naturally observe the environment for parking spaces and their availability states. If this is automated, for example by sensors attached to a vehicle as described in Section 2.3.1, such parking data could always be recorded, not only when actively searching for a parking bay. As shown in Section 2.2.1, parking services benefit from using parking availability data. The quality of data improves if data from all agents is combined. Therefore, it makes sense for agents to share data in order to minimize parking search times. Parking agents compete with others in the same environment, i.e. agents act selfish in order to optimize their own parking time. Still, it seems promising to share information in order to avoid preventable detours. To sum up, the research question is: **How can the parking search problem be formulated and efficiently be solved for a fleet of selfish agents that share data?**

Chapter 4 therefore investigates how this can be done and how large the improvements are: A novel Markov Decision Process formulation is introduced, which covers multiple agents sharing destination and observation data. Methods to determine an optimal policy are presented for both complete data and sparse data. As described in Section 2.2.1, determining a policy for the single agent case is already computationally infeasible in larger scenarios. The complexity of the new multi agent problem is even larger. Therefore, existing approximate solvers (Replanner and Hindsight Planner) are adapted for this MDP. Evaluations in a simulation using real-world data indicate that the presented method can effectively be used in realistic scenarios.

3.1.2 Redeployment of electric ambulances

The Dynamic Ambulance Redeployment problem given in Section 2.2.2 was formulated before the recent shift towards electric mobility which also includes electric ambulances. Existing solutions assume that the need for refueling vehicles can be neglected during planning as refueling takes little time and gas stations are omnipresent. This is valid for traditional combustion engine ambulances. However, electric ambulances need significant downtime for charging. Therefore, the question arises if the existing DAR problem formulation is adequate for electric ambulances. The existing research gap regarding electric ambulances is addressed by the research question: **Do existing solutions for the ambulance redeployment problem work well for electric ambulances? If not, how can better solutions be obtained?**

This question is investigated in Chapter 5: The existing Dynamic Ambulance Redeployment problem is extended to include not only position and destination of ambulances, but also their battery state. The base stations' charging capabilities are also considered. Combined with information about charging rates and usage estimates, redeployment actions can thus be decided based on properties relevant for electric ambulances in this new Dynamic Electric Ambulance Redeployment (DEAR) problem formulation. The chapter further presents a method for solving this problem: Based on the intuition that the amount of available energy is a proxy for ambulance availability (due to ambulances with lower energy being less available), the introduced MED method aims to minimize spatial energy deficits. To capture the dynamic nature of the problem, future energy deficits are minimized. Evaluation with real-world data shows that electric ambulance redeployment using the DEAR formulation and solver brings lower ambulance response times than using previously existing methods.

3.2 Obtaining real-time data

3.2.1 Placing a limited number of sensors

Section 2.3 states that the best data quality can be achieved if each parking bay is observed by a sensor at any given time. However, because installation and maintenance of sensors comes with a cost, outfitting only a subset of locations with sensors may be preferable in practice. This causes spatial gaps, i.e. locations for which no observations exist. Data for this locations can be interpolated by spatial interpolation methods (see Section 2.4.1). In order to make most of a limited amount of sensors, it is prudent to determine the optimal subset for sensor placement. Research gaps regarding the effecting data quality have been identified, especially regarding the amount of data required for the sensor placement method, as obtaining data for a data-driven placement strategy comes with additional costs. This results in the research question: **How do data-driven methods compare to data-agnostic methods when distributing a limited number of stationary sensors over a set of spatially distributed entities?**

Chapter 6 considers this question by presenting and evaluating a number of selection methods. Methods include data-agnostic methods which only consider the set of possible locations and the street graph. These methods can be applied without much preparation. In contrast, included data-driven methods additionally make use of existing parking availability data. This data may be obtained through preliminary field studies. A simulation framework to evaluate different methods is presented. Exemplary evaluations using real parking availability data are conducted. Scenarios include complete and limited data, respectively. Results show that data-driven methods yield slightly lower interpolation errors than data-agnostic methods. If data is expensive, data-agnostic methods can be reasonable.

3.3 Improving real-time data

3.3.1 Exploiting daily cycles for short term predictions

As described in Section 2.4.2, measurements of a stationary entity are temporally intermittent if obtained by moving mobile sensors. The age of the most recent measurement increases until the next sensor visits, causing the recorded information to become increasingly unreliable (*information decay*). Section 2.2.1 discussed that Markov chains are generally an elegant way to estimate the current state based on the most recent measurement and previously determined transition probabilities. However, methods described in related work assume constant transition probabilities which are not observed in reality: Availability states of real parking bays show a significant cyclic probability distribution change with a 24 hour period, e.g., much less activity at night. A research gap thus concerns the integration of time-dependent model behavior. The resulting research question is: **How can the state of an entity with a temporally non-stationary but cyclic probability distribution be estimated when recent observations are sparse?**

Chapter 7 therefore introduces a novel time-inhomogeneous Markov model which imputes the missing times using non-constant (time-inhomogeneous) transition probabilities. The chapter gives methods for determining transition probabilities on complete data, and, more importantly, on sparse data as obtained in real-world mobile sensor deployments. Evaluations using a simulation environment and real parking data show that the new cyclic model is able to model the *information decay* over time better than previous methods.

3.4 Evaluating real-time data

3.4.1 Effects of data quality

When designing mobility services in order to solve mobility problems such as the parking search problem, one should ask for the potential benefits of possible solutions, e.g., achievable time or energy savings. If such services are using data, the influence of data quality should also be determined, as increasing data quality usually comes with a price (see Section 2.3). Existing literature regarding mobility services focuses on methods for solving mobility problems with or without data, but not the effects of different data qualities. In order to assess the potential benefits of a data-driven mobility service, however, data quality is a crucial element. This yields the research question: **How to quantify the potential of data-driven mobility services in relation to data quality?**

The question is answered in Chapter 8: A simulation-based framework for evaluating mobility services under different data qualities is introduced, focusing on quantifying potential benefits. Different scenarios for data qualities are defined: The *status-quo scenario* approximates the use-case without a mobility service, e.g., a human driver searching for a parking space. Secondly, the *baseline scenario* gives an upper bound, e.g., when no parking is necessary. Thirdly, the *optimal scenario* gives the benefit assuming omniscient knowledge. These are then evaluated against each other and other settings, such as different sensor sets. Exemplary experiments with real and synthetic parking data show that an optimal parking recommendation service leads to travel times not much worse than travel times when taking a taxi, while being a significant improvement over the status quo parking search without any recommendation system.

Chapter 4

Efficient Parking Search using Shared Fleet Data

Publication

Niklas Strauß, Lukas Rottkamp, Sebastian Schmoll, and Matthias Schubert. "Efficient Parking Search using Shared Fleet Data". In: *2021 22nd IEEE International Conference on Mobile Data Management (MDM)*. IEEE, 2021, pp. 115–120. ISBN: 9781665428453. DOI: 10.1109/MDM52706.2021.00026

Extended Version

A long version of the above publication is published on arXiv:

Niklas Strauß, Lukas Rottkamp, Sebastian Schmoll, and Matthias Schubert. "Efficient Parking Search using Shared Fleet Data". In: *arXiv* (2024). DOI: 10.48550/arXiv.2404.10646. URL: https://doi.org/10.48550/arXiv.2404.10646

Contribution

Niklas Strauß is the first author of this publication. The publication covers central aspects of his master's thesis, which was supervised by Lukas Rottkamp and Matthias Schubert. In his thesis, Niklas Strauß presented multi-agent variations of previous work by Lukas Rottkamp and Sebastian Schmoll. Niklas Strauß also modified existing implementations and ran evaluations. Niklas Strauß produced the manuscript. Lukas Rottkamp and Matthias Schubert contributed by discussing ideas and reviewing the manuscript.

Efficient Parking Search using Shared Fleet Data

Niklas Strauß
Institute for Informatics
LMU Munich
Munich, Germany
strauss@dbs.ifi.lmu.de

Lukas Rottkamp
Institute for Informatics
LMU Munich
Munich, Germany
lukas.rottkamp@campus.lmu.de

Sebastian Schmoll
Institute for Informatics
LMU Munich
Munich, Germany
schmoll@dbs.ifi.lmu.de

Matthias Schubert
Institute for Informatics
LMU Munich
Munich, Germany
schubert@dbs.ifi.lmu.de

***Abstract*—Finding an available on-street parking spot is a relevant problem of day-to-day life. In recent years, several cities began providing real-time parking occupancy data. Finding a free parking spot in such a smart environment can be modeled and solved as a Markov decision process (MDP). The solver has to consider uncertainty as available parking spots might not remain available until arrival due to other vehicles claiming spots in the meantime. Knowing the parking intention of every vehicle in the environment would eliminate this uncertainty but is currently not realistic. In contrast, acquiring data from a subset of vehicles appears feasible and could at least reduce uncertainty.**

In this paper, we examine how sharing data within a vehicle fleet might lower parking search times. We use this data to better estimate the availability of parking spots at arrival. Since optimal solutions for large scenarios are computationally infeasible, we base our methods on approximations shown to perform well in single-agent settings. Our evaluation features a simulation of a part of Melbourne and indicates that fleet data can significantly reduce the time spent searching for a free parking bay.

***Index Terms*—Parking, Multi-Agent Routing, Sequential Decision Making Under Uncertainty**

I. Introduction

Searching for free urban resources like on-street parking spots or charging stations can be annoying and time-consuming. According to [1], on average 30% of traffic in cities is caused by parking search. Lowering this has beneficial effects on the environment, drivers and traffic speed.

In recent years, several cities started providing real-time occupation information of on-street parking spots. However, knowing the current state of a parking spot does not eliminate uncertainty about its availability at arrival time as other drivers may claim or leave it in the meantime. Knowing the parking intentions of all drivers would eliminate this uncertainty. Unfortunately, it seems unlikely that all vehicles share their data in the near future. However, a subset of vehicles (a "fleet") may share data, e.g., vehicles with the same route guidance system or belonging to the same company. We propose two approaches using fleet data to reduce uncertainty about future resource states and thus average search time. As optimal solutions are infeasible for larger scenarios, our approaches are based on approximate solvers, replanning and hindsight planning, which have been shown to perform well in single-agent settings [2]. We model fleet parking guidance as a competitive environment with selfish agents, i.e., each tries to minimize its own search time regardless of others.

Our first approach is called *Reservations*: Agents share their targeted parking spot and consider a parking spot to be occupied when they know another agent also targets it and would arrive earlier. This cannot completely eliminate uncertainty regarding agents' parking intention as vehicles not in the fleet also occupy parking spots and thereby cause unexpected rescheduling. Our second approach, *Multi-Agent Dynamic Probability Adaption*, accounts for that by adjusting the availability probability of nearby resources by approximating the agents' behavior when their intended parking spot becomes occupied. We evaluate our approaches with an agent-based simulation, using real and synthetic occupancy data. Results show that our multi-agent improvements reduce the time spent to find a parking spot by up to around 84%.

To summarize, the contributions of this paper are:

- Formalizing resource routing in fleet scenarios.
- Fleet-based resource search based on reservations.
- Fleet-based resource search using adaption heuristics.

This paper is structured as follows: Section II provides an overview of related work. Section III contains the definition of the problem setting and the single-agent solutions our methods are based on. Sections IV and V present our novel solutions for fleet scenarios which are evaluated in an agent-based simulation in section VI. We conclude the paper in section VII.

II. Related Work

Searching for an on-street parking spot is a common problem of daily life. Thus, many approaches have been proposed to either predict the availability of parking spots, simulate the behavior of agents looking for parking spots or find routes that minimize parking search time [3]. Many existing parking guidance approaches do not solve competition between users, rely on hardware to limit access to parking spots or assume that all drivers use the same system. [4] propose parking meters reporting availability so that users can navigate to available spots. In one variation, a resource will only be seen as available by a driver closer to it than competing agents. This is similar to our concept of reservations. However, they do not consider parking activity by vehicles not part of the fleet. [5] introduce a model to predict the utilization rate of a street segment that combines historical and real-time data. At each decision point, the system recommends a street segment matching user preferences. In multi-user scenarios, a multi-user factor decreases the occupancy probabilities when other agents arrive at a street segment earlier. This approach shows some similarity to our multi-agent probability adaption. As they use Integer Programming to minimize the objective function, their approach is NP-hard [6] and thus may be difficult to apply in real-time with large numbers of agents and resources. [7] assign users to parking spots through a central system using a queuing model. A major drawback is that they rely on hardware restricting access to users with a reservation. [8] compute routes to available spots by solving a Time-Varying Traveling Salesmen Problem. Agents observe the state of nearby resources and share it. In contrast to our methods, occupied resources are ignored though they may become available later. [9] use a game-theoretic framework for competitive multi-agent parking search. Their goal is to find a minimal-cost assignment of agents to parking spots. This differs from our setting as they assume that either all vehicles are part of the system or information about other vehicles is only estimated by a prior probability distribution.

[10] define parking search using sensor data as a dynamic resource routing (DRR) problem and solve it as a fully observable Markov decision process (MDP). A policy is computed by bounded real-time dynamic programming using novel bounds and estimation methods. Even though the authors considerably lowered the computational overhead compared to other MDP solvers, the exponential growth of the state space still limits the applicability to larger settings. Thus, [2] propose novel approximate methods based on replanning and hindsight planning. The results indicate that search times come close to the optimal solution of [10] which makes efficient policies in large settings possible. Unlike our approaches, explicit information about other vehicles searching for parking spots is not considered. Our approaches are based on [2] and we include their original methods as a single-agent baseline to determine the benefits of our proposed fleet solutions.

[11] have shown that predicting the availability of a parking spot in the near future is crucial for the performance of dynamic resource routing. A vast number of approaches for such predictions exists [3]. However, predicting the expected occupancy of resources is not equivalent to predicting the probability distribution that a parking spot is vacant at a certain time, which is necessary for many approaches. In this paper, we apply a continuous-time Markov chain (CTMC) [2], [12] to incorporate recent observations (e.g. current resource states) as well as long term observations (e.g. average occupancy time).

III. Methodology

We now formalize the search for an available resource in a fully observable multi-agent setting as an MDP and review the approximate solutions our methods are based on.

A. Problem Setting

The goal of DRR (dynamic resource routing) is to guide an agent to a resource $r_i \in R$, which is available upon arrival, in a directed graph $G = (N, E, C)$, such that the expected total travel time is minimized. The graph G represents a road network: Nodes N correspond to intersections, edges $E = N \times N$ to road segments and $C : E \to R^+$ is a function that defines the cost of traversing an edge. Each resource is located on an edge. Each agent $\alpha_i \in \Lambda$ has a different destination, start intersection and can begin its trip at any time. As this information is not known in advance, Λ is not stationary over time. We define $c_T(r, \alpha_i)$ as the terminal cost, e.g., the time for walking from parking spot r to α_i's destination. The total travel time consists of the driving time and c_T. Whenever an agent is at an intersection, a decision needs to be made whether to take a resource on one of its outgoing edges or to drive to another intersection. We call a setting fully observable if agents always know the current state of all resources.

In a competitive and independent multi-agent setting, a separate MDP is associated with each agent α. An MDP can be defined as a 4-tuple (S, A, C, P), where S is the set of all possible states and A denotes the set of actions. $A_s \subset A$ is the set of available actions while in state $s \in S$. $C : A \to R$ is a function that defines the cost for action $a \in A$. $P : A \times S \to [0, 1]$ denotes the probability of traversing from state $s \in S$ to state $s' \in S$ after choosing action $a \in A_s$.

A policy $\pi(s) = a$ is a mapping from any state $s \in S$ to an action $a \in A_s$. Solving an MDP corresponds to finding a policy that minimizes the expected future costs over an infinite time horizon. We assume that all fleet vehicles are guided by the same system and thus, policies vary only with respect to the agent's destination. Thus, for a given agent, the policies of other agents can be considered static. The expected future costs are commonly denoted as utility U with Bellman equation:

$$U^\pi(s) = C(s_t, \pi(s_t)) + \sum_{s' \in S} P(s'|\pi(s), s) U^\pi(s') \quad (1)$$

The Q-Value of a state-action pair (a, s), where $a \in A_s$ and $s \in S$, describes the expected costs when performing the action a and following the optimal policy π. It is defined as follows:

$$Q_\pi(s, a) = C(s, a) + \sum_{s' \in S} P(s'|a, s) U^\pi(s') \quad (2)$$

An optimal policy takes the action with lowest Q-Value.

In our fleet scenario, the system dynamics are determined by drivers outside the system and fleet vehicles. This is modeled by the transition probabilities of the MDP. Given the policy of an agent, its behavior can be determined and shared with other agents to reduce uncertainty about the future. Therefore, the MDP contains the location and destination of all fleet agents.

We now formulate the fleet DRR problem as an MDP: The **state** is defined as $s = (l, \{L_i\}, \{D_i\}, \{r_j\})$, where $l \in N$ is the node at which the vehicle is currently located, $\{L_i\}$ the set of fleet agents' positions, $\{D_i\}$ the set of their destinations and $r_j \in \{\text{available}, \text{occupied}\}$ the availability of the j^{th} resource. We define two **actions** $a \in A_s$: *Take Road* means to move along an edge of the road network. Exactly one exists for every outgoing edge from the current node. Its cost $c(a)$ is the edge cost. The *Take Resource* action represents driving from a node to a resource on an outgoing edge, parking and walking to the destination. Its cost is the driving time from node to resource r plus the terminal cost $c_T(r)$. An action determines the agent's position in the next state with certainty, while the state's resource availabilities are not deterministic. The **transition probability** $P(s'|a, s, \{\pi_i\})$ of the next state s' depends on action a and the static policies $\{\pi_i\}$ of the other agents. For *Take Road*, it is the product of the transition probabilities of all resources according to the probabilistic model. The *Take Resource* action always leads to the terminal state, as the agent has parked. If other agents execute *Take Resource*, these resources transition to the occupied state.

B. Continuous-Time Markov Models

Most methods in this paper require a stochastic process for describing the future availability of parking spots relative to the last time the parking spot was observed. [12] use continuous-time Markov chains (CTMC) for modeling the time-dependent availability of each resource to incorporate short-term observations (real-time sensor data) and long-term observations (average vacancy/occupancy duration). The availability of each resource is given for the current time $t = 0$. There exists a CTMC for each resource and the CTMCs of all resources are assumed to be mutually independent. Using the Kolmogorov equations, we can compute the transition matrix:

$$P(t) = \begin{pmatrix} T^t_{a,a} = \frac{\mu}{\lambda+\mu} + \frac{\lambda}{\lambda+\mu} e^{-(\lambda+\mu)t} & T^t_{a,o} = 1 - T^t_{a,a} \\ T^t_{o,a} = \frac{\mu}{\lambda+\mu} - \frac{\mu}{\lambda+\mu} e^{-(\lambda+\mu)t} & T^t_{o,o} = 1 - T^t_{o,a} \end{pmatrix} \tag{3}$$

We denote $T^t_{f,f'}$ as the probability for being in state f' after time t has passed while in state f at $t = 0$. Sojourn times are modeled as exponentially distributed random variables with parameter λ when in state *available* and μ when *occupied*. Therefore, λ^{-1} describes the average time a resource stays available and μ^{-1} the average time it stays occupied.

C. Replanning

Solving the DRR problem for a single agent is computationally problematic for large settings as the state space complexity is exponential in the number of resources [2]. Our fleet DRR's state space is additionally exponential in the number of agents.

The replanning solution in [2] avoids working on the state space of the MDP but instead utilizes an extended street network for deterministic planning: The network graph G is extended by a virtual goal node N_{goal} to model the *Take Resource* action. For each resource r_i, a virtual edge from any intersection N_i to N_{goal} is added, where N_i is the starting node of an edge r_i is assigned to. The cost $c_v(r_i) = c(N_i, N_{\text{goal}})$ of this virtual edge is set to the time needed to drive along the road to reach this resource and the terminal cost c_T. If the resource is occupied, the cost is set to the expected time required to circle the block until it is available, denoted as $t_{\text{tr}}(r_i)$. Because planning is not done in the state space of the MDP, we cannot determine the exact transition probabilities between resource states. However, this is unnecessary in many DRR problems as the travel time between neighboring intersections is typically small compared to the time between resource state changes. Thus, for the most likely future, it can be assumed that a resource keeps its state. In a majority of settings, the Replanner detects mistakes very early and thus does not suffer a large penalty if it has to replan.

D. Hindsight Planning

Hindsight planning is typically more effective than replanning in "probabilistically interesting" tasks [2], [13]. It approximates the value of a state by sampling futures, optimizes these with a deterministic solver in hindsight and then combines the solutions. This is often faster than solving the probabilistic problem. [2] proposed a Hindsight Planner for DRR. A determinization or future D is a non-probabilistic configuration of resource states. Let $C(s, a^*, D)$ denote the costs of the optimal solution a^* in state s using the determinization D. The hindsight utility value or expected costs is defined as follows:

$$U_{\text{hs}}(s) = E_D[C(s', a^*, D)] \tag{4}$$

Using this definition, we can now define $Q_{\text{hs}}(s, a)$:

$$\hat{Q}_{\text{hs}}(s, a) = c(a) + E[U_{\text{hs}}(s')] \tag{5}$$

An optimal policy π^* can be approximated by taking the action having the best expected one-step look ahead hindsight value. $\hat{Q}_{\text{hs}}$ approximates the Q-function by reversing the order of minimization and expectation, i.e., instead of taking the policy with minimum expected cost, we use the expected cost of optimal policies w.r.t. D. This can be efficiently approximated by solving determinizations of the probabilistic problem. Note that U_{hs} is a lower bound of the optimal utility values U^* as it is assumed that the outcome of each action is known.

In DRR, sampling a future is essentially equal to sampling from the probability distribution of the resource states. The optimal solution of that determinization is choosing an upon arrival available resource with the least total cost. The hindsight costs $C(s, a, D)$ are computed by calculating the arrival time at each resource and then sampling resource states at that time using the prediction model. If the resource is not available, we need to add the time until the resource becomes available again. As a driver is not allowed to wait at a resource, we determine the number of round trips around the block until it is available again. This is very time-consuming, so we use the mean time until the resource becomes available again instead, denoted as the minimum expected wait time t_{claim}.

IV. Reservations

In our setting, a resource can become occupied either through a fleet vehicle or a vehicle not part of the fleet. The latter case cannot be prevented and is anticipated by a model for resource state changes, in our case, the CTMC described above. We can, however, consider intended occupations by other fleet vehicles to reduce uncertainty, as any other fleet vehicle heading for the same parking spot and arriving earlier dramatically reduces the expected availability for the agent. We call this process reservation as the spot can be considered occupied even if it is currently not. More formally, a reservation is a tuple $(r_i, \alpha_i, t_{\text{arrival}})$, where t_{arrival} denotes the time when α_i arrives at r_i. Note that a resource with reservation still can be occupied by non-fleet vehicles or fleet agents arriving at the resource before the reservation becomes active.

When using the Replanner with reservations, agents treat a "reserved" resource as occupied in case they expect to reach it later than any other agent. If an agent changes its target resource, the existing reservation is deleted and a reservation for the new target is created. In [2], the D*-Lite algorithm is applied to reduce computation time. This is not possible with reservations as the costs of virtual edges depend on resource availability which depends on arrival time and thus on the path previously taken. Finding the shortest path in a time-dependent graph is an NP-hard problem in general [14], [15]. Pre-computing all pairwise travel times between intersections can be done in polynomial time [16], [17], as we do not consider time-dependent costs. Calculating the arrival time at each resource can be achieved in constant time during the query phase. The memory cost can be reduced to $O(N^2)$ [18].

When applying the Hindsight Planner with reservations, a modification is needed as no specific resource is targeted until the end when a *Take Resource* action is chosen: For each agent, the most often visited resource in all determinizations and its expected arrival time at that resource are computed. Based on this estimate, we generate a reservation for each agent. An exception are agents choosing the *Take Resource* action for whom a short-term reservation is created instead. This resolves conflicts in the immediate future when the parking intention of an agent is certain. Reserved resources can be considered occupied in all futures, which restricts the sampling space.

V. Multi-Agent Dynamic Probability Adaption

The Hindsight Planner with reservation overestimates the probability of a resource being available as it does not incorporate the behavior of other agents when they fail to take their targeted resource. Thus, we propose to decrease the availability probabilities of resources near the target accordingly.

To calculate the probability of taking another resource, we propose a self-interacting time-dependent biased random walk on a subset of the graph within an isochrone around the targeted resource to limit the number of streets to consider. In contrast to a pure random walk, the jumping probabilities of a biased random walk are not equal and can depend on several factors, including previously chosen nodes [19]. It can efficiently approximate agents' behavior due to its structural similarity to the DRR MDP. At each intersection in the MDP, an action is selected: Drive to another intersection or park at a nearer resource. The policy specifies a probability distribution over those actions. The biased random walk aims to estimate these probabilities to approximate the policy. Let N_i be the node where the agent is located and N_j is a node reachable from N_i. The set of resources located on the edge (N_i, N_j) is denoted as $R_{\text{reachable}}(N_i, N_j)$. $P_{i,j}(t)$ denotes the probability that at least one resource on the edge (N_i, N_j) is vacant at time t.Note that we assume the availability probabilities of the resources to be mutually independent. The bias $\gamma_{i,j} \in]0, 1]$ of the random walk is the product of visit decay factor $\theta_{i,j}$ and a penalty factor $\delta_{i,j}$ for moving away from the target:

- $\theta_{i,j} = \begin{cases} 0.95 & \text{if edge i,j was already visited} \\ 1 & \text{otherwise} \end{cases}$
- $\delta_{i,j} = \frac{c((N_i,N_j),\text{target}(\alpha))}{\text{IsochroneLimit}}$, with $c((n_i, n_j), \text{target}(\alpha))$ being the time driving from the end of the road to the destination of the agent α.

The jumping probability is defined as follows:

$$J_{i,j}(t) = \gamma_{i.j}(1 - \prod_{r \in R_{i,j}} P_{i,j}(r = \text{occupied}, t)) \qquad (6)$$

With each jump from n_i to n_j, the accumulated time t_{acc} is increased by the costs of driving $c(N_i, N_j)$. The random walk starts at the end of the road in which the target resource is located because agents can only make decisions when they are at intersections. The accumulated time t_{acc} is initialized with the time needed to drive from the target resource to the next intersection. The path probability P_{path} reflects the likelihood of not having found a resource and being at the end of the path. This probability is the product of all jumping probabilities $J_{i,j}$ for the path. It is initialized with the probability P_{initial} that the preferred resource is occupied. We then sample how often we end up at a resource r by only considering resources located on the last street of the random walk. A random walk may end after each jump with probability $1 - P_{\text{path}}$. Let $E_{t_{\text{arrival}}}(r)$ denote the expected arrival time at a resource. It can be calculated using the mean accumulated times t_{acc} of paths that end in a street from which r is reachable. Probability adaptions for a resource are applied by subtracting the parking probability $P_{\text{park}}(r)$ from all predicted availability probabilities after time $E_{t_{\text{arrival}}}(r)$. The parking probability $P_{\text{park}}(r)$ is calculated by equally distributing the expected path probabilities $E[P_{\text{path}}]$ to all resources located on that street. When an agent changes its target resource, all adaptions created by that agent are reversed and the process is repeated with the new target resource. Note that using the expected arrival time at the resource $E_{t_{\text{arrival}}}(r)$ is not entirely accurate. However, it increases the computational performance. As a further enhancement, one could create multiple adaptions for each walk to create more precise predictions about when agents arrive at certain parking spots. The complete algorithm is shown in Algorithm 1.

VI. Evaluation

In this section, we describe the evaluation of our approaches with an agent-based simulation and present its results. The simulation was run on a Linux server VM (Intel® Xeon® Silver 4108 CPU at 1.8 GHz and 59 GB RAM) using one thread per run. All algorithms are implemented in Java. The least-cost paths between all pairs of intersections and the *Minimum Expected Wait Time* have been pre-computed. Our simulation is based on a modified version of the COMSET simulator [20].

A. Experiment Design

1) Road Network and Resources: All experiments were conducted on a sub-graph of the Melbourne road network consisting of 3185 nodes and 6384 edges obtained from OpenStreetMap. Attached to the edges are 4608 on-street parking spots, shown in Figure 1, whose locations are published by

Algorithm 1 Biased random walk to create probability adaptions for resource r_{target} with expected arrival time t_{arrival}.

$paths = \emptyset$
for i = 0 ... $samples$ **do**
  $P_{\text{path}} = P(r_{\text{target}} = \text{available}, t_{\text{arrival}})$
  $currentNode = r_{\text{target}}$.road.to
  $t_{\text{acc}} = t_{\text{arrival}} + t_{\text{partial}}$
  while true **do**
    $\vec{p}$ = calculate biased probability $J_{i,j}(t_{\text{acc}})$ of each outgoing edge
    $\vec{p'} = \frac{\vec{p}}{\sum_{p_i \in \vec{p}} p_i}$
    $\vec{p'}_k = \sum_{i=0}^{k} \vec{p'}_i$
    $\vec{p'}_k = \sum_{i=1}^{k} (\vec{p'}_{i-1}, \vec{p'}_i)$
    $random \in U[0, 1]$ {choose from uniform distribution}
    $nextEdge$ = choose edge k with $random \in p_k$
    P_{path} *= $\vec{p}_k$
    t_{acc} += $c(nextEdge)$
    $currentNode = nextEdge$.road.to
    if $random > p_k$ or no edges available **then**
      $paths$.add(($currentPath$, $pathProbability$, $arrivalTime$))
      break
    else
      $currentPath$.add(($nextEdge$)
CreateAdaptions($paths$)

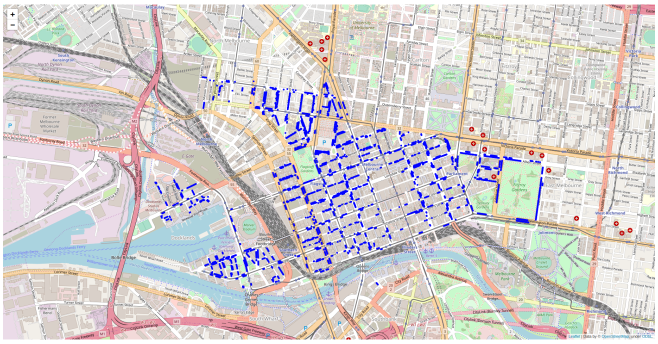


Fig. 1: Locations of parking spots (blue dots) present in the simulation.

the City of Melbourne under a creative commons license [1]. To accommodate for traffic jams and turn delays, the driving speed of an agent is set to 25% of the road segment's speed limit. This factor has been calibrated through taxi trip times in Manhattan recorded in the TLC Yellow dataset. Walking distance (to determine terminal costs c_T) is measured by greater circle distance which means an agent can walk through buildings/parks and cross roads. Walking speed is 1.42$\frac{\text{m}}{\text{s}}$ [21].

2) Resource Volatility: To evaluate various environments, we use a real-world and a synthetic setting: *Real-World* occupations have been recorded by the city of Melbourne through in-ground sensors [2]. A two hour time period on a working day in 2017 is used in our simulation. The average time a parking spot stays available is approx. 75 min; the average time it stays occupied approx. 29 min. *Synthetic Resource* occupations are sampled from a CTMC model with $\mu^{-1} = 2091$ seconds and $\lambda^{-1} = 120$ seconds, i.e., spots are occupied for about 35 min on average and occupied after an expected time of 2 min of being free. This resembles situations where vacant spots are rare (on average 5.4% of resources are free), thus presenting a greater challenge than the *Real-World* setting.

[1]On-street parking spot dataset: https://data.melbourne.vic.gov.au/Transport-Movement/On-street-Parking-spots/crvt-b4kt

[2]Occupation dataset: https://data.melbourne.vic.gov.au/Transport/On-street-Car-Parking-Sensor-Data-2017/u9sa-j86i

	Single Destination		Data-Driven Destinations	
Algorithm	Real	Synthetic	Real	Synthetic
RANDOM	1634	842	794	957
PARKAGENT	961	964	588	555
RPL	667	954	41	104
HS	515	476	122	95
RPL + R	**106**	276	**34**	100
HS + R	109	262	97	**73**
HS + A	111	**237**	37	81

TABLE I: Mean parking time in seconds. "RPL" is Replanning; "HS" Hindsight Planning; "+ R" Reservations; "+ A" Probability Adaptions.

3) Agents' Destinations: In the *Single Destination* setting, we create high competition by simulating 20 agents with same destinations and start times to examine how algorithms handle competition. We also include *Data-Driven Destinations* in which the number of agents and their destinations are estimated from real data: Destination clusters are inferred from Melbourne parking events using DBSCAN [22]. Destinations are randomly determined from two clusters. Start times are randomly selected uniformly over one hour of a working day. 729 agents are heading into the first cluster and 63 to the other.

4) Approaches: Multiple approaches have been implemented as simulation agents. The replanning approach is the Spatial Replanner with improved cost model [2]. Hindsight planning based approaches use 100 determinizations to sample resource states. Hindsight planning with reservations is based on the resource chosen in most determinizations. Probability adaptions perform 30 biased random walks within an isochrone of 300 seconds driving time around the most chosen resource. Two baseline approaches not using availability information are included: A simple agent driving to the destination street and then taking random streets until an available parking spot is found. Secondly, Parkagent [23] with its default parameters to simulate a human's search behavior.

B. Evaluation Results

Table I shows mean parking times of evaluated approaches. In settings with real-world occupations, replanning with reservations has the best results, closely followed by hindsight planning with adaptions. In synthetic settings, hindsight planning beats replanning. In synthetic single destination settings, adaptions are best. Reservations are slightly better in data-driven settings. All multi-agent approaches provide effective parking guidance and always perform better than Parkagent.

To evaluate the effectiveness of multi-agent improvements, we compare the relative parking time difference of agents with multi-agent improvements against their baseline approaches:

	Single Destination		Data-Driven Destinations	
Algorithm	Real	Synthetic	Real	Synthetic
RPL + R	84.11	71.11	18.11	3.91
HS + R	78.85	44.85	22.69	22.93
HS + A	78.38	50.15	75.17	14.44

TABLE II: Reduction in percent of the total parking time by multi-agent improvements over their corresponding single-agent approach in various fully observable settings. "RPL" is Replanning; "HS" Hindsight Planning; "+ R" Reservations; "+ A" Probability Adaptions.

Table II shows a significant reduction in parking times, in all settings, using our multi-agent approaches. For example, in the data-driven destination setting with real-world occupations, adaptions bring a 75% reduction compared to single-agent hindsight planning. Overall, improvements lie between around 79% and 14%. Hindsight planning with adaptions is often more effective than hindsight planning with reservations. Reservations with replanning significantly reduce parking time, especially in real-world occupation and single destination scenarios. However, in the data-driven destination setting with synthetic occupations, it was only decreased by around 4%. Multi-agent improvements aim to reduce conflicts between agents with similar destinations. Our experiments show that this is achieved, as no such unsuccessful resource claims occur anymore when reservations or probability adaptions are used.

The median planning computation time per trip is around 0.1 seconds for replanning approaches and about 10 seconds for hindsight planning approaches. Reservations and adaptions do not have a significant influence on computation time. As our experiments contain hundreds of agents and thousands of parking spots, this leads to the conclusion that all approaches can be used in real-time, even in large-scale scenarios.

VII. Conclusion

In this paper, we proposed multi-agent variations of existing approaches for solving the dynamic resource routing problem in fully observable scenarios: We formalized the problem and presented two approaches to solve it: Reservations and dynamic probability adaptions. Each was solved by a Replanner and a Hindsight Planner. Agent-based simulations were conducted to gain insights into their effectiveness and the impact of sharing fleet data. Our experimental evaluation shows that both are able to improve parking guidance significantly. In situations with very few available parking spots, hindsight planning with reservations or adaptions can deliver the best results. Replanning with reservations does work very well in settings close to the real world. It is efficient, easy to implement and can provide parking guidance without a prediction model. We conclude that all approaches presented in this paper can, given their effectiveness and efficiency even in large-scale scenarios, be deployed in real-world parking guidance systems of a vehicle fleet.

Acknowledgments

We thank the City of Melbourne, Australia, for providing the parking datasets used in this paper. This work has been funded by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036A. The authors of this work take full responsibilities for its content.

References

[1] D. C. Shoup, "Cruising for parking," *Transport Policy*, vol. 13, no. 6, pp. 479–486, 2006.
[2] S. Schmoll, S. Friedl, and M. Schubert, "Scaling the dynamic resource routing problem," in *Proceedings of the 16th International Symposium on Spatial and Temporal Databases*, 2019, pp. 80–89.
[3] F. Al-Turjman and A. Malekloo, "Smart parking in IoT-enabled cities: A survey," *Sustainable Cities and Society*, vol. 49, 2019.
[4] P. Basu and T. D. C. Little, "Networked Parking Spaces : Architecture and Architecture of a Parking Meter Network," *Science*, 2002.
[5] K. S. Liu, J. Gao, X. Wu, and S. Lin, "On-street parking guidance with real-time sensing data for smart cities," in *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2018, pp. 1–9.
[6] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of computer computations*. Springer, 1972, pp. 85–103.
[7] Y. Geng and C. G. Cassandras, "Dynamic resource allocation in urban settings: A "smart parking" approach," in *2011 IEEE International Symposium on Computer-Aided Control System Design (CACSD)*. IEEE, 2011, pp. 1–6.
[8] V. Verroios, V. Efstathiou, and A. Delis, "Reaching available public parking spaces in urban environments using ad hoc networking," *Proceedings - IEEE International Conference on Mobile Data Management*, vol. 1, pp. 141–151, 2011.
[9] D. Ayala, O. Wolfson, B. Xu, B. Dasgupta, and J. Lin, "Parking slot assignment games," *Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, pp. 299–308, 2011.
[10] S. Schmoll and M. Schubert, "Dynamic resource routing using real-time dynamic programming," *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2018-July, no. Vi, pp. 4822–4828, 2018.
[11] F. Bock, S. Di Martino, and M. Sester, "What is the impact of on-street parking information for drivers?" in *International Symposium on Web and Wireless Geographical Information Systems*. Springer, 2019, pp. 75–84.
[12] G. Jossé, K. A. Schmid, and M. Schubert, "Probabilistic resource route queries with reappearance," *EDBT 2015 - 18th International Conference on Extending Database Technology, Proceedings*, pp. 445–456, 2015.
[13] S. Yoon, A. Fern, R. Givan, and S. Kambhampati, "Probabilistic planning via determinization in hindsight," *Proceedings of the National Conference on Artificial Intelligence*, vol. 2, pp. 1010–1016, 2008.
[14] B. C. Dean, "Shortest paths in fifo time-dependent networks: Theory and algorithms," *Rapport technique, Massachusetts Institute of Technology*, p. 13, 2004.
[15] L. Foschini, J. Hershberger, and S. Suri, "On the complexity of time-dependent shortest paths," *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 327–341, 2011.
[16] R. W. Floyd, "Algorithm 97: Shortest Path," *Commun. ACM*, vol. 5, no. 6, p. 345, jun 1962.
[17] S. Pettie, "A new approach to all-pairs shortest paths on real-weighted graphs," *Theoretical Computer Science*, vol. 312, no. 1, pp. 47–74, 2004.
[18] H. Samet, J. Sankaranarayanan, and H. Alborzi, "Scalable network distance browsing in spatial databases," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008, pp. 43–54.
[19] Y. Azar, A. Z. Broder, A. R. Karlin, N. Linial, and S. Phillips, "Biased random walks," *Combinatorica*, vol. 16, no. 1, pp. 1–18, 1996.
[20] B. X. Robert van Barlingen, João Ferreira, Tijana Klimovic, Jeroen Schols, Wouter de Vries, "COMSET-GISCUP," 2019. [Online]. Available: https://github.com/Chessnl/COMSET-GISCUP
[21] R. C. Browning, E. A. Baker, J. A. Herron, and R. Kram, "Effects of obesity and sex on the energetic cost and preferred speed of walking," *Journal of applied physiology*, vol. 100, no. 2, pp. 390–398, 2006.
[22] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.
[23] I. Benenson, K. Martens, and S. Birfir, "Parkagent: An agent-based model of parking in the city," *Computers, Environment and Urban Systems*, vol. 32, no. 6, pp. 431–439, 2008.

Chapter 5

DEAR: Dynamic Electric Ambulance Redeployment

Publication

Lukas Rottkamp, Niklas Strauß, and Matthias Schubert. "DEAR: Dynamic Electric Ambulance Redeployment". In: *Proceedings of the 18th International Symposium on Spatial and Temporal Data*. SSTD '23. Calgary, AB, Canada: Association for Computing Machinery, 2023, pp. 11–20. ISBN: 9798400708992. DOI: 10.1145/3609956.3609959

Contribution

Lukas Rottkamp and Niklas Strauß share first authorship of this publication. Niklas Strauß and Matthias Schubert identified the need to extend existing ambulance redeployment work to be suitable for electric vehicles. Together with Lukas Rottkamp, they devised the necessary extension and discussed possible solutions. Niklas Strauß contributed the idea of determining energy differences between areas to use them in a solution. Lukas Rottkamp devised and formalized an agent based on this idea. Lukas Rottkamp modified an existing simulator (for non-EV ambulance redeployment problem) to include battery states and related concepts. Lukas Rottkamp implemented the energy-related agents and ran evaluations. Niklas Strauß and Matthias Schubert contributed to this process by regular discussions. Lukas Rottkamp wrote the initial version of the manuscript. Niklas Strauß overhauled the two sections regarding the formal problem definition and the proposed heuristic including simplifying some of the equations previously introduced by Lukas Rottkamp. All authors further polished the text of the manuscript before submission.

DEAR: Dynamic Electric Ambulance Redeployment

Lukas Rottkamp*
MCML, LMU Munich
Munich, Germany
rottkamp@cip.ifi.lmu.de

Niklas Strauß*
MCML, LMU Munich
Munich, Germany
strauss@dbs.ifi.lmu.de

Matthias Schubert
MCML, LMU Munich
Munich, Germany
schubert@dbs.ifi.lmu.de

ABSTRACT

Dynamic Ambulance Redeployment (DAR) is the task of dynamically assigning ambulances after incidents to base stations to minimize future response times. Though DAR has attracted considerable attention from the research community, existing solutions do not consider using electric ambulances despite the global shift towards electric mobility. In this paper, we are the first to examine the impact of electric ambulances and their required downtime for recharging to DAR and demonstrate that using policies for conventional vehicles can lead to a significant increase in either the number of required ambulances or in the response time to emergencies. Therefore, we propose a new redeployment policy that considers the remaining energy levels, the recharging stations' locations, and the required recharging time. Our new method is based on minimizing energy deficits (MED) and can provide well-performing redeployment decisions in the novel Dynamic Electric Ambulance Redeployment problem (DEAR). We evaluate MED on a simulation using real-world emergency data from the city of San Francisco and show that MED can provide the required service level without additional ambulances in most cases. For DEAR, MED outperforms various established state-of-the-art solutions for conventional DAR and straightforward solutions to this setting.

CCS CONCEPTS

• **Information systems → Spatial-temporal systems**; • **Computing methodologies → Simulation environments**.

KEYWORDS

Ambulance Redeployment, Optimization, Spatio-Temporal Data

ACM Reference Format:
Lukas Rottkamp, Niklas Strauß, and Matthias Schubert. 2023. DEAR: Dynamic Electric Ambulance Redeployment. In *Symposium on Spatial and Temporal Data (SSTD '23), August 23–25, 2023, Calgary, AB, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3609956.3609959

1 INTRODUCTION

The Emergency Medical Service (EMS) is a critical part of health infrastructure all over the world [15]. Paramedics are often the first professional aid in health emergencies and are responsible for safe and quick transport to a secondary care unit such as a hospital. A low response time to emergency calls has increased survival and recovery rates in life-threatening health conditions such as cardiac arrest [5, 18]. Ambulance response times to emergencies depends on various factors, such as the emergency call itself, the processing time needed for dispatch, the readiness of a qualified paramedic team, and its travel time to the incident location. Travel time is a substantial factor. While it can be accelerated by using high-powered vehicles and specialized training for driving in emergency conditions, the initial distance of the ambulance to the incident site is the most prominent factor, with various approaches trying to minimize this distance by proper ambulance placement.

Today, most ambulances are outfitted with internal combustion engines (ICE) using fossil fuels. However, the growing public demand for less air pollution and less release of greenhouse gases promotes the transition towards electric vehicles (EV). Electric ambulances further come with additional benefits, such as a smoother acceleration improving in-ambulance care. Thus, a first generation of electric ambulances is already commercially available.

Ambulances are usually positioned at base stations strategically placed over a city or coverage area to minimize incident response times. Incoming emergency calls are assigned to an ambulance, which drives to the incident location. Some incidents can be resolved on-site, while in other cases, patients need to be transported to a hospital. After completing their assignment, ambulances return to a base station. While ambulances could return to their origin station, it is often advisable to select another base station based on the actual ambulance distribution at this time. This selection of base stations is known as the Dynamic Ambulance Redeployment (DAR) problem in literature [13, 16, 23].

In this paper, we show that existing approaches do not perform well when confronted with electric ambulances. First, we present a formal definition of the Dynamic Electric Ambulance Redeployment Problem (DEAR), which extends existing DAR formalizations by battery levels, range restrictions, charging stations, and recharging. Based on this extension, we can examine the performance of established state-of-the-art methods for dynamic ambulance redeployment, which do not consider these aspects. Afterwards, we present the minimizing energy deficits (MED) approach, designed to avoid these shortcomings and provide state-of-the-art ambulance redeployment for E-Ambulances. Our method is based on matching the predicted future demand in the area of each base station to the joint energy level of the ambulances. The energy level of vehicles at a base station is extrapolated for the same time frame as the future demand and considers any recharging activity increasing the energy level. Based on both estimations on future development, MED assigns ambulances to those base stations where the deficits between the energy level and the demand are expected to be the largest. We compare MED to various state-of-the-art conventional

*Both authors contributed equally to this research.

SSTD '23, August 23–25, 2023, Calgary, AB, Canada

ACM ISBN 979-8-4007-0899-2/23/08.
https://doi.org/10.1145/3609956.3609959

ambulance redeployment methods on an extended environment of [23]. Our results demonstrate that the conventional DAR methods suffer significant performance decreases in various settings. In contrast, MED can cope well with the requirements of E-Ambulances, often compensating for their drawbacks against using conventional ICE ambulances.

To summarize, our contributions are as follows:

- We formalize DEAR, an extension of the DAR problem considering electric ambulances.
- We extended a DAR simulation environment based on real-world data to consider the DEAR setting and examine the performance of conventional DAR methods.
- We propose MED and present experimental results showing that it copes well with DEAR compared to existing DAR methods and basic DEAR approaches.

The remainder of this paper is structured as follows: Related work is presented in Section 2. We then formulate the Dynamic Electric Ambulance Redeployment Problem (DEAR) in Section 3 and propose MED in Section 4. We evaluate established DAR approaches and MED for DEAR using a simulation based on real-world incident data from San Francisco in Section 5 and summarize our work in Section 6.

2 RELATED WORK

The ambulance location problem (ALP) is an established research topic. Existing approaches can be classified into static and dynamic methods: In static methods, ambulances are stationed at fixed base stations and always return to the same base station after an incident has been handled [7, 8, 19]. One way to obtain a static assignment is to solve the *Maximum Expected Covering Location Problem* (MEXCLP) [8, 13]. Its solution maximizes the expected coverage of incident locations. In contrast to the *Maximum Coverage Location Problem* [7] it is based on, the underlying model assumes an ambulance to be *busy* with a certain probability. In this way, ambulances that are unavailable due to being on a mission, are not included in the coverage calculation. This reasonable modification has been proven to be advantageous compared to earlier methods [12, 13]. *Expected Response Time Model* (ERTM) [3] is another static approach that has shown excellent performance due to its direct minimization of the expected response time [3, 23].

Current state-of-the-art ALP solutions use a dynamic assignment due to the volatility of the problem [13]. The dynamic assignment of ambulances is also called *Real-Time Ambulance Redeployment Problem* or *Dynamic Ambulance Redeployment Problem* (DAR). Dynamic redeployment leads to better response times than static return policies because the stochastic nature of incoming emergency calls can lead to imbalances in ambulance distribution which are ignored by static approaches [10, 11]. The redeployment decision is primarily based on the locations of ambulances and base stations but may also take other factors, such as demand distributions, into account. The *DMEXCLP* approach by [13] is a dynamic variation of MEXCLP. At each redeployment step, it selects the base station providing the largest coverage increase in the respective situation according to the MEXCLP strategy. This way, DMEXCLP takes the actual distribution of ambulances into account. A reinforcement-learning based approach "Reinforcement Learning Deep Score Network" (DRLSN) is presented by [14].

A vision paper by [20] highlights the growing importance of electric ambulances and the associated challenge of keeping a fleet of ambulances charged. It suggests a high-level framework for ambulance scheduling concerning the optimal use of renewable energy sources, including predictive components for patient demand and energy production and use. Though this work is related, it does neither propose a formalization of DEAR nor does it provide a method for the redeployment problem for electric ambulances.

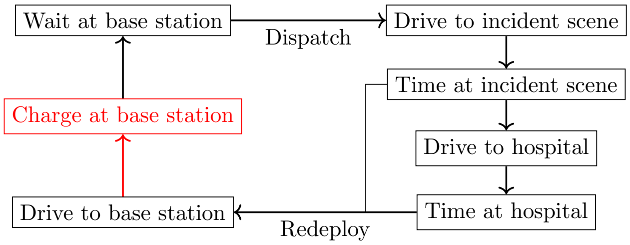


Figure 1: Simplified schematic overview of the modeled EMS process. Specifics for electric ambulances are shown in red.

3 PROBLEM DEFINITION

In this section, we will provide a formal definition of the Dynamic Electric Ambulance Redeployment (DEAR) problem, outlining the operational process of the Emergency Medical Services (EMS) provider and considering the specifics of electric ambulances. Figure 1 provides a visual representation of the EMS process. When an incident occurs, the EMS operator receives a call, and an available ambulance is dispatched from a base station to the incident location. In our scenario, the ambulance closest in driving time is dispatched to ensure a prompt response. If no ambulance is available, the incident is handled as soon as an ambulance becomes available again. Upon arrival at the incident, on-site care is provided to the patient. Depending on the patient's condition, subsequent transport to a hospital may be necessary. Otherwise, the ambulance is redeployed from the incident site to a base station. Once the ambulance arrives at a base station, it becomes idle and available for dispatch. Considering electric vehicles introduces unique challenges compared to Internal Combustion Engine (ICE) vehicles. The downtime for refueling ICE vehicles is typically not a significant concern due to their long ranges and fast refueling times. However, electric vehicles have shorter ranges and require substantial charging time. Therefore, factors such as charging downtime, battery levels, and the availability of fast chargers at base stations need to be considered in the EMS process. It is crucial only to dispatch an electric ambulance if its battery is sufficiently charged to not run out of energy while handling the incident. Therefore, we define a minimum dispatch range τ_{MDR} (measured in time units) as the worst-case trip, starting from the current base station to any incident location, followed by transportation to any hospital, and finally redeployment to a base station.

Electric ambulances can be charged at regular AC outlets (we refer to them as slow chargers), which are already available in large numbers at base stations. However, slow chargers have limited power output, resulting in extended charging times and longer downtimes of ambulances. Charging times can be significantly reduced by installing high-voltage DC chargers (fast chargers specifically installed for electric vehicles) at base stations. However, their number is limited because installation presents a significant cost factor and constraints caused by the capabilities of the energy grid.

Assigning chargers to ambulances at a base station requires a charging policy when the number of ambulances exceeds the number of fast chargers. The objective is to charge ambulances in a manner that allows them to reach the minimum dispatch range τ_{MDR} as quickly as possible, thereby maximizing the number of available ambulances. It is also important to avoid an unreasonably high number of re-plugging actions by staff. To achieve these goals, we implement the following approach: Ambulances below τ_{MDR} are categorized as high-priority and are charged first. If there are more high-priority ambulances than available chargers or fast chargers, the ambulance with the shortest time required to reach τ_{MDR} is prioritized for charging. This ensures that ambulances are prepared for service at the earliest possible time. Once an ambulance reaches τ_{MDR}, it becomes a low-priority ambulance. For charging low-priority ambulances, we prioritize ambulances with the lowest battery level to minimize the number of re-plugging actions. Re-plugging can occur when an ambulance at the station is sufficiently charged to provide the minimum dispatch range, is fully charged, arrives, or is dispatched.

Now, we present a formal definition of the novel DEAR problem, considering the aforementioned characteristics. In this task, an operator needs to dynamically select a base station to redeploy an ambulance to after the ambulance finishes handling an incident, either from the incident site or the hospital.

The road network is represented as a graph $G = (V, E)$, where V is the set of nodes representing locations in the road network, and E is the set of directed edges representing road segments connecting the nodes.

Incidents are emergencies requiring medical attention by an ambulance and are denoted as I. Each incident is mapped to the nearest node in the graph.

Base Stations Let W be the set of base stations available within the road network, where ambulances are stationed and dispatched to incidents. Each base station is mapped to the closest node in the road network. Base stations are equipped with charging infrastructure to support the operation of electric ambulances. They possess an unlimited number of slow chargers (regular AC outlets) and have varying numbers of fast chargers. Not all base stations are guaranteed to have fast chargers available.

Hospitals The set H represents the hospitals. Similar to base stations, hospitals are mapped to the closest node in the graph.

Ambulances are electric vehicles, introducing specific characteristics that affect their operational constraints. Key properties include battery level and capacity, energy use per time, and charging characteristics. The charging rate of an ambulance depends on various factors, including its current battery level and the power output of the charger. A linear charging function is utilized, although other charging functions may also be employed. We assume that all ambulances are the same type, i.e., their key properties are equal. Let us note that our method can easily be adapted to more specific settings if required. Ambulances are initially assigned to base stations and can be dynamically redeployed to other base stations depending on incident demand. We allow an ambulance to be redeployed only after finishing handling an incident.

Travel Times In our setting, the travel times $\tau(i, j)$ between two nodes $i, j \in V$ are assumed to be deterministic and do not vary with traffic conditions. When responding to an incident or transporting a patient, ambulances use lights and sirens to alert other drivers, enabling them to travel at fast speeds [4]. We denote the travel time with lights and sirens activated as $\hat{\tau}(i, j)$.

4 MED: MINIMIZE ENERGY DEFICIT

In this section, we introduce our approach Minimize Energy Deficit (MED) for the DEAR problem.

While approaches for solving the DAR problem can be applied, they do not take the additional complexity of electric ambulances into account. Our evaluation demonstrates that this leads to drastically degraded response times or requires multiple additional ambulances to maintain EMS service levels compared to combustion engines.

Thus, it is crucial for a redeployment policy to take battery levels and charging into account. MED is based on the concept of matching the anticipated energy demand at different stations with the expected energy supply at those stations. While the energy demand depends on the incidents and, consequently, the amount of energy needed to handle all incidents. On the other hand, the expected energy supply depends mainly on the distribution of ambulances across the base stations, which is influenced by redeployment decisions. Whenever a redeployment decision needs to be made, our approach deploys the ambulance to the station, which minimizes the energy deficit.

Our proposed method consists of three steps described in the remainder of this section:

(1) Determine the expected energy demand.
(2) Determine the expected energy supply.
(3) Calculate and minimize the energy deficit.

4.1 Expected Energy Demand

We introduce the concept of energy demand θ_w, which refers to the expected energy required to handle incoming incidents within the lookahead duration Δt at base station w. It is determined based on the expected number of incidents in the vicinity of the base station $d_w(t_{\text{now}}, \Delta t)$ during the lookahead duration and an expected energy use per incident ρ_w. The expected energy demand can be expressed as the product of these:

$$\theta_w = d_w(t_{\text{now}}, \Delta t)\rho_w \tag{1}$$

We define the demand forecast $d_w(t_{\text{now}}, \Delta t)$ as a function that estimates the expected number of incidents in the vicinity of the station w from the current time t_{now} until $t_{\text{now}} + \Delta t$. Numerous approaches have been proposed in the literature for predicting ambulance demand [21, 22, 25, 26]. These methods include but are not limited to machine learning techniques, time series analysis, and statistical models. In this paper, we compute an hourly historical average for demand prediction. [6] shows that this method yields a strong baseline for predicting ambulance demand. Let us note that our approach does not depend on a specific forecasting method and likely benefits from more accurate predictions. We leave the exploration of more sophisticated demand models to future research.

The vicinity V_w of a base station w is defined to be the incident locations $i \in V$ where the travel time $\tau(w, i)$ is shorter than from any other station. Mathematically, this can be expressed as follows:

$$V_w = \{i \in V | \tau(w, i) \leq \min_{w' \in W} \tau(w', i)\} \tag{2}$$

Using historical incident data, we calculate the average number of incidents per hour $\kappa_w(h)$ in the vicinity of each base station w and each hour of day $h \in \{0, ..., 23\}$. Let $\beta_h \in [0, 1]$ represent the fraction of hour h in the time interval $[t_{\text{now}}, t_{\text{now}} + \Delta t]$. The demand forecast is then given by:

$$d_w(t_{\text{now}}, \Delta t) = \sum_{h \in \{0,...,23\}} \beta_h \kappa_w(h) \tag{3}$$

Determining the expected energy per incident within the proximity of each station holds significant importance. This necessitates evaluating the energy expenditure for traveling from a base station to the incident location, potentially to a hospital and returning to a station. A simplistic approach would assume that incidents solely occur at the centers of each demand area (i.e., the base stations), then travel to the nearest hospital, and finally return to the closest station. However, such an approach lacks accuracy. Therefore, we assume that the locations of incidents are uniformly spatially distributed across all possible incident locations $i \in V_w$ in the vicinity of w. We consider the probability of requiring transportation to a hospital, as well as accounting for the distribution of patients transported to different hospitals and the expected energy for redeployment to a station. The hospital distribution and the probability of requiring hospital transportation are derived from historical data.

We denote the proportion of incidents requiring hospital transportation as α, while α_h is the fraction of these incidents handled by hospital h. To calculate the expected energy use per incident ρ_w in the vicinity of station w, we first determine the expected driving time for fully handling an incident and redeployment to a station. Subsequently, we estimate the energy usage by multiplying the resulting driving times with the parameter $P_{driving}$, which approximates the energy consumed per unit of time:

$$\mathbb{E}(\rho_{hospital}(i)) = \sum_{h \in H} \alpha_h \frac{1}{|W|} \sum_{w' \in W} (\hat{\tau}(i, h) + \tau(h, w')) \tag{4a}$$

$$\mathbb{E}(\rho_{base}(i)) = \frac{1}{|W|} \sum_{w' \in W} \tau(i, w') \tag{4b}$$

$$\rho_w = P_{driving} \frac{1}{|V_w|} \sum_{i \in V_w} \hat{\tau}(w, i) + \alpha \mathbb{E}(\rho_{hospital}(i)) + (1-\alpha) \mathbb{E}(\rho_{base}(i)) \tag{4c}$$

4.2 Expected Energy Supply

This section focuses on outlining the methodology for calculating the expected energy supply ϕ_w, at a base station w over a specific time period. The actual energy supply depends on the demand, as ambulances may leave the base station to respond to incidents. While it is theoretically possible to model the distribution of incidents and to sample from an exponentially expanding set of future scenarios to derive estimates, finding optimal solutions is computationally intractable. Even approximations similar to the hindsight planning approaches in [24] are impracticable due to the inherent complexity and real-time constraints of the DEAR problem. To address this, we propose calculating an optimistic, expected energy supply $\hat{\phi}_w$, assuming that no incidents occur and no ambulances are redeployed during the prediction horizon, effectively disregarding the demand. This simplification allows for a deterministic calculation. However, we account for the probability of ambulances being dispatched and subsequently reducing the energy supply during the lookahead duration Δt. This is achieved by introducing a charging discount factor $\gamma \in [0, 1]$ to adjust the expected energy supply, resulting in $\phi_w = \gamma \hat{\phi}_w$. Note that even with those assumptions, determining the expected energy supply still requires simulating the complex charging logic and considering the arrivals of ambulances en route to the base station.

4.3 Minimize Energy Deficit

After we have defined the expected energy demand and supply, we continue by specifying how to calculate the energy deficits and subsequently dynamically redeploy ambulances. To define the energy deficit δ_w at a specific base station w, we calculate the difference between the expected energy demand and supply: $\delta_w = \theta_w - \phi_w$. However, simply minimizing this deficit has certain limitations. For instance, if a station already has sufficient supply to meet the demand, adding more supply would be unnecessary, even if it reduces the deficit. Therefore, we introduce a weighted deficit ω_w using a soft plus function [9]. This function assigns lower importance to stations with negative deficits (i.e., surplus supply compared to demand) and prioritizes stations with high deficits. The weighted deficit is calculated as follows:

$$\omega_w = \log(1 + exp(\frac{1}{100} \delta_w)) \tag{5}$$

In the last step, we describe the methodology for using the weighted energy deficit ω_w to make redeployment decisions. Whenever an ambulance a needs to be redeployed, we simulate sending the ambulance to each base station w to obtain $\omega_w(a)$. This is used to calculate the reduction in the expected weighted energy deficit $\omega_w - \omega_w(a)$ at each station. Subsequently, we redeploy the ambulance to the station that yields the most significant reduction.

4.4 Computational Complexity

Making ambulance redeployment decisions is a time-critical task, and any method should be able to compute a redeployment decision within seconds. Consequently, we designed our approach with this requirement in mind. To make each redeployment decision, we must assess the expected energy demand ρ_w and the expected energy supply ϕ_w at each station w both with and without the ambulance being redeployed. The energy demand consists of two components, the expected number of incidents and energy use per incident. The complexity of the former depends on the demand prediction model used. In this paper, we use the historical average, which can be pre-computed so that a prediction can be made in constant time. The second component, the expected energy use per incident, is a constant factor that can also be pre-computed. Therefore, calculating the expected energy demand has constant complexity. The primary computational effort lies in determining future energy supplies, which involves simulating the charging logic of each ambulance. As we assume optimistically that ambulances will not be deployed, they will eventually reach full charge, and their energy supply will no longer change. In other words, each ambulance adds a particular constant computational effort to simulate. From a computational point of view, the time complexity of determining future energy supplies is linear in the number of ambulances, resulting in a complexity of $O(|A|)$.

Our approach has an overall worst-case time complexity of $O(|A||W|)$. For each station, we need to calculate the expected energy demand (with constant complexity due to pre-computation) and compute the expected energy supply twice.

We implemented our method in C++ to obtain evaluation results presented in the next section. Executed on a notebook with Intel® Core™ i7-10750H CPU, one redeployment decision is obtained in approximately 0.23 milliseconds during a typical evaluation run with 45 base stations and 25 ambulances. Repeating the measurements with 1,000 ambulances in the environment (an unreasonably high number for benchmark purposes only), one decision is obtained in approx. 0.26 ms. These results satisfy the real-time requirement.

5 EVALUATION

In this section, we evaluate various solutions in a DEAR setting based on real-world emergency data from the city of San Francisco. We will first detail our experimental setup and, afterward, examine the impact of electric ambulances on DAR solutions and the performance of our newly proposed method *MED*.

5.1 Simulation environment

We evaluate various scenarios using an event-based simulator that replays real-world emergency data. This simulator mirrors the DEAR problem defined in Section 3 to simulate the operations of the EMS with electric ambulances. The foundation of our simulator is an openly accessible simulation environment for dynamic ambulance redeployment developed by [23]. Since this simulation does not consider electric vehicles, we extended it to include vehicles' battery state, charging, and energy use. Further, base stations were modified to contain a definable number of chargers of specified charging power with the problem definition's charging logic. Note that charging electric vehicles is a complex process influenced by factors such as battery level, battery condition, and ambient temperature. Similarly, energy usage depends on variables like driving profile, traffic conditions, and secondary loads such as heating or equipment required for patient care. Given the complexity of modeling these factors accurately, we simplify our simulation by utilizing constant values for charging power and driving energy usage, respectively.

The simulated EMS system is based on the city of San Francisco, USA. The system contains eleven hospitals and 45 base stations. Their locations are depicted in Figure 2. The road network graph used in the simulation was acquired from OpenStreetMap [1], with intersections representing the graph nodes. Hospitals and base stations were attached to the nearest node in the graph. Driving times

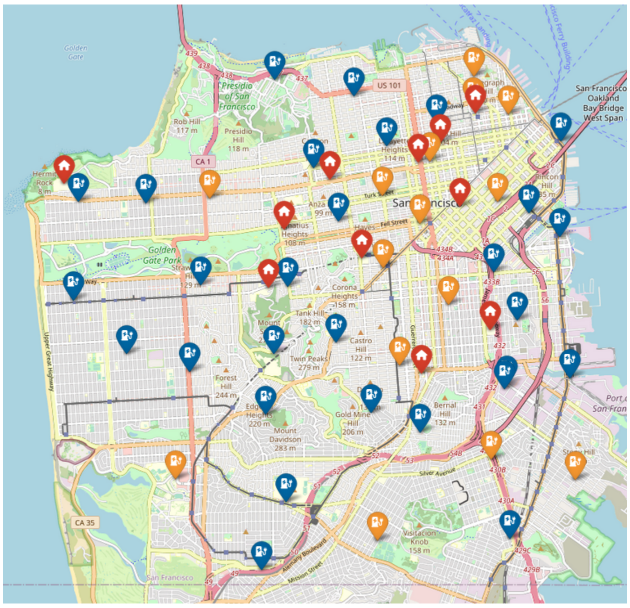


Figure 2: Simulation environment of San Francisco, USA. Locations of base stations are marked in orange if a fast charger is present and otherwise, in blue; Hospitals are marked in red. Note that population density is highest in the downtown area (top right). Map data © OpenStreetMap contributors.

were computed based on the shortest path with respective street limits depending on the road type. To account for traffic and slowing down due to turns and crossings, we calibrated the driving times based on estimates by HERE Traffic [2] by multiplying a constant factor. Based on this method, the average speed, including traffic congestion, was estimated to be 32 $\frac{km}{h}$. Ambulances returning to a base station are assumed to drive at traffic speed. However, when moving toward an incident or hospital, ambulances are granted certain exemptions from traffic regulations, allowing them to drive faster. Nevertheless, traffic congestion and safety considerations still limit realistic driving speed. Thus, we scaled driving times accordingly, resulting in an average emergency speed of 50 $\frac{km}{h}$ as suggested by [11].

The city of San Francisco has made real incident data publicly available through their *Fire Department Calls for Service* dataset [3]. This dataset contains historical records of health emergency calls, including information such as the date, time, and location of each emergency. This enables us to simulate the historical occurrence of incidents with arbitrary configurations of base stations, ambulances, and redeployment methods. However, it is important to note that while the dataset indicates whether a hospital was targeted, specific details such as the hospital's name or location are not disclosed. Selecting a suitable hospital involves a complex decision process, including various factors such as the patient's medical needs, hospital occupancy levels, patient preferences, and the proximity to hospitals [1]. Since this information is unavailable to us,

[1] ODbL license https://www.openstreetmap.org/copyright. Map data copyrighted by OpenStreetMap contributors and available from https://www.openstreetmap.org.

[2] https://www.here.com/platform/traffic-solutions/real-time-traffic-information

[3] https://data.sfgov.org/Public-Safety/Fire-Department-Calls-for-Service/nuek-vuh3

we determine target hospitals by random sampling according to the real-world distribution of patient transports to hospitals between February 2022 and February 2023 published by the Data Working Group (DWG) of the City of San Francisco [4]. Note that this random sampling was done as a preprocessing step, ensuring the hospital transportations are consistent across all experiments. Locations of incidents were mapped to the nearest graph node in our simulation.

5.1.1 Placement and power of fast chargers. As discussed in our introduction, cities will likely outfit only a subset of base stations with fast chargers, primarily due to installation costs. Therefore, we strategically locate fast chargers at the stations with the highest demand, assigning one fast charger per station. Note that according to our problem definition, additional slower chargers are already present at every station. In our evaluation, we considered three different types of fast chargers, each offering different charging powers. The first type is a high-power DC charger delivering 50 kW of charging power. The second type is a cheaper three-phase AC charger delivering 22 kW charging power. Lastly, we considered a more expensive option of 100 kW charging.

5.1.2 Electric Ambulance Models. The battery capacity and average driving energy use (P_{driving}) in our simulation are based on real-world electric ambulances. We base our experiment values on electric ambulance "WAS 500" because it is a suitable replacement for ICE ambulances and technical data is readily available [5]. We set the battery capacity to 87 kWh, based on the specifications provided in the datasheet of the ambulance. To determine P_{driving}, we consider the average speed and the energy usage from the datasheet. We calculate this value as 30 kW.

5.2 Metrics

As motivated in our introduction, minimizing ambulance response times is critical for EMS providers. In an ambulance redeployment context, response times are usually defined as the time between dispatching an ambulance at its base station and its arrival at the incident scene. Aggregated metrics used for evaluating the performance of EMS systems are the average response time (**ART**) and the fraction of response times within a certain response time threshold (RTT) [17, 23]. RTT values and targeted fractions are set differently by different institutions [17]. San Francisco's Emergency Medical Services Agency aims to arrive at life-threatening incidents within a 10 minute threshold at least 90% of the time [2, 23]. We use this metric extensively in our evaluation, denoting it as **RTT10**. We occasionally also include RTT fractions for 8 minutes (**RTT8**) and 12 minutes (**RTT12**).

5.3 Baselines

We compare our method **MED** (Minimize Energy Deficit) with several straightforward baselines as well as several state-of-the-art approaches for redeploying combustion engine ambulances. The most simple baseline is **RAND**, which redeploys the ambulance to a random base station. **NEAR** selects the base station which can be reached fastest by the ambulance (i.e. minimizes driving time). **NEARC** and **NEARF** similarly select the nearest station but consider only stations with chargers (NEARC) or free, fast chargers (NEARF), respectively. Note that this method checks availability at query time. We also include state-of-the-art approaches from the DAR problem discussed in (Section 2) and refer to them as conventional approaches. These approaches consists of static methods, namely **ERTM**[3] and **MEXCLP**[8, 13], a dynamic method called **DMEXCLP**[13], and the reinforcement learning based approach **DRLSN**[14]. Let us note that *DRLSN* is trained in an environment considering DEAR, and thus, it can learn the specific behavior of E-Ambulances. However, we did not change the agent itself as a straightforward extension of observation data did not yield improved results.

[4]http://sfemergencymedicalresponse.weebly.com/ambulance-destinations.html
[5]https://www.was-vehicles.com/en/innovation/was-500-electric-ambulance.html

Table 1: RTT10 performance of conventional methods in the ICE case compared to the EV case with different charging powers and 24 ambulances.

Scenario	ERTM	DRLSN	MEXCLP	DMEXCLP
ICE	0.88	**0.90**	0.83	0.89
EV 22 kW	**0.47**	0.40	0.30	0.20
EV 50 kW	0.72	**0.85**	0.58	0.57
EV 100 kW	0.76	**0.86**	0.67	0.56

5.4 Results

In this section, we present the results of our experimental evaluation based on the previously described simulation environment to answer the following research questions:

(1) How large is the effect of replacing ICE ambulances with EVs using established DAR methods?
(2) Does our approach *MED* perform better than methods from related work for DEAR?
(3) What is the influence of simulation parameters such as the number of available chargers?
(4) How sensitive is our approach to variation of its parameters?

For all experiments, methods were evaluated by simulating one year of incidents (*test set*) in our simulation. The resulting response times were then aggregated to obtain RTT10 and ART metrics. The respective previous year (*validation set*) was used to determine the method's parameters, such as historical demand and selecting hyper-parameters. The best set of hyper-parameters (according to the RTT10 metric) was selected for evaluation on the test set. Experiments were conducted for the years 2015 to 2022. Due to the numerous parameters involved, including different combinations of years, ambulance quantities, charger quantities, charging power, etc., we cannot present all results here. Unless indicated otherwise, the experiments were conducted with incidents from the year 2022, using 15 fast chargers, each providing 50 kW charging power. Additionally, we included variations of these parameters to facilitate a comprehensive comparison of methods under different scenarios.

5.4.1 Effect of switching to electric ambulances. In this section, we analyze the effectiveness of methods for ordinary DAR settings (conventional approaches) when being applied to the DEAR problem. We present the results for ICE and EV scenarios containing 24 ambulances in Table 1, as 24 ambulances are required for the

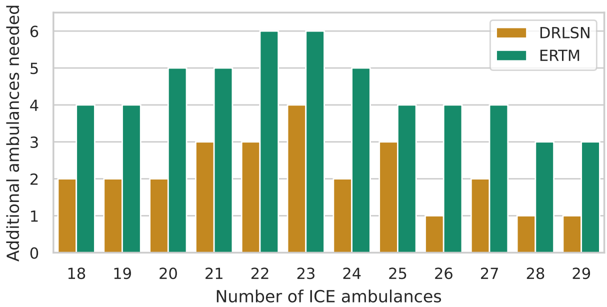


Figure 3: Number of additional ambulances needed to reach same performance (RTT10 metric) as in the non-EV scenario for best methods from related work. 50 kW charging power.

first method to reach the RTT10 target of 90%. We observe a significant decline in the RTT10 metric when introducing energy use and charging, with some cases showing a reduction of more than 50% in performance. Using 22 kW fast chargers results in inferior performance: *ERTM* receives the best 22 kW RTT10 score (0.47), which is not acceptable for an EMS provider, despite its significantly better performance (0.88) in the ICE case. When using 50 kW or 100 kW fast chargers, the decline in performance is less severe but still substantial. *DRLSN* achieves best RTT10 scores in the ICE (0.90), 100 kW (0.86) and 50 kW (0.85) cases. Notably, its reward-based algorithm shows the ability to learn certain characteristics of the EV environment despite not explicitly observing energy-related data. Its poor performance in the 22 kW case may be explained by rewards being too sparse to enable effective training. Like *ERTM*, both *MEXCLP* and *DMEXCLP* show drastic decreases in performance. Although the dynamic method *DMEXCLP* performs better than the static approaches *ERTM* and *MEXCLP* in the ICE case, it experiences substantial difficulties in the EV scenarios, even showing worse results in the 100 kW case compared to 50 kW. Overall, results indicate that using fast chargers with 22 kW charging power will not enable acceptable performance with these methods. Increasing the charging power to 50 kW improved the results, but additional ambulances are still necessary. Installing 100 kW chargers does not appear to improve results substantially. As infrastructure investments generally increase with higher charging power, emergency medical service providers should be aware of this effect when transitioning to electric ambulances.

Figure 3 provides insights into the number of additional ambulances needed when transitioning from ICE to EV ambulances. It depicts the number of additional ambulances required to reach an equal or better RTT10 performance compared to non-EV ambulances for the *ERTM* and *DRLSN*. We use 50 kW chargers in the scenario, as there is a minimal improvement when using 100 kW. *ERTM* requires an additional 3 to 6 ambulances. *DRLSN* requires 2 to 4 additional ambulances when replacing up to 25 ICE ambulances. In settings replacing more than 25 ICE the number of additional ambulances can decrease to 1.

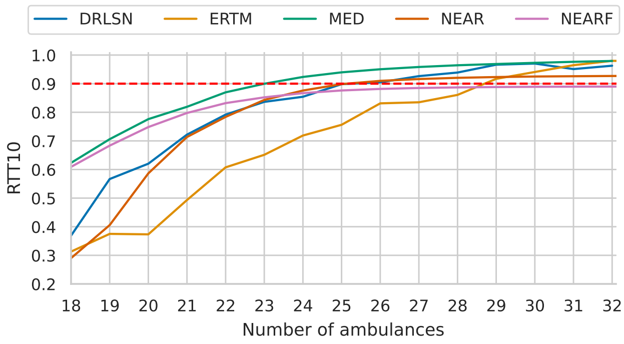
Overall, our results show that employing conventional methods from related work on the DEAR problem requires more ambulances to achieve a similar level of performance compared to ICE ambulances. Additionally, an interesting finding is that the difference between 50 kW and 100 kW charging is minimal in contrast to charging with 22 kW.

Table 2: Performance of all methods when using 24 ambulances and 50 kW charging power.

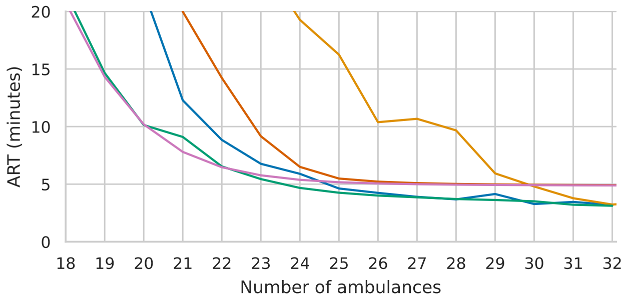
Method	RTT8	RTT10	ART
MED	**0.87**	**0.92**	**4.64**
NEAR	0.79	0.88	6.50
NEARF	0.79	0.87	5.38
DRLSN	0.81	0.85	5.90
NEARC	0.75	0.84	5.89
ERTM	0.68	0.72	19.27
MEXCLP	0.50	0.58	32.44
DMEXCLP	0.49	0.57	33.87
RAND	0.01	0.01	153.09

5.4.2 Performance of MED. We now introduce results for our approach *MED* and compare them to state-of-the-art conventional methods developed for DAR, as well as our DEAR baselines. Results for all methods are shown in Table 2. We again chose 24 ambulances and 50kW charging power due to the previously mentioned practical relevance of this scenario. Our approach *MED* outperforms all other methods across all metrics. Specifically, it achieves an RTT10 value of 0.92, which is well within the 90% target. The average response time (ART) of 4.64 minutes is about 45s faster than the second-best method *NEARF*, and 75s less than *DRLSN*, the best conventional method from related work. It is worth noting that another nearest station method, *NEAR*, also demonstrates surprisingly good performance, securing the second-best RTT10 value of 0.88. The best performing conventional method is *DRLSN* (0.85), followed by *ERTM* (0.72). The difference between RTT10 and ART scores, especially when considering the comparatively good performance of simplistic baselines such as *NEAR* or *NEARF* underlines the observation that conventional methods do not perform well in the evaluated EV scenario. In contrast, *MED* performs better in DEAR (RTT10 of 0.92) than the best DAR approach in the corresponding ICE scenario (RTT10 of 0.90, compare Figure 1).

The relationship between the number of deployed ambulances and the performance is illustrated in Figure 4 for the best-performing methods. *MED* consistently demonstrates strong results across all metrics. Analyzing the RTT10 performance in Figure 4a, it becomes evident that *MED* outperforms other approaches with a substantial gap to the second best method up to a number of 29 ambulances. As noted before, it is the first to exceed the 90% RTT10 target (dashed red line). Furthermore, its performance considering the ART metric (Figure 4b) is superior to others in the most interesting region (due to its closeness to the 90% RTT10 target) of about 24 ambulances. When 22 or fewer ambulances are used, method *NEARF* yields lower ART values. This is because in these cases, demand for ambulances, and the energy use that comes with it, is so high that all other objectives fade in comparison to obtaining energy as fast as possible. As the *NEARF* method is designed to immediately drive to the nearest free charger, regardless of its location or any



(a) RTT10 metric. Dashed line indicates 90% target.



(b) ART metric.

Figure 4: Performance comparison of best methods for 50 kW charging power.

other criteria, it fulfills this objective well. In situations where a substantial number of ambulances (27 or more) are available, methods from previous research narrow the gap. At this point, the locations and availability of chargers become less critical as it becomes more likely that a charged ambulance is stationed sufficiently close to any incident. Furthermore, slow charging is sufficient to make sure that drained ambulances will be available at a later point in time. It is, however, interesting that the gap for the RTT10 metric (c.f. Figure 4a) closes more slowly that the gap in ART (c.f. Figure 4b). This indicates that *MED* still allows significantly fewer incidents that are not handled within the 10-minute limit than compared methods up to 29 ambulances.

As emergency service providers usually aim to fulfill a certain minimum service level, we provide the number of ambulances needed to reach a 90% fraction of common RTT values in Table 3. An important observation is that *MED* requires the lowest number of ambulances to reach the target in all cases. A RTT8 target is reached by deploying 26 ambulances with *MED*, whereas the second best method, *DRLSN*, requires 29 ambulances. The RTT10 and RTT12 targets are reached with 24 and 22 ambulances, respectively, requiring two and one ambulances less than the runner-up. It is worth mentioning that most methods failed to reach the RTT8 target for fleet sizes up to 40, which is the maximum number of ambulances considered in our experiments.

Table 3: Number of ambulances needed to reach the 90% RTT target for various RTT values. 50 kW charging power.

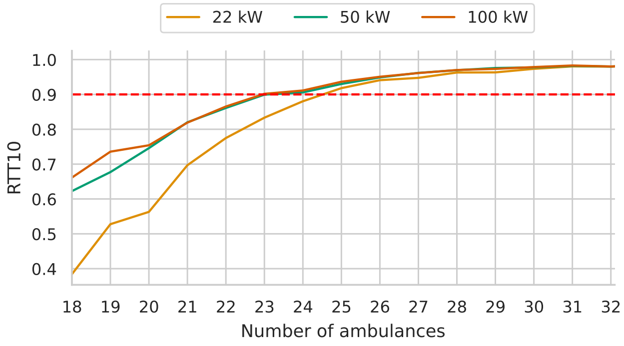
Method	8 min	10 min	12 min
MED	**26**	**24**	**22**
DRLSN	29	26	25
NEAR	> 40	26	24
ERTM	30	29	29
DMEXCLP	> 40	32	31
MEXCLP	> 40	32	29
NEARC	> 40	> 40	23
NEARF	> 40	> 40	23
RAND	> 40	> 40	> 40

Table 4: Performances of MED compared to best method from related work for each evaluation year. In each year, MED performed best, followed by DRLSN. The number of ambulances in each row was determined as the lowest amount that reached 90% RTT10 for the given year. Column *Diff* for RTT10 is the decrease of incidents that could not be reached within 10 minutes. Column *Diff* for ART is the decrease in response times.

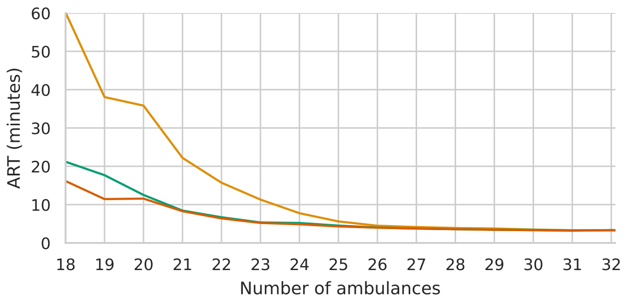
Year	RTT10			ART		
	MED	DRLSN	Diff	**MED**	DRLSN	Diff
2015	0.901	0.860	-29.18%	4.810	5.551	-13.34%
2016	0.907	0.867	-30.57%	5.101	5.916	-13.78%
2017	0.919	0.877	-34.28%	4.494	5.345	-15.92%
2018	0.912	0.867	-33.63%	4.679	5.491	-14.79%
2019	0.910	0.875	-28.19%	4.579	5.469	-16.27%
2020	0.908	0.872	-28.18%	4.635	5.476	-15.37%
2021	0.905	0.852	-35.42%	4.927	5.767	-14.57%

To see if the superior performance of *MED* can be reproduced in other years, we repeated the experiment above for each pair of years starting in 2015. This includes fitting parameters on the given year and testing methods' performance in the following year. The results summarized in Table 4 show that *MED* can reach the 90% RTT10 target with fewer ambulances than methods from related work each year. The difference in incidents that could not be reached within 10 minutes is considerably lower in these cases, namely between 28.18% to 35.42% lower. Average response times also decrease consistently for all years. In absolute numbers, this means reducing average response times by about 50 seconds in our experiments, which can be valuable in critical emergencies.

These results demonstrate the superior performance of *MED* for the DEAR problem across various scenarios. Furthermore, as *MED* in DEAR displays a similar or better performance than compared methods in the ordinary DAR environments based on ICE ambulances, we can conclude that switching to an equally-sized fleet of E-Ambulances can be done without significantly decreasing response times.



(a) RTT10 metric.



(b) ART metric.

Figure 5: Performance of MED for different charging powers.

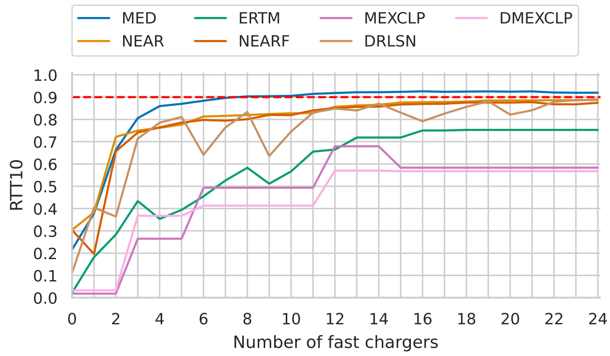


Figure 6: Comparison of RTT10 performance for different numbers of fast chargers. Scenario with 24 ambulances and 50 kW charging power.

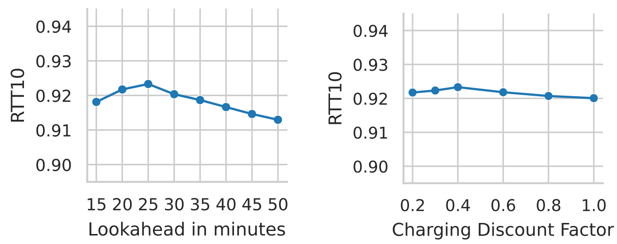


Figure 7: Performance of MED for varying hyper-parameters. Scenario with 24 ambulances and 50 kW charging power.

5.4.3 Varying power and number of chargers. The performance of *MED* for different charging rates and numbers of deployed ambulances is presented in Figure 5. It can be seen again that the difference between 50 kW and 100 kW fast charging power is minimal. However, 22 kW charging results in inferior performance. For example, the RTT10 target is reached with 25 ambulances instead of only 24 for both higher charging rates. This disparity becomes more apparent as the number of ambulances decreases, as the per-ambulance energy use and corresponding charging activity increases in such scenarios. With increasing numbers of ambulances, the charging pressure vanishes, which can be seen in the convergence of all powers' measurements. Figure 6 depicts the results of different methods for varying the number of installed fast chargers. As before, we use 24 ambulances as the lowest amount to be sufficient to reach the 90% RTT10 target. *ERTM*, *MEXCLP*, and *DMEXCLP* exhibit a slow increase of performance when increasing the number of fast chargers and thus appear to be especially ill-suited for the EV scenario. In contrast, the performance of *MED*, *NEAR*, *NEARF* and *DRLSN* follows an early quick increase with a slower rise once about three fast chargers are installed, i.e., they appear to either use less energy or utilize fast chargers better, or both. It should be noted that *MED* is the only approach that meets the 90% RTT10 level. Furthermore, *MED*'s performance does not substantially increase when more than 11 chargers are installed in the environment. To summarize, our method tailored for EV scenarios requires not only fewer ambulances but also fewer fast chargers.

5.4.4 Parameter sensitivity. Figure 7 shows how varying *MED*'s two parameters affect its RTT10 performance, using the scenario of 24 ambulances and 50 kW charging. Examining the parameter lookahead duration Δt (Figure 7 left), the optimal value is 25 minutes, with a roughly linear decrease when higher or lower values are used. The sensitivity of our approach to this parameter is low, as doubling it to 50 minutes only marginally decreases RTT10 performance by about 0.01. Varying the charging discount factor γ (Figure 7 right) appears to have little effect on performance. The optimum is at a value of 0.4, which can be explained by charging processes at base stations being frequently interrupted due to incoming incidents in this challenging scenario.

5.4.5 Qualitative analysis. Figure 8 shows a snapshot of our simulation from the point of view of our approach *MED*. The weights assigned by the method (orange bars) are calculated in a way that expected demand (red bars) is offset by available energy (blue bars), i.e., ambulances assigned to the respective base station. The energy distribution appears to be pretty spread out to minimize response times. Several base stations necessarily contain zero energy because, in this scenario, 25 ambulances have to cover all 45 base stations. However, the gaps are mostly in lower demand areas and can be covered by nearby base stations with assigned ambulances.

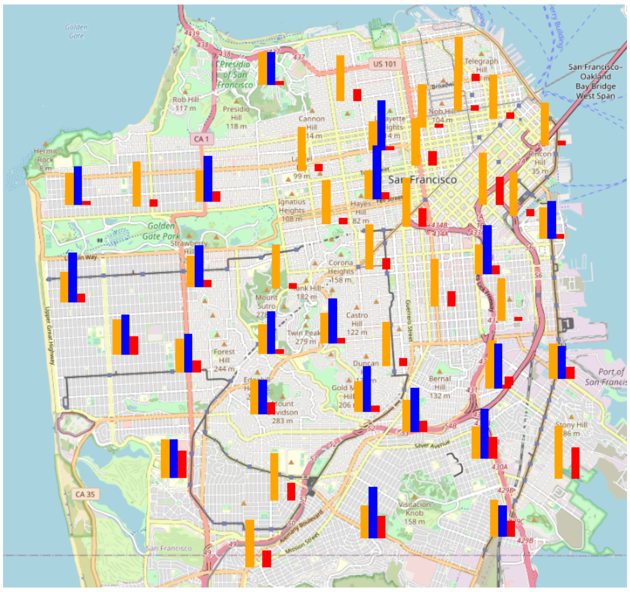


Figure 8: Snapshot of a simulation with 25 ambulances. Bar plots indicate base stations' estimated future energy values as calculated by MED: Supply (blue); Demand (red); Deficits (yellow) after nonlinear scaling (higher scores mean higher priority). Map data © OpenStreetMap contributors.

6 CONCLUSION

In this paper, we introduce the Dynamic Electric Ambulance Redeployment (DEAR) problem extending the Dynamic Ambulance Redeployment (DAR) problem to electric ambulances. We propose the Minimize Energy Deficits (MED) method, which determines redeployment actions by estimating the future energy deficit over all base stations. The energy deficit of a base station weighs a prediction of future demand against a prediction of the available energy level corresponding to the remaining range of stationed ambulances. We conducted experiments in a realistic scenario using an event-based simulator based on real-world incidents. Results show that MED reaches better performance than compared DAR methods, as well as baselines for EV settings. Furthermore, our results indicate that transitioning to electric ambulances can be done without increasing the number of available ambulances while maintaining comparable response times.

For future work, we plan to explore using more sophisticated prediction methods for demand and available energy. Furthermore, we want to examine sequential planning approaches considering multiple decisions in advance.

REFERENCES

[1] San Francisco Emergency Medical Services Agency. [n. d.]. *San Francisco EMS Ambulance Destinations.* http://sfemergencymedicalresponse.weebly.com/ambulance-destinations.html

[2] San Francisco Emergency Medical Services Agency. 2011. *Prehospital Provider Standards (Policy 4000).* https://www.sfdph.org/dph/files/EMS/Policy-Protocol-Manuals/Policy-Manual/1538-4000ProviderStandards09-01-2011.pdf

[3] Pieter L Van Den Berg and J Theresia Van Essen. 2019. Comparison of static ambulance location models. *International Journal of Logistics Systems and Management* 32, 3-4 (2019), 292–321.

[4] Lawrence H Brown, Christa L Whitney, Richard C Hunt, Michael Addario, and Troy Hogue. 2000. Do warning lights and sirens reduce ambulance response times? *Prehospital Emergency Care* 4, 1 (2000), 70–74.

[5] Andreas Bürger, Jan Wnent, Andreas Bohn, Tanja Jantzen, Sigrid Brenner, Rolf Lefering, Stephan Seewald, Jan-Thorsten Gräsner, and Matthias Fischer. 2018. The effect of ambulance response time on survival following out-of-hospital cardiac arrest: an analysis from the German resuscitation registry. *Deutsches Ärzteblatt International* 115, 33-34 (2018), 541.

[6] Nabil Channouf, Pierre L'Ecuyer, Armann Ingolfsson, and Athanassios N Avramidis. 2007. The application of forecasting techniques to modeling emergency medical system calls in Calgary, Alberta. *Health care management science* 10 (2007), 25–45.

[7] Richard Church and Charles ReVelle. 1974. The maximal covering location problem. In *Papers of the regional science association*, Vol. 32. Springer-Verlag Berlin/Heidelberg, 101–118.

[8] Mark S Daskin. 1983. A maximum expected covering location model: formulation, properties and heuristic solution. *Transportation science* 17, 1 (1983), 48–70.

[9] C Dugas, Y Bengio, F Bélisle, and C Nadeau. 2001. Incorporating second order functional knowledge into learning algorithms. *Advances in Neural Information Processing Systems* 13 (2001), 472–478.

[10] Shakiba Enayati, Maria E Mayorga, Hari K Rajagopalan, and Cem Saydam. 2018. Real-time ambulance redeployment approach to improve service coverage with fair and restricted workload for EMS providers. *Omega* 79 (2018), 67–80.

[11] Michel Gendreau, Gilbert Laporte, and Frédéric Semet. 2001. A dynamic model and parallel tabu search heuristic for real-time ambulance relocation. *Parallel computing* 27, 12 (2001), 1641–1653.

[12] Jeffrey Goldberg, Robert Dietrich, Jen Ming Chen, M George Mitwasi, Terry Valenzuela, and Elizabeth Criss. 1990. Validating and applying a model for locating emergency medical vehicles in Tuczon, AZ. *European Journal of Operational Research* 49, 3 (1990), 308–324.

[13] Caroline J Jagtenberg, Sandjai Bhulai, and Robert D van der Mei. 2015. An efficient heuristic for real-time ambulance redeployment. *Operations Research for Health Care* 4 (2015), 27–35.

[14] Shenggong Ji, Yu Zheng, Zhaoyuan Wang, and Tianrui Li. 2019. A deep reinforcement learning-enabled dynamic redeployment system for mobile ambulances. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 1 (2019), 1–20.

[15] Olive C Kobusingye, Adnan A Hyder, David Bishai, Manjul Joshipura, Eduardo Romero Hicks, and Charles Mock. 2006. Emergency medical services. *Disease Control Priorities in Developing Countries. 2nd edition* (2006).

[16] Matthew S Maxwell, Mateo Restrepo, Shane G Henderson, and Huseyin Topaloglu. 2010. Approximate dynamic programming for ambulance redeployment. *INFORMS Journal on Computing* 22, 2 (2010), 266–281.

[17] Laura A McLay and Maria E Mayorga. 2010. Evaluating emergency medical service performance measures. *Health care management science* 13 (2010), 124–136.

[18] Jill P Pell, Jane M Sirel, Andrew K Marsden, Ian Ford, and Stuart M Cobbe. 2001. Effect of reducing ambulance response times on deaths from out of hospital cardiac arrest: cohort study. *Bmj* 322, 7299 (2001), 1385–1388.

[19] Charles ReVelle and Kathleen Hogan. 1989. The maximum availability location problem. *Transportation science* 23, 3 (1989), 192–200.

[20] Emmanouil S Rigas, Antonis Billis, and Panagiotis D Bamidis. 2022. Can Artificial Intelligence Enable the Transition to Electric Ambulances? In *Challenges of Trustable AI and Added-Value on Health.* IOS Press, 73–77.

[21] Hubert Setzler, Cem Saydam, and Sungjune Park. 2009. EMS call volume predictions: A comparative study. *Computers & Operations Research* 36, 6 (2009), 1843–1851.

[22] Krisjanis Steins, Niki Matinrad, and Tobias Granberg. 2019. Forecasting the demand for emergency medical services. (2019).

[23] Niklas Strauß, Max Berrendorf, Tom Haider, and Matthias Schubert. 2022. A Comparison of Ambulance Redeployment Systems on Real-World Data. In *Proceedings of the 1st Workshop on Urban Internet-of-Things Intelligence (UNIT 2022) co-located with the 22nd IEEE International Conference on Data Mining (ICDM 2022).*

[24] Niklas Strauß, Lukas Rottkamp, Sebastian Schmoll, and Matthias Schubert. 2021. Efficient Parking Search using Shared Fleet Data. In *2021 22nd IEEE International Conference on Mobile Data Management (MDM).* IEEE, 115–120.

[25] Zhaonan Wang, Tianqi Xia, Renhe Jiang, Xin Liu, Kyoung-Sook Kim, Xuan Song, and Ryosuke Shibasaki. 2021. Forecasting ambulance demand with profiled human mobility via heterogeneous multi-graph neural networks. In *2021 IEEE 37th International Conference on Data Engineering (ICDE).* IEEE, 1751–1762.

[26] Zhengyi Zhou and David S Matteson. 2015. Predicting ambulance demand: A spatio-temporal kernel approach. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining.* 2297–2303.

Chapter 6

Efficient On-Street Parking Sensor Placement

Publication

Lukas Rottkamp, Matthias Schubert, and Niklas Strauß. "Efficient On-Street Parking Sensor Placement". In: *Proceedings of the 15th ACM SIGSPATIAL International Workshop on Computational Transportation Science*. IWCTS '22. Seattle, Washington: Association for Computing Machinery, 2022. ISBN: 9781450395397. DOI: `10.1145/3557991.3567796`

Contribution

Lukas Rottkamp is the first author of this publication. He compared and improved methods for placing a fixed number of sensors in order to maximize data value. He implemented an evaluation environment and several placement methods. He then ran evaluations using on-street parking data and compiled the manuscript. Matthias Schubert contributed to this publication through regular discussion of the ongoing work. Matthias Schubert further reviewed and edited the manuscript before submission.

Efficient On-Street Parking Sensor Placement

Lukas Rottkamp
LMU Munich
Munich, Germany
lukas.rottkamp@campus.lmu.de

Matthias Schubert
LMU Munich
Munich, Germany
schubert@dbs.ifi.lmu.de

Niklas Strauß
LMU Munich
Munich, Germany
strauss@dbs.ifi.lmu.de

ABSTRACT

When placing sensors in an environment, it may not be possible to directly cover all entities of interest with sensors due to cost or other restraints. This leads to a sensor placement problem in which only a subset of all sensible sensor locations is equipped with sensors. If data concerning the system to be measured is already available or easily procured, sensor locations can be selected in a data-driven approach. Without data, alternative methods have to be applied. In this paper, we present and compare various data-driven and data-agnostic methods for selecting parking sensor locations in a city environment. Experiments using real-world data show that methods only requiring parking bays' locations compare reasonable well to data-driven approaches requiring environment data which may be expensive to acquire.

CCS CONCEPTS

• **Applied computing** → **Transportation**; *Forecasting*; • **Information systems** → **Spatial-temporal systems**; • **Computing methodologies** → *Model development and analysis.*

KEYWORDS

sensor placement, smart city, parking

ACM Reference Format:
Lukas Rottkamp, Matthias Schubert, and Niklas Strauß. 2022. Efficient On-Street Parking Sensor Placement. In *The 15th ACM SIGSPATIAL International Workshop on Computational Transportation Science (IWCTS '22), November 1, 2022, Seattle, WA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3557991.3567796

1 INTRODUCTION

Accurate data is important when making statements regarding the current or future state of a system such as an urban environment. Such data can be automatically recorded by sensors and processed according to the needs of relevant use-cases. For example, the City of Melbourne, Australia, fitted individual parking bays in its Central Business District with in-ground occupancy sensors to obtain real-time parking occupancy information. This naturally comes with costs for planning, installation and maintenance. Secondary factors such as obtaining necessary permits, privacy concerns, property restrictions or similar issues may complicate the installation of a complete sensor network. Consequently, outfitting only a subset of possible sensor locations with sensors may be preferred or even required. This comes with the disadvantage of having blind spots which needs to be addressed in order to still be able to feed suitable data to smart city applications. Although methods for filling gaps exist, they come with an error due to the inherent uncertainty caused by the environment's dynamic behavior, e.g. visitors or commuters parking their cars. In order to enable a good service quality, data providers seek to minimize such errors.

Selecting a subset of possible sensors can be done in a data-driven way by obtaining data from a particular environment and then, using this data to calculate optimal sensor positions. In the parking use case, this could be done by conducting a study in which parking occupancy is recorded by human observers during a short time-frame. As this also comes with drawbacks such as personnel costs and an increased project duration and complexity, strategies not requiring such data may be beneficial if their results are not much worse than data-driven methods. It is important to note that such strategies do not have to work completely without data: Data regarding the environment is usually available, such as a map of the street network including parking bays.

In this paper, we review data-driven and data-agnostic methods relevant for placing sensors in order to interpolate parking occupancy data. We state methods not needing initial occupancy data but only a street network graph including parking bay locations. We then evaluate various methods using real-world parking data and conclude that our method is a viable alternative when obtaining initial occupancy data is not possible or expensive.

2 RELATED WORK

Using information obtained through spatial sensor networks is a common theme in smart city applications. For example, it was shown that information about the current occupancy state of parking bays can be used to guide drivers to a free parking bay faster than approaches without such data [15]. Future parking occupancy can be predicted given data of sensor networks [2]. Methods for monitoring parking availability include in-ground sensor networks [19] and analysis of camera footage [16]. A comprehensive overview of parking bay sensors and use-cases making use of parking data is given in [6]. In case not every parking opportunity is directly covered by sensors, interpolation methods can be used to estimate parking occupancy at unobserved locations [3].

Approaches using stationary sensors described above assume an already specified and fixed sensor network, either by outfitting each parking bay in the target area with a sensor or making use of an existing sensor infrastructure. In contrast, this paper is concerned with selecting a limited number of street segments for sensor placement.

IWCTS '22, November 1, 2022, Seattle, WA, USA

ACM ISBN 978-1-4503-9539-7/22/11…$15.00
https://doi.org/10.1145/3557991.3567796

Basic approaches for selecting a subset of possible sensor locations for sensor placement in a spatial environment include spreading the sensors randomly or uniformly over the area in question, e.g. by drawing a suitable grid and placing sensors near grid intersections. A more sophisticated method is to solve a coverage problem requiring that all areas or entities of interest are covered by at least one sensor, i.e., are within its detection range. This problem can be formalized as an optimization objective given coverage constraints and efficiently be solved using integer linear programming solvers. For example, [1] use a coverage approach to optimize placement of data relay nodes given a fixed set of parking sensors. Finding an exact solution may be computationally infeasible if the set of possible sensor combinations is large. In this case, heuristics such as simulated annealing or genetic algorithms can be applied [14].

Data-driven approaches need data of the type sensors would yield before the actual sensor placement is decided. These data may be collected by temporary sensors or a field study (e.g. personnel placed on streets with notepads). In some cases, it may be inferred from other data, although this adds another layer of uncertainty. Outfitting only a subset of possible sensor locations with sensors is known as the *sparse sensor placement optimization for reconstruction (SSPOR)* problem [11]. While work on SSPOR often uses spatial examples such as temperature interpolation, methods do not explicitly include spatial information. Instead sensors are placed only using sensor data. The PySensors toolkit for sensor placement presented by [4] contains algorithms to solve SSPOR: It takes a set of complete measurements and determines which components should be selected to reconstruct the remaining ones in an optimal way. The same problem is solved by Polire, an open source toolkit for spatial interpolation and sensor placement [9, 13]. It applies a greedy algorithm which selects sensors according to gains in a specified criterion, such as mutual information or entropy. The Chama framework for sensor placement covers many relevant algorithms and strategies for sensor placement but focuses on global event detection such as detecting an earthquake or a pollutant leaking into a system [7]. We on the other hand are not interested in detecting global events but detecting a multitude of individual parking events. [8] determine how a fixed budget may be best spent on individual observations when each observation is connected with a certain cost. They do not determine a fixed subset of sensors but determine which locations should best be queried at which time. In contrast, we are interested in determining a permanent network of stationary sensors.

3 PROBLEM DEFINITION

A spatial sensor selection problem includes entities of which state information should be collected by sensors. Multiple problem definitions are possible depending on the nature of this entity. In our definition, we consider n discrete entities p_i, e.g. each being the set of individual parking bays of a given street segment. This covers the use-case of recording on-street parking availability with in-ground devices as the infrastructure needed for a street segment (such as a relay node processing raw data of individual sensors and transmitting occupancy data to a central database) can be use by multiple sensors connected by wires or near-field communication [1]. In this case, we call the combined set of devices monitoring exactly one street segment a sensor. It also covers the use-case of recording such data by analyzing camera images, as a camera can typically be installed in a way that covers all parking bays of a street segment, but not multiple street segments at once.

Note that in our definition, a street segment never contains an intersection. Further, the link between two intersections is divided into multiple street segments if it would otherwise contain too many parking bays. We call a street segment connected to an entity a *candidate segment* as it is a candidate for sensor placement. Sensors are never placed on street segments that are not candidate segments.

Each entity is connected to a measurable value $v_{i,t}$ ($0 \leq v_{i,t} \leq 1$) for each time t, e.g. the fraction of occupied parking bays at time t. The spatial relation of p_i is given as their location in a graph G with nodes N and edges E. Each p_i is connected to an edge $e_i \in E$, e.g. a street segment. Each node carries location information in form of Cartesian coordinates. Therefore, each entity p_i can be assigned a location l_i that is defined as the midpoint between both nodes of its edge e_i.

As stated above, each entity p_i is monitored by exactly one sensor. Sensor presence is indicated by an indicator variable $s_i \in \{0, 1\}$ which is set to 1 if a sensor is present at p_i, otherwise 0. Each of m sensors placed covers exactly one entity, i.e. $\sum_{1 \leq i \leq n} s_i = m$.

The sensor selection problem now selects the set of entities $\hat{\sigma}$ to be equipped with a sensor that maximizes an objective function λ over all valid sensor subsets $\sigma_k \in \Omega$:

$$\hat{\sigma} = \underset{\sigma_k \in \Omega}{\operatorname{argmax}}\, \lambda(\sigma_k) \tag{1}$$

with

$$\Omega = \{(s_1, ..., s_n) \forall s_i \in \{0, 1\}\}, \sum_{1 \leq i \leq n} s_i = m \tag{2}$$

Reasonable functions for λ include averaged interpolation or prediction errors when using the subset to reconstruct actual values $v_{i,t}$. Assuming a predictor $\Gamma(\sigma, i)$ taking a subset of sensors to predict the value of the i-th sensor, the mean absolute error (MAE) may be used:

$$\lambda_{\mathrm{MAE}}(\sigma) = \sum_{0 \leq i \leq n, \forall t} |v_{i,t} - \Gamma(\sigma, i)| \tag{3}$$

Note that the best subset depends on both predictor Γ and objective function λ. These have to be chosen according to the use-case.

4 SENSOR PLACEMENT METHODS

Various methods for sensor placement are mentioned in Section 2 above. These include data-driven and data-agnostic approaches. Data-driven approaches use observation data for placement of sensors. In our parking use-case, this is parking occupancy data. Data-agnostic methods don't use such data but may use *metadata* such as locations of candidate segments and the street graph. Some methods are deterministic while others involve random components such as random initialization or random tie-breaks. Each method is given a input parameter m denoting the exact size of the target subset of candidate segments to select for sensor placement, i.e. the number of sensors to place.

We now describe a number of methods which are relevant for our parking use-case and included in our evaluation.

4.1 Data-agnostic placement methods

Simple placement methods *Random* and *Largest* are included as benchmark methods for comparison. We devised methods *Clusters* and *MaxMin* to exploit the observation that pairwise correlation between spatial resources depends on the distance between them: A smaller distance tends to coincide with higher similarity. This may be due to their shared neighborhood with points of interest targeted by drivers. Other reasons may include differing parking rules or peculiarities of the street network such as dead-end streets or especially busy areas. While we are certainly not the first to use the underlying algorithms, we are not aware of other attempts of using them in related problem settings. Method *Coverage* is included for comparison as coverage-based methods are routinely used for sensor placement [1, 18]. Note that we don't use it in the "traditional" way of optimizing *direct* sensor coverage as explained below.

4.1.1 Random. Candidate segments are selected for sensor placement by random draw. Each segment has the same probability of being drawn.

4.1.2 Largest. Candidate segments are selected only by their respective number of individual parking bays contained. Segments with higher counts are selected first. This method is explicitly included as a "higher-bound" benchmark as we have no reason to believe that it leads to advantageous selections.

4.1.3 Clusters. This method is shown in Algorithm 1: To select n sensors, segments are first clustered into n clusters through K-Means clustering using Lloyd's algorithm [10]. Each cluster's center point is calculated as the mean location of all candidate segments it contains. The nearest not previously selected candidate segment to each center point is determined and selected for sensor placement. Note that the algorithm uses locations given in a local Cartesian coordinate system. Locations denoted in geographic coordinates are projected to a suitable local Cartesian coordinate system first. The method is not deterministic as results of Lloyd's algorithm depend on its random initialization and ties in distance are resolved randomly.

4.1.4 MaxMin. This method selects candidate segments so that the minimum pairwise graph distance over all selected sensors is maximized. This effectively spreads the sensors as widely as possible while preventing sensors to be near to each other. The large number of possible subsets prevents us from obtaining an optimal solution due to the high computational complexity. Instead, we use a greedy heuristic shown in Algorithm 2. We restart this algorithms multiple times, always keeping the best result seen so far, to minimize the risk of ending up in local optimum worse than the global optimum.

4.1.5 Coverage. This method also exploits the spatial relationship of parking segments. Here, a candidate segment is defined to be covered if at least one sensor is present within a certain graph distance d. Note that "coverage" in this sense does not refer to direct coverage through actual observance by the sensor but indirect coverage due to a statistically higher likeness of occupancy because of spatial closeness. The method places m sensors so that d is minimized under the constraint that each candidate segment is covered.

```
Data: P                         ▷ Set of all possible sensor locations
Data: k ≥ 1                     ▷ Amount of sensors to select
clusterCenters ← clusterKMeans(P, k);
bestSubset ← {};
foreach center ∈ clusterCenters do
    minDist ← inf;
    foreach c ∈ P \ bestSubset do
        d ← dist(center, c);
        if d < minDist then
            minDist ← d;
            bestCandidate ← c;
        end
    end
    bestSubset ← bestSubset ∪ {bestCandidate};
end
return bestSubset;
```

Algorithm 1: Part of selection method Clusters.

```
Data: P                         ▷ Set of all possible sensor locations
Data: k ≥ 1                     ▷ Amount of sensors to select
bestSubset ← pickOneRandomly(subsetsOfSize(P, k));
maxMinDist ← minPairwiseDist(bestSubset);
repeat
    improved ← False;
    S ← bestSubset;
    foreach T ∈ subsetsOfSize(S, k − 1) do
        foreach c ∈ P \ S do
            U ← T ∪ {c};
            d ← minPairwiseDist(U);
            if d > maxMinDist then
                maxMinDist ← d;
                bestSubset ← U;
                improved ← True;
            end
        end
    end
until not improved;
return bestSubset;
```

Algorithm 2: Part of selection method MaxMin.

This can efficiently be done by pre-computing solutions to the the set-cover problem [12] for a each distance d in a suitable range. Each solution of the set-cover problem is a minimal set of sensors so that each candidate segment is covered assuming a sensor range of d. An optimal set of m sensors can then be looked up in the pre-computed solution list by selecting the solution with lowest distance under the condition that m sensors are selected.

Minimal set sizes for our evaluation environment Melbourne are shown in Figure 1. For example, the installation of 50 sensors can be done in a way so that no parking segment is more than about 300 meters away from a sensor.

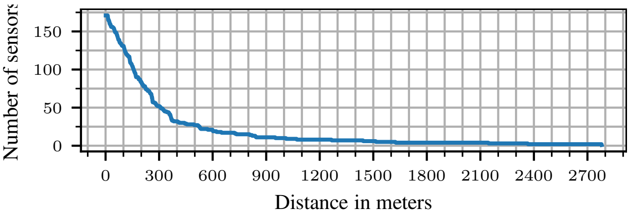


Figure 1: Minimum amount of sensors needed to cover all parking segments for various coverage distances in our evaluation environment.

4.2 Data-driven placement methods

Data-driven approaches are given observation data of all candidate segments. This means they can effectively select the best sensor sets in hindsight. They are connected to higher costs in real-world applications in case such data needs to be obtained first. The following methods are included:

4.2.1 MinError. This method places sensors in a way that minimizes the evaluation metric given occupancy data. Because the computational complexity prevents us from finding the optimal solution for larger experiments such as ours, we implemented a greedy search heuristic: Starting with an empty set, sensors are added one-by-one until the targeted number of sensors is reached. The sensors added in each step is the one which, compared to all other remaining candidates, leads to the smallest total error. While this heuristic only approximates the optimal selection, it produces results very close to the optimal solution in smaller experiments and we are confident it is a good benchmark method.

4.2.2 PySensors. The PySensors [4] framework is included using the original implementation[1]. It includes three choices of basis functions, *Identity*, *SVD* and *Random*, each requiring parameters. For our evaluation, we determined the best basis and parameters experimentally in preliminary experiments. It should be noted that PySensors makes no use of segments' locations nor the street graph but is a pure data-driven selection method working on occupancy data.

4.2.3 Polire. Like *PySensors*, the Polire framework [13] does not consider location information but only the supplied occupancy time series. We include it in our evaluation via the original implementation[2]. Users can choose between stationary or non-stationary models, different kernels and parameters. Those were again determined beforehand in preliminary experiments.

5 INTERPOLATION METHODS

Having selected a subset of segments for sensor placement, predictor Γ (see Section 3) infers the states of remaining parking segments from data obtained through those sensors. We use spatial interpolation techniques for this task as parking bays are spatially related. Two interpolation methods are used in our evaluation:

[1]https://github.com/dynamicslab/pysensors
[2]https://github.com/sustainability-lab/polire/blob/SenSys20_Poster/polire/placement/base/base.py

The **KNN** (k nearest neighbors) method as shown in Equation 4 is employed to calculate parking availability A at location x given the set R_k of the k nearest parking bays and their respective availability values A_i measured by sensors. In our evaluation, we set the parameter k to 5 as this value gave lowest interpolation errors in preliminary experiments.

$$A(x) = \frac{\sum_{i \in R_k} A_i}{k} \tag{4}$$

We also include the **IDW** (inverse distance weighting) method [17] shown in Equation 5. Here, availability A at location x is calculated by a weighted average over all available sensor values A_i. The weights $w_i(x)$ depend on the distance between location and sensor raised to the power of α. We set α to 2 as preliminary experiments gave good interpolation performance using this value.

$$A(x) = \frac{\sum_{i \in R} w_i(x) A_i}{\sum_{i \in R} w_i(x)}, w_i(x) = \frac{1}{d(x, x_i)^{\alpha}} \tag{5}$$

According to [3], IDW is well-suited for estimating parking availability given a limited number of sensors. This method seems especially beneficial as the information given by a sensor in spatial settings like ours is expected to statistically decrease with distance.

6 EVALUATION

Evaluation follows the two-step process described in Section 3. First, a subset of parking segments is selected according to the respective sensor placement method. Occupancy values are then "virtually" measured by those sensors according to ground truth data. These values are then used to obtain all remaining parking segments' occupancy values using an interpolation method. Finally, the interpolation error is calculated using ground truth data. Non-deterministic methods were evaluated multiple (7) times to reduce influence of random effects.

Two experiments have been conducted to gain insights into the performance of methods: The first scenario assumes complete availability of observation data. This enables us to evaluate models' best-case performance and gain insights in the general task of interpolating parking occupancy. A low error in this scenario is however not alone representative of a method's real-world capability as there is no need to place sensors if all data is be available anyway. The interpolation error on previously unseen data is also a crucial metric. Therefore, we include a second scenario which only presents a fraction of data to selection algorithms. In practice, this data may be acquired through a field study. Interpolation errors are then calculated over a test set consisting of a time span not included in the initial training data.

6.1 Evaluation dataset

Real-world parking occupancy ground truth data for our evaluation is taken from the City of Melbourne, Australia, open data platform[3]. This platform provides the dataset *On-street Parking Bays*[4] containing locations of on-street parking bays in the Central Business

[3]City of Melbourne open data platform: https://data.melbourne.vic.gov.au; Data licensed under *Creative Commons Attribution 3.0 Australia*: https://creativecommons.org/licenses/by/3.0/au/
[4]https://data.melbourne.vic.gov.au/Transport-Movement/On-street-Parking-Bays/crvt-b4kt

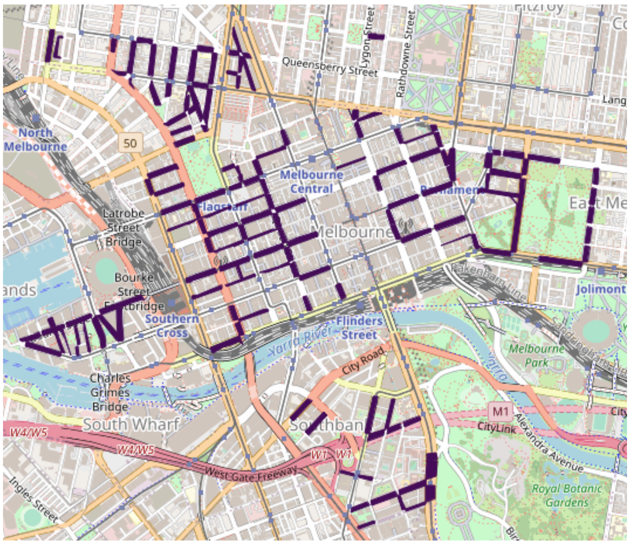


Figure 2: On-Street parking segments used for evaluation.

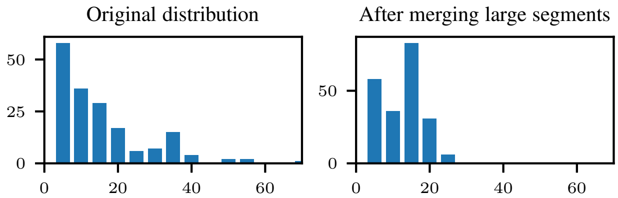


Figure 3: Histogram of parking segment sizes before and after processing.

District of Melbourne. We connected these parking bays to a street graph obtained through OpenStreetMap[5]. Parking bays for loading only and bays requiring special permissions were removed.

Individual on-street parking occupancy data of the same parking bays is available in dataset *On-street Car Parking Sensor Data - 2017*[6]. In this dataset, arrivals and departures of vehicles are recorded to the second. We resampled the occupancy into 5-minute time slots, each of which represents a state of the overall parking situation. Individual parking bays were then aggregated according to their street segments, a street segment being defined a segment of a street between two intersections. Segments with more than 30 parking bays were split into multiple smaller ones then amounting to sizes of between 15 and 20 bays. This was done as exceptionally large segments (usually caused by a street with few intersections) may not be covered by a single sensor such as a camera sensor. The resulting parking segment size histogram is presented in Figure 3. After processing, we obtain 189 parking street segments.

The following datasets were included for evaluation:

[5]https://www.openstreetmap.org
[6]https://data.melbourne.vic.gov.au/Transport/On-street-Car-Parking-Sensor-Data-2017/u9sa-j86i

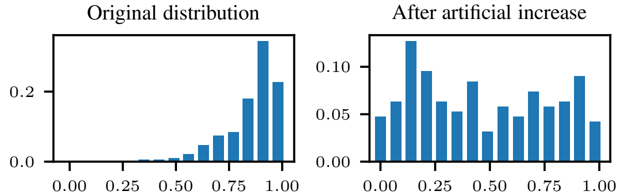


Figure 4: Distribution of parking bays having at least one free bay, before and after artificial increase.

6.1.1 Regression. This dataset contains each parking street segment's occupancy for each 5-minute time slot. These values are calculated as average occupancy values over all individual parking bays of respective segments, giving a numeric target variable between 0 and 1.

6.1.2 ThreeBins. This dataset represents a classification problem as numeric values of dataset *Regression* are sorted into the three bins *low occupancy*, *medium occupancy* and *high occupancy*. It is motivated by existing parking occupancy tools indicating estimated availability using traffic light colors. This is easier to communicate to end users than numeric values. Further, numeric values may suggest a sense of accuracy that may be difficult to achieve when predicting parking accuracy. On the other hand, a more coarse classification such as this one is satisfactory for a user looking for parking opportunities.

6.1.3 OneFreeHigh. In this dataset, the target variable is binary and states if at least one individual parking bay of the parking street segment in question is free at a given point in time. This is motivated by the fact that in practice, drivers searching for a parking opportunity in their immediate surroundings are primarily interested in streets containing at least one free parking bay. It should be noted that according to the Melbourne in-ground sensor data, most parking segments contain at least one free parking bay at a given time. Naturally, a parking information system is most appreciated when finding a free parking opportunity presents a challenge to motorists, i.e. when parking opportunities are rare. To evaluate the various methods in such a setting, we artificially increased the parking occupancy of all parking bays so that only about every second parking segment contains at least one free parking bay. This means an increase of parking demand by 38%. A comparison of the resulting increased occupancy distribution versus original occupancy distribution is shown in Figure 4.

6.2 Experiment 1: Training on complete data

Experiment 1 covers six months, from June to November 2017. Data-driven methods will receive the complete ground-truth occupancy data. As described above, this is unrealistic in practice but yields insights about the best-case performance of placement methods. No-data methods make no use of occupancy data.

Each selection method is executed multiple times: The respective number of sensors to select increases from only one sensor to 171 sensors (out of 189 possible sensors). Interpolation errors for each method and parameter are then calculated as described above.

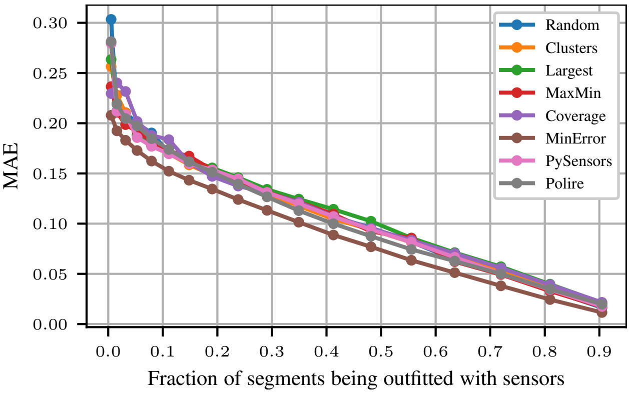
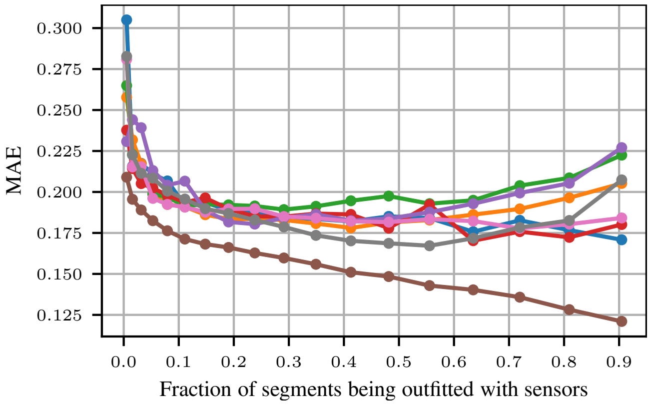


Figure 5: Mean absolute interpolation errors (over all parking segments) by size of sensor subset on *Regression* dataset. Each line represents a selection method.

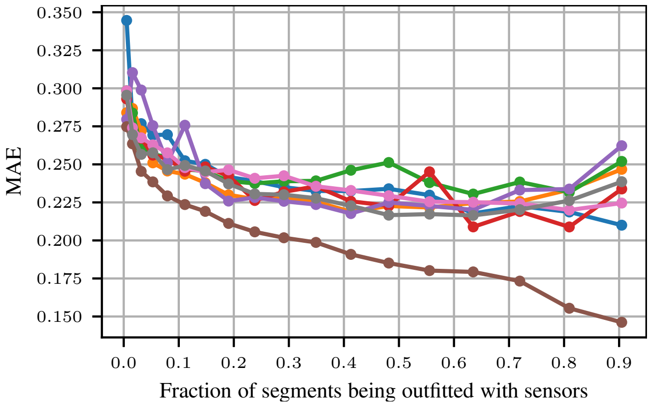
6.2.1 Comparing results for different methods. The mean absolute IDW interpolation errors of evaluated methods given varying numbers of segments to select are shown in Figure 5. For non-deterministic methods, multiple repetitions were run whose errors are averaged. Not surprisingly, errors decrease with growing amount of sensors. Our "lower-bound" benchmark method *MinError* clearly produces the smallest error as expected given its direct optimization of the IDW error. Other methods are not easily distinguished in this overview.

For a better evaluation, it is appropriate to exclude candidate segments fitted with sensors from evaluation, as these naturally show no error. We will focus on these error values in the remainder. The are shown in Figure 6a for the *Regression* dataset. Our earlier observations appear in more detail. Method *Polire* shows second-best performance while *Random* appears to be average. Errors for datasets *ThreeBins* and *OneFreeHigh* are shown in Figure 6a and Figure 6c, respectively. The ranking of methods is similar to the one observed for the *Regression* dataset. A notable exception is the error curve of method *Clusters* when evaluated using dataset *OneFreeHigh*: It is now intersecting with *Polire* multiple times. This is especially remarkable given their different nature, as *Clusters* does not need occupancy data but only parking bays' locations.

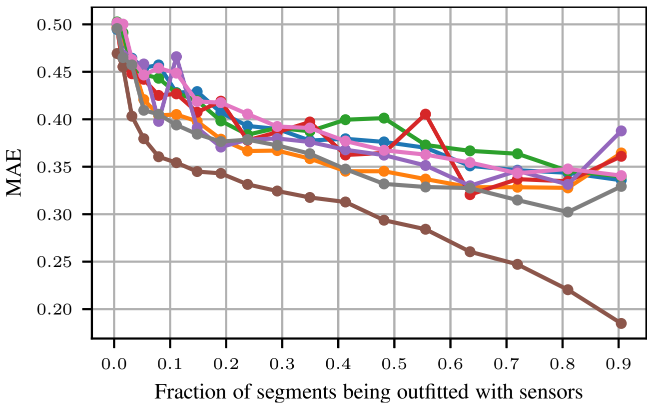
A condensed comparison of errors is shown in Table 1. Relative errors compared to *MinError*, the best method for each dataset, are included. *MinError* being the overall best method is not surprising as it by design directly optimizes the evaluation metric. Second-best is *Polire*, also requiring complete data. Method *Clusters* reaches third place even though it doesn't use occupancy data. Sensors selected by method *PySensors* produces slightly higher errors than some data-agnostic methods. It can be seen that methods' ranks are often equal for both interpolation methods. Selection method *Random* ranks in the bottom half of results. However, error differences are relatively small in some cases. We attribute this to the fact that random placement tends to cover the whole area which is beneficial for interpolation as spatial resources such as parking bays are typically locally correlated. Further, if random placement considers not just an area but the set of possible sensor locations as in the setting of this paper, it implicitly draws from a density distribution of entities to be measured, potentially reducing the interpolation error even more.



(a) Error using *Regression* dataset.



(b) Error using *ThreeBins* dataset.



(c) Error using *OneFreeHigh* dataset.

Figure 6: Mean absolute interpolation errors (only unselected parking segments) by size of sensor subset. Each line represents a selection method (color definition in Figure 5).

Method	Regression	ThreeBins	OneFreeHigh	Mean
MinError	.161 (+0%)	.207 (+0%)	.327 (+0%)	+0.0%
Polire	.193 (+19%)	.239 (+15%)	.377 (+15%)	+16.9%
Clusters*	.196 (+21%)	.240 (+16%)	.384 (+17%)	+18.4%
MaxMin*	.192 (+18%)	.241 (+16%)	.398 (+21%)	+19.0%
PySensors	.194 (+20%)	.244 (+18%)	.407 (+24%)	+21.1%
Random*	.197 (+21%)	.247 (+19%)	.404 (+23%)	+21.8%
Coverage*	.202 (+25%)	.247 (+19%)	.396 (+20%)	+21.9%
Largest*	.203 (+25%)	.249 (+20%)	.408 (+24%)	+23.7%

Table 1: Mean IDW interpolation MAE values of selection methods over all subset sizes. Relative difference to best in column is shown in brackets, with mean in right column. Methods not using parking data are marked with asterisks.

Method	Regression	ThreeBins	OneFreeHigh	Mean
MinError	.174 (+0%)	.226 (+0%)	.384 (+0%)	+0.0%
Polire	.198 (+13%)	.246 (+8%)	.404 (+5%)	+9.3%
Clusters*	.194 (+11%)	.246 (+8%)	.414 (+7%)	+9.3%
Coverage*	.194 (+11%)	.244 (+7%)	.421 (+9%)	+9.6%
MaxMin*	.194 (+11%)	.247 (+9%)	.424 (+10%)	+10.2%
PySensors	.197 (+13%)	.251 (+10%)	.429 (+11%)	+11.8%
Random*	.199 (+14%)	.252 (+11%)	.428 (+11%)	+12.3%
Largest*	.205 (+17%)	.254 (+12%)	.429 (+11%)	+13.8%

Table 2: Mean KNN interpolation MAE values of selection methods over all subset sizes. Relative difference to best in column is shown in brackets, with mean in right column. Methods not using parking data are marked with asterisks.

Method	Regression	ThreeBins	OneFreeHigh
IDW	.196	.244	.398
KNN	.197	.249	.423

Table 3: Mean interpolation errors of interpolation methods IDW and KNN for evaluated datasets.

6.2.2 Comparison of interpolation methods. Table 2 shows results for KNN interpolation errors in contrast to the IDW interpolation errors discussed above. Method *MinError*'s error is much higher than before, lowering the relative difference to other methods. This is expected as *MinError* directly minimizes the IDW error but now interpolation uses the KNN method. The ranking of methods is almost the same as before, suggesting that IDW and KNN produce similar estimates.

Aggregated values for interpolation methods are shown in Table 3. Note that these exclude method *MinError* due to its minimization of IDW error as this would skew results towards IDW. Still it can be seen that IDW interpolation yields slightly better overall results. This is not surprising as IDW weights nearer sensor values higher than those farther away which exploits the environment's spatial correlation motivated above. Still differences are very small.

Method	Regression	ThreeBins	OneFreeHigh
Random	.0093	.0102	.0101
Clusters	.0015	.0021	.0041
PySensors	.0041	.0047	.0101

Table 4: Mean standard deviation of error for evaluated methods and datasets. For each evaluated subset size, the standard deviation was calculated over all repetitions. Averages over all subset sizes are shown here.

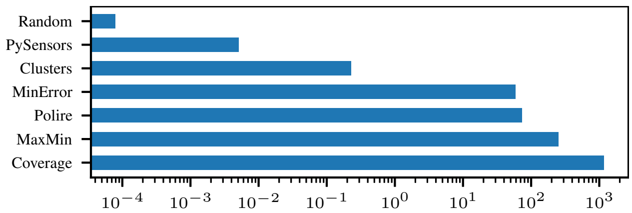


Figure 7: Mean CPU time (in seconds) per selection run during Experiment 1.

6.2.3 Variance analysis. Methods *Random*, *Clusters* and *PySensors* are non-deterministic as they contain random choices at some point. To average out random effects, evaluation results discussed above have been averaged over a number of runs. It is important to also analyze the errors' variance as a high variance method may lead to disappointing results when applied in a real-world setting without ground-truth data. Consistently high variance would also question the reliability of the evaluation procedure itself. The mean standard deviation of individual methods is shown in Table 4: Results are low compared to the absolute errors in Table 1. Values for method *Clusters* are significantly smaller than those of method *Random*, indicating a stable clustering. The error variability of method *PySensors* is caused by its use of randomized SVD solver [5]. Its magnitude is again small compared to absolute error values. Overall, the low variance increases confidence in the applicability of evaluated methods.

6.2.4 Comparison of selection time. Figure 7 shows the CPU time needed per selection run during evaluation using an Intel® Core™ i7-10750H CPU. This comparison should be taken with a grain of salt as implementations of methods are not primarily optimized for best performance. Still the range over multiple orders of magnitude is something to consider when applying methods in larger environments than ours. Generally, the runtime of selection methods is of lesser importance as they are only run during the planning phase before sensors are being installed. For interpolation between measurements of installed sensors, fast implementations of IDW and KNN methods exist.

6.3 Experiment 2: Limited training set

This experiment covers the scenario of conducting a field study to obtain data for data-driven placement methods. As Melbourne ground truth data is readily available in the datasets described above, we conduct a "virtual" field study by extracting a time span

Method	Regression	ThreeBins	OneFreeHigh	Mean
MinError	.165 (+0%)	.213 (+0%)	.333 (+0%)	+0.0%
Polire	.191 (+15%)	.234 (+9%)	.379 (+13%)	+13.1%
Clusters*	.195 (+18%)	.239 (+12%)	.385 (+15%)	+15.4%
MaxMin*	.190 (+15%)	.240 (+12%)	.399 (+19%)	+16.1%
PySensors	.194 (+17%)	.242 (+13%)	.407 (+22%)	+17.9%
Random*	.195 (+18%)	.246 (+15%)	.405 (+21%)	+18.4%
Coverage*	.201 (+21%)	.246 (+15%)	.397 (+19%)	+18.8%
Largest*	.201 (+22%)	.249 (+16%)	.409 (+22%)	+20.6%

Table 5: Experiment 2: Mean IDW interpolation MAE values of selection methods over all subset sizes. Relative difference to best in column is shown in brackets, with mean in right column. Methods not using parking data are marked with asterisks.

Method	Regression	ThreeBins	OneFreeHigh
MinError	3.240%	7.092%	4.731%
Polire	0.623%	0.996%	0.034%
PySensors	0.097%	1.210%	0.886%

Table 6: Mean IDW-interpolation MAE increases from training to testing error of selection methods over all subset sizes.

of training data from these datasets. We chose a span of four consecutive weeks for training as shorter spans resulted in large training set error variability due to the limited number of samples. A second extract of the following four months is then used as test data on which sensor selections are evaluated.

6.3.1 Comparing results for different methods. A comparison of test errors for evaluated placement methods can be seen in Table 5. They closely resemble the results of Experiment 1 shown in Table 1. This is no surprise for data-agnostic methods as their selections do not depend on the training data and all test data of Experiment 2 is also included in Experiment 1. Data-driven methods on the other hand are now working on substantially reduced training data (four weeks instead of six month during Experiment 1) which however did not effect their performance. This indicates that four weeks of training data are sufficient for data-driven selection methods.

6.3.2 Comparing training and testing errors. Table 6 shows relative increases of errors when comparing test dataset errors with training errors. Method *MinMax* shows largest increases, probably due to overfitting as it directly optimizes the error metric during training. Methods *Polire* and *PySensors* show smaller deviations. This may indicate that they internally created more robust models which generalize better than the aggressive method of *MinMax*. Generally, the moderate increase indicates that good selections during training are still good during later time spans. This is an important insight as it confirms our initial proposition that placing sensors at only a subset of parking street segments is a viable strategy.

7 CONCLUSION

In this paper, we described and compared various sensor placement methods. Our evaluation using real-world parking data shows that data-driven placement methods lead to slightly lower interpolation errors than data-agnostic methods not receiving such data but only metadata such as locations of on-street parking bays. Data-driven methods however require data typically obtained through preliminary surveys or installation of temporary sensors which may be expensive and time-consuming.

We conclude that data-agnostic methods are a reasonable alternative if suitable data is not readily available. Especially our proposed cluster-based method appears to be a good choice in such cases.

REFERENCES

[1] Antoine Bagula, Lorenzo Castelli, and Marco Zennaro. 2015. On the design of smart parking networks in the smart cities: An optimal sensor placement model. *Sensors* 15, 7 (2015), 15443–15467.

[2] Fabian Bock, Sergio Di Martino, and Antonio Origlia. 2017. A 2-step approach to improve data-driven parking availability predictions. In *Proceedings of the 10th ACM SIGSPATIAL workshop on computational transportation science.* 13–18.

[3] Fabian Bock and Monika Sester. 2016. Improving parking availability maps using information from nearby roads. *Transportation Research Procedia* 19 (2016), 207–214.

[4] Brian M de Silva, Krithika Manohar, Emily Clark, Bingni W Brunton, Steven L Brunton, and J Nathan Kutz. 2021. PySensors: A Python package for sparse sensor placement. *arXiv preprint arXiv:2102.13476* (2021).

[5] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. 2009. Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions. (2009).

[6] MY Idna Idris, YY Leng, EM Tamil, NM Noor, Z Razak, et al. 2009. Car park system: A review of smart parking system and its technology. *Information Technology Journal* 8, 2 (2009), 101–113.

[7] Katherine A Klise, Bethany L Nicholson, and Carl Damon Laird. 2017. *Sensor placement optimization using Chama.* Technical Report. Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).

[8] Andreas Krause, Eric Horvitz, Aman Kansal, and Feng Zhao. 2008. Toward community sensing. In *2008 International Conference on Information Processing in Sensor Networks (ipsn 2008).* IEEE, 481–492.

[9] Andreas Krause, Ajit Singh, and Carlos Guestrin. 2008. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research* 9, 2 (2008).

[10] Stuart Lloyd. 1982. Least squares quantization in PCM. *IEEE transactions on information theory* 28, 2 (1982), 129–137.

[11] Krithika Manohar, Bingni W Brunton, J Nathan Kutz, and Steven L Brunton. 2018. Data-driven sparse sensor placement for reconstruction: Demonstrating the benefits of exploiting known patterns. *IEEE Control Systems Magazine* 38, 3 (2018), 63–86.

[12] Seapahn Meguerdichian and Miodrag Potkonjak. 2003. *Low power 0/1 coverage and scheduling techniques in sensor networks.* Technical Report. Citeseer.

[13] S Deepak Narayanan, Zeel B Patel, Apoorv Agnihotri, and Nipun Batra. 2020. A toolkit for spatial interpolation and sensor placement. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems.* 653–654.

[14] Sharon L Padula and Rex K Kincaid. 1999. *Optimization strategies for sensor and actuator placement.* Technical Report.

[15] Lukas Rottkamp and Matthias Schubert. 2020. Quantifying the potential of data-driven mobility support systems. In *Proceedings of the 13th ACM SIGSPATIAL International Workshop on Computational Transportation Science.* 1–10.

[16] Xavier Sevillano, Elena Màrmol, and Virginia Fernandez-Arguedas. 2014. Towards smart traffic management systems: Vacant on-street parking spot detection based on video analytics. In *17th International Conference on Information Fusion (FUSION).* IEEE, 1–8.

[17] Donald Shepard. 1968. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference.* 517–524.

[18] Chenxi Sun, Victor OK Li, Jacqueline CK Lam, and Ian Leslie. 2019. Optimal citizen-centric sensor placement for air quality monitoring: a case study of city of Cambridge, the United Kingdom. *IEEE Access* 7 (2019), 47390–47400.

[19] Carol Zimmerman, Rachel Klein, Jeremy Schroeder, Katie Turnbull, Kevin Balke, Mark Burris, Emily Saunoi-Sandgren, Elliot Martin, Susan Shaheen, Caroline Rodier, et al. 2014. *San Francisco urban partnership agreement: national evaluation report.* Technical Report. United States. Department of Transportation. Intelligent Transportation Systems Joint Program Office.

Chapter 7

A Time-Inhomogeneous Markov Model for Resource Availability under Sparse Observations

Publication

Lukas Rottkamp and Matthias Schubert. “A Time-Inhomogeneous Markov Model for Resource Availability under Sparse Observations”. In: *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. SIGSPATIAL ’18. Seattle, Washington: Association for Computing Machinery, 2018, pp. 460–463. ISBN: 9781450358897. DOI: 10.1145/3274895.3274945

Extended Version

A long version of the above publication is published on arXiv:

Lukas Rottkamp and Matthias Schubert. “A Time-Inhomogeneous Markov Model for Resource Availability under Sparse Observations”. In: *arXiv* (2024). DOI: 10.48550/arXiv.2404.12240. URL: https://doi.org/10.48550/arXiv.2404.12240

Contribution

Lukas Rottkamp is the first author of this publication. He proposed a novel Cyclic Markov Model to model resource states given sparse measurements. He devised corresponding parameter fitting methods for complete and sparse data. He then implemented the methods and ran experiments with parking availability data. Finally, he produced the manuscript. Matthias Schubert contributed to this publication through regular discussion of the ongoing work especially regarding suitable experiments for evaluation and the structure of the manuscript. Matthias Schubert reviewed drafts of the manuscript and helped polishing it before it was submitted to the venue.

A Time-Inhomogeneous Markov Model for Resource Availability under sparse Observations

Lukas Rottkamp
Audi Electronics Venture GmbH
Gaimersheim, Germany
lukas.rottkamp@audi.de

Matthias Schubert
LMU Munich
Munich, Germany
schubert@dbs.ifi.lmu.de

ABSTRACT

Accurate spatio-temporal information is crucial for smart city applications such as modern routing algorithms. Often, this information describes the state of stationary resources, e.g. the availability of parking bays, charging stations or the amount of people waiting for a vehicle to pick them up near a given location. Predicting future states of the monitored resources is often mandatory because a resource might change its state within the time until it is needed. It is often not possible to obtain complete history of a resource's state. For example, the information might be collected from traveling agents visiting the resource with an irregular frequency. Thus, it is necessary to develop methods which work on sparse observations for training and prediction. In this paper, we propose time-inhomogeneous discrete Markov models to allow accurate prediction even when the frequency of observation is very rare. Our new model is able to blend recent observations with historic data and also provide useful probabilistic estimates for future states. Since resource availability in a city is typically time-dependent, our Markov model is time-inhomogeneous and cyclic within a predefined time interval. We propose a modified Baum-Welch algorithm capable of training our model with sparse data. Evaluations on real-world datasets of parking bay availability show that our new method indeed yields good results compared to methods designed for training on complete data and non-cyclic variants.

CCS CONCEPTS

• **Computing methodologies → Modeling methodologies**; • **Information systems → Geographic information systems**; *Sensor networks*; • **Mathematics of computing →** *Kalman filters and hidden Markov models*;

KEYWORDS

Smart City data, spatial resources, predictive models

ACM Reference Format:

Lukas Rottkamp and Matthias Schubert. 2018. A Time-Inhomogeneous Markov Model for Resource Availability under sparse Observations. In *26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '18), November 6–9, 2018, Seattle, WA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3274895.3274945

SIGSPATIAL '18, November 6–9, 2018, Seattle, WA, USA

ACM ISBN 978-1-4503-5889-7/18/11.
https://doi.org/10.1145/3274895.3274945

1 INTRODUCTION

Knowledge of a city's resources (e.g. parking bays, charging stations, stations for rental bikes) and their current availability state is becoming more and more important. For instance, modern routing algorithms integrate this information to minimize travel times[1, 5, 13] and augmented reality applications use such data to assist users[7, 10]. Additionally, it is often required to estimate future resource availability: Routing can be improved if states of parking bays at arrival time are known[4]. If the parking situation is expected to be especially dire, people might consider switching to public transportation or depart earlier. Similar benefits arise for other types of resources such as rental stations.

In some cases, real-time information of resources is available, e.g., a city might decide to equip all its parking bays with permanently installed in-ground sensors. This is coupled with considerable expenses, and thus alternative ways of acquiring availability information have been proposed: The "ParkNet"[11] project successfully measured parking bay availability based on ultrasonic sensors of passing vehicles[11]. Another method uses smartphones: [9] describe multiple indicators for detecting parking and unparking activity. Both methods result in sparse data and thus, a prediction method for resource availability should be capable to cope with sparse observations in order to be widely applicable. A further requirement to model a city's resources is to incorporate time-dependent behavior. For example, a free parking spot in an office district will typically stay vacant much longer during nighttime than a few hours later when employees flock to nearby offices.

In this paper, we propose a novel prediction method for spatio-temporal resources which fulfills all the requirements named above. Our approach is based on a cyclic time-inhomogeneous discrete Markov model which learns transition possibilities from observed long-term observations but can predict short-term availability of a resource based on the last known observation. Our model is cyclic in time to model the typical change of availability during a certain time period such as a day or a week. Within this period we partition time into a set of discrete steps. For each of these steps, we allow varying transition probabilities which makes our model inhomogeneous in time. To allow the modeling of multiple spatially clustered resources, the states of our Markov model correspond to the number of available resources of a particular location. In order to train our model based on sparse training samples, we allow unknown observations and modify the well-known Baum-Welch expectation-maximization algorithm for hidden Markov models to estimate the parameters of our model. To show that our method provides accurate predictions based on the given information, we test our model on the real-world application of modeling parking bays based on two real-world datasets.

2 RELATED WORK

Predicting the availability state of resources, especially parking bays, is a task with great practical relevance and thus draws a lot of attention within the scientific community. [15] surveys various approaches to solve this problem. A method similar to our proposal is [3]. The authors apply a continuous-time, homogeneous Markov model with constant arrival and parking rates to meet the challenge of out-dated information in a vehicular ad hoc network (VANET). In contrast, our approach considers a time-inhomogeneous model which is capable to model availability patterns within the day.

Resource availability is also modeled by [4]. They propose a continuous-time Markov model in order to generate routes minimizing the expected time until a free resource is found near the destination. The model is time-homogeneous, i.e. a Markov model whose sojourn times in the two states "available" and "consumed" are exponentially distributed with *fixed* parameters defining the means of the respective distribution. These parameters are not inferred algorithmically but based on estimates by the authors.

Availability of parking spots has been estimated based on information about other spots *at the same time*[16]. The authors combine historical knowledge with real-time observations, but model dynamics, i.e. the development of state probabilities over time, are not covered. Thus, no prediction of future states is possible.

The prediction of resource availability *without* taking recent knowledge into account can be understood as a regression or classification problem. Consequently, machine learning methods are applied. For example, [2] predict the availability of parking bays using Support Vector Regression. The Melbourne dataset also used in our paper's evaluation was used before by [19], who compare Support Vector Regression, an approach using regression tree and neural networks. Note that we take recent knowledge into account.

Above approaches using Markov models are not applicable for data containing missing values. This was done by [18] with an expectation-maximization algorithm for a discrete homogeneous Markov model. A different approach was taken by [8] who train their discrete homogeneous models using a Bayesian approach. The estimation of parameters for homogeneous hidden Markov models trained on data including missing values was examined by [17].

3 PROBLEM SETTING

We define a *resource cluster* as a group of one or more resources with binary states of availability. Each cluster is modeled by one model. Thus, spatially related resources can be modeled by a joint model. A cluster's state can be measured at a given point of time t, i.e. the individual states of its resources become known. The sum of all resources being in the *available* state at this moment is denoted as O_t, the observation at time t, following the notation of [12]. Without loss of generality, we discretize time into a sequence of equally-spaced time-steps. Thus, we obtain an observation sequence $O = [O_1, O_2, ..., O_m, ...]$, in which O_m represents the last known observation. In this paper, we particularly consider sparse data and thus many O_t will be unknown. We denote these by $O_t = -1$.

Given an observation sequence O of a resource cluster, for example a group of one or more parking bays, our goal is to predict the number of available resources $E(m + d)$ of this cluster at an arbitrary time d after the last observation at time m.

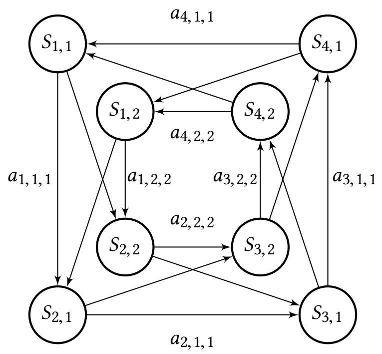


Figure 1: A cyclic Markov model with period length $p = 4$ and two states. $S_{t,i}$ is used as a shortcut for $q_t = S_i$. An edge between nodes $S_{t,i}$ and $S_{t+1,j}$ represents a transition probability denoted by $a_{t,i,j}$. Some labels are omitted for clarity.

4 TIME-INHOMOGENEOUS MARKOV MODEL

To satisfy the requirements given in the introduction, we modify the standard discrete, homogeneous Markov model given in [12]: This model is specified by a set of states S, its state transition probability distribution A and the initial state distribution π. The state of a resource cluster at time t is denoted by q_t and assumes one of the N possible states in $S = \{S_1, ..., S_N\}$. In our case, $S = \{0, ..., M\}$ with each state representing a number of currently available resources in a cluster of size M. A is given by a matrix $A = \{a_{ij}\}$ with a_{ij} being the probability of arriving in S_j when taking one time-step forward from S_i: $a_{ij} = P[q_{t+1} = S_j | q_t = S_i]$.

In our cyclic, inhomogeneous variant, transition probabilities vary with regard to the respective cycle position x in a cycle of period length p. Thus, we no longer have a single transition matrix A, but a set of transition matrices $A_x = \{a_{xij}\}$ with $a_{xij} = P_x[q_{t+1} = S_j | q_t = S_i]$. Note that while t may be any positive integer, x runs from 1 to the period length p. The cyclic model's initial state distribution π is likewise extended to include the cycle position x: $\pi = \{\pi_{xi}\}$ with $\pi_{xi} = P_x[q_1 = S_i]$. Figure 1 gives an example for a cyclic Markov chain with $p = 4$ and $N = 2$. In real-world applications, p will typically be larger, e.g. $p = 1440$ for a one day period discretized in minutes.

Knowing the state s_m at time m, calculating the state probability vector s_{m+d} after d time-steps is straightforward: We modify the original procedure of [12] to use the appropriate state transition matrix A_x when iterating through the chain. Note that x and t are aligned in a way that q_1 corresponds the first cycle position ($x = 1$).

$$s_{m+d} = s_m \prod_{j=m}^{m+d-1} A_{((j-1) \bmod p)+1} \tag{1}$$

We then obtain $E(m + d)$ using the equation $E(t) = \sum_{i=1}^{N} S_i s_t(i)$.

4.1 Estimating model parameters

If gap-less training data is available, model parameters can be calculated in the established way[14], which has to be only slightly adapted to reflect the newly introduced time-inhomogeneity. This modified algorithm is described next. Secondly, if training data contains missing values, we need to employ another algorithm. We propose a modified Baum-Welch algorithm described thereafter.

4.1.1 Estimating parameters from complete data. Analogous to [12], we use relative frequencies in observation sequence O to calculate state transition probabilities $P(q_{t+1} = S_j | q_t = S_i, O)$ for each (t,i,j):

$$P(q_{t+1} = S_j | q_t = S_i, O) = \frac{P(q_t = S_i, q_{t+1} = S_j | O)}{P(q_t = S_i | O)} \tag{2}$$

In the homogeneous case, both nominator and denominator would be accumulated over all t for all transitions (i,j). In the cyclic case, we only need to consider t in the set $\theta(x) = \{t \mid 1 \leq t < T \wedge ((t-1) \bmod p)+1 = x\}$ of times belonging to the same cycle position x. The state transition probabilities a_{xij} for the respective cycle positions x and states i,j, can then be calculated by accumulating and normalizing only the probabilities satisfying this restriction:

$$\begin{aligned} a_{xij} &= P(q_{t+1} = S_j | q_t = S_i, O, t \in \theta(x)) \\ &= \frac{\sum_{t \in \theta(x)} P(q_t = S_i, q_{t+1} = S_j | O)}{\sum_{t \in \theta(x)} P(q_t = S_i | O)} \end{aligned} \tag{3}$$

4.1.2 Estimating parameters from sparse data. If the observation sequence O contains missing values, we apply an algorithm based on the Baum-Welch algorithm for hidden Markov models (HMM). The motivation is that although the states of our model are not hidden in the usual way (i.e. states are never directly measurable), they are hidden in the sense of "being unknown to us" at times of missing values. To match our model to the HMM notation, we only have to introduce an observation symbol probability distribution $B = b_j(k)$ for each j of our N states[12]: $b_j(k) = P[v_k \ at \ t | q_t = S_j]$. In our case, the set of symbols $V = \{v_1, ..., v_M\}$ equals the set of states S, so $N = M$. B is not dependent on the cycle position x, as this was not needed for our use-cases. However, this extension can easily be done and may for example be appropriate if a camera-based recognition system principally has a lower accuracy at night.

To be able to use the Baum-Welch algorithm, we have to adapt it to our cyclic method: The equations for the forward, backward and intermediate variables α, β, and ξ have to be adjusted from their original forms in [12] by replacing a_{ij} with a_{xij}, with x being the position in the cycle. The equations for γ are not changed, as they do not include state transition probabilities. As defined above, x is calculated from t using the formula $x = ((t-1) \bmod p) + 1$, with p being the cycle length. Note that α and β are calculated recursively and initialized as in [12]: $\alpha_1(j) = b_j(O_1)$ and $\beta_T(i) = 1$.

$$\begin{aligned} \alpha_{t+1}(j) &= b_j(O_{t+1}) \sum_{i=1}^{N} \alpha_t(i) a_{xij} \\ \beta_t(i) &= \sum_{j=1}^{N} a_{xij} b_j(O_{t+1}) \beta_{t+1}(j) \\ \xi_t(i,j) &= \frac{\alpha_t(i) a_{xij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{xij} b_j(O_{t+1}) \beta_{t+1}(j)} \end{aligned} \tag{4}$$

The next iteration's a_{xij} are determined as in section 4.1.1:

$$a_{xij} = P(q_{t+1} = S_j | q_t = S_i, O, t \in \theta(x)) = \frac{\sum_{t \in \theta(x)} \xi_t(i,j)}{\sum_{t \in \theta(x)} \gamma_t(i)} \tag{5}$$

To cope with missing observations, we apply the "trick" of setting $b_j(-1) = 1$, i.e. assuming that the probability of an unknown observation is always 1[6, 17]. The reasoning behind this is that if we don't know the state at this point in time, all states are possible and therefore also all possible paths containing this state have to be considered when estimating state transition probabilities.

Given above equations, the estimation of A_x is done as usual: First, each matrix A_x is initialized with suitable values. For example, the majority of the probability mass is assigned to staying in the same state, i.e. setting diagonal values to almost one. Then, using A_x and fixed parameters π and b, the equations given above are evaluated to obtain values for α, β, ξ and finally a_{xij}. These a_{xij} then form the new A_x, to be used in the next iteration of this process. This cycle is repeated until a convergence criterion is met and the last A_x is returned[12]. We stopped when a_{xij} changed less than an ϵ threshold between two subsequent iterations.

5 EVALUATION

Two real-world datasets of parking events are used: Dataset *Canberra* recorded by the ACT government during their "SmartParking" trial[1] and dataset *Melbourne* provided by the City of Melbourne[2]. Each dataset contains gap-less data recorded by permanently installed in-ground sensors. Only "on-street" bays without exceptional permission (such as short-term parking opportunities or bays for the disabled) were considered. These were grouped into resource clusters by spatial similarity (i.e. street segment). This resulted in 160 bays in 21 clusters for *Canberra* and 5222 bays in 291 clusters for *Melbourne*. For both datasets, weeks three and six of a total of eight adjacent weeks were taken as the gap-less training set Tr_0, while the remaining six weeks form the testing set Te. Only weekdays were considered for both training and testing. To evaluate the algorithms under sparse data, additional training sets were constructed from Tr_0: In our application, gaps appear when a resource is not observed in a given minute. Assuming that arrivals of agents are independent and identically distributed, the time between two subsequent arrivals (and thus observations) is exponentially distributed with mean time between arrivals β. This distribution is used to sample sparse training sets Tr_β from Tr_0[14].

We compare the following prediction models and training procedures: Our proposed model trained using the modified Baum-Welch algorithm (**BW**) and the standard algorithm for gap-free data (**STD**), respectively. Then, a trivial model always "predicting" the last measured value (**LAST**). This is expected to perform very good for short-term predictions as changes of resource availability happen over time. Further, a model always returning the historic average (**AVG**) for the minute-of-day in question. AVG represents the complement to LAST because it only depends on historic data but does not consider the last known state. We also compare to SVM regression (**SVM**) [2] which represents a state-of-the-art approach for long-term predictions. For each resource cluster, one model was trained for each training set and tested independently from other clusters. All results were averaged over four repetitions, to

[1] "Smart Parking Stays": https://www.data.act.gov.au/Transport/Smart-Parking-Stays/3vsj-zpk7 licenced under "Creative Commons Attribution 4.0 International"
[2] "Parking bay arrivals and departures 2014": https://data.melbourne.vic.gov.au/Transport-Movement/Parking-bay-arrivals-and-departures-2014/mq3i-cbxd licenced under "Creative Commons Attribution 3.0 Australia"

Table 1: Normalized MAE for evaluated models.

	5 to 20 bays per cluster				1-bay-clusters	
	Canberra		Melbourne		Can.	Mel.
	hom.	inhom.	hom.	inhom.	inhom.	inhom.
BW	**0.197**	**0.142**	**0.174**	**0.121**	**0.258**	**0.230**
SVM	–	0.172	–	0.150	0.277	0.278
STD	0.202	0.167	0.175	0.151	0.293	0.241
AVG	0.423	0.348	0.315	0.250	0.475	0.350
LAST	0.206	0.206	0.186	0.186	0.273	0.255

reduce the variation introduced by the random sampling process. The cycle period was chosen to be one day, i.e. each weekday was assumed to show roughly the same behavior as is supported by the data. As we discretized time by the minute, this leads to a period length of 1440. Models were trained on datasets of different levels of sparsity ($\beta \in \{30, 60, 120\}$). Then, each minute in the testing data between 7 a.m. and 11 p.m. (at night, almost all bays are free which poses no challenge) was taken as a target and predicted by each model given a measurement $d \in \{15, 30, 60, 120, 240\}$ minutes before. The individual errors were accumulated per model to obtain mean absolute error (MAE) values. Because the number of resources differs between resource clusters, MAEs are normalized by the total number of resources of the respective cluster to allow the comparison of clusters of different size. For example, a model for a cluster of size twenty evaluating to an absolute MAE of 2.0 will result in a normalized MAE of 0.1.

Results for clusters containing 5 to 20 bays are shown in Table 1, both for the models mentioned above and their time-homogeneous counterparts, i.e. models using the same algorithm but not taking the time-of-day into account. As LAST always ignores the time-of-day by definition, results of both variants are the same. No time-homogeneous version of SVM was evaluated as it would merely output a constant prediction as the latest observation was not included in its training. When comparing the respective values, it becomes obvious that the time-inhomogeneous models show much better performance than their counterparts. This supports our choice to introduce time-inhomogeneous variants. Furthermore, predictions given by BW consistently result in lowest errors.

To evaluate whether our grouping of spatially similar parking bays into resource clusters and jointly modeling them by one model for each of those clusters is warranted, another experiment was conducted: In this experiment, each resource cluster (and thus model) only contains a single parking bay. The results are also listed in Table 1. Comparing these errors to previous results shows that combining similar bays improves predictive performance.

6 CONCLUSION

In this paper, we investigate the problem of predicting the availability state of resources given historic data and a recent measurement. Resources in Smart City settings are usually depending on the time of day. To address this, we propose a time-inhomogeneous Markov model able to model such cyclic behavior. As the historic data typically available in our setting is sparse, we present a modified Baum-Welch algorithm able to train this model with sparse data.

Evaluations on real-world datasets show that the proposed time-inhomogeneous Markov model combined with our modified Baum-Welch training algorithm yields better predictions on sparse data than other approaches such as the standard Markov model or an SVM. Also, time-inhomogeneous variants perform consistently better than their time-homogeneous counterparts, supporting our proposal to take the time-of-day into account. Evaluations further show that it is favorable to predict spatially related resources using a joint model instead of building isolated models for each individual resource.

REFERENCES

[1] Daniel Ayala, Ouri Wolfson, Bo Xu, Bhaskar Dasgupta, and Jie Lin. 2011. Parking slot assignment games. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 299–308.

[2] Fabian Bock, Sergio Di Martino, and Antonio Origlia. 2017. A 2-Step Approach to Improve Data-driven Parking Availability Predictions. In *Proceedings of the 10th ACM SIGSPATIAL Workshop on Computational Transportation Science*. ACM, 13–18.

[3] Murat Caliskan, Andreas Barthels, Bjorn Scheuermann, and Martin Mauve. 2007. Predicting parking lot occupancy in vehicular ad hoc networks. In *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*. IEEE, 277–281.

[4] Gregor Jossé, Klaus Arthur Schmid, and Matthias Schubert. 2015. Probabilistic Resource Route Queries with Reappearance.. In *EDBT*, Vol. 15. 445–456.

[5] Gregor Jossé, Matthias Schubert, and Hans-Peter Kriegel. 2013. Probabilistic parking queries using aging functions. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 452–455.

[6] Limin Liu. 1997. Hidden Markov models for precipitation in a region of Atlantic Canada. (1997).

[7] Viktor Losing, Lukas Rottkamp, Michael Zeunert, and Thies Pfeiffer. 2014. Guiding visual search tasks using gaze-contingent auditory feedback. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. ACM, 1093–1102.

[8] Junsheng Ma, Xiaoying Yu, Elaine Symanski, Rachelle Doody, and Wenyaw Chan. 2016. A Bayesian Approach in Estimating Transition Probabilities of a Discrete-time Markov Chain for Ignorable Intermittent Missing Data. *Communications in Statistics-Simulation and Computation* 45, 7 (2016), 2598–2616.

[9] Shuo Ma, Ouri Wolfson, and Bo Xu. 2014. UPDetector: Sensing parking/unparking activities using smartphones. In *Proceedings of the 7th ACM SIGSPATIAL international workshop on computational transportation science*. ACM, 76–85.

[10] Felix Mata and Christophe Claramunt. 2013. Augmented navigation in outdoor environments. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 524–527.

[11] Suhas Mathur, Tong Jin, Nikhil Kasturirangan, Janani Chandrasekaran, Wenzhi Xue, Marco Gruteser, and Wade Trappe. 2010. Parknet: drive-by sensing of road-side parking statistics. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM, 123–136.

[12] Lawrence R Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77, 2 (1989), 257–286.

[13] Kotagiri Ramamohanarao, Jianzhong Qi, Egemen Tanin, and Sadegh Motallebi. 2017. From How to Where: Traffic Optimization in the Era of Automated Vehicles. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 10.

[14] Sheldon M Ross. 2014. *Introduction to probability models*. Academic press.

[15] Yogesh Tayade and MD Patil. 2016. Advance Prediction of Parking Space Availability and Other Facilities for Car Parks in Smart Cities. *International Research Journal of Engineering and Technology* 3, 5 (2016), 2225–2228.

[16] Bo Xu, Ouri Wolfson, Jie Yang, Leon Stenneth, S Yu Philip, and Peter C Nelson. 2013. Real-time street parking availability estimation. In *Mobile Data Management (MDM), 2013 IEEE 14th International Conference on*, Vol. 1. IEEE, 16–25.

[17] Hung-Wen Yeh, Wenyaw Chan, and Elaine Symanski. 2012. Intermittent missing observations in discrete-time hidden markov models. *Communications in Statistics-Simulation and Computation* 41, 2 (2012), 167–181.

[18] Hung-Wen Yeh, Wenyaw Chan, Elaine Symanski, and Barry R Davis. 2010. Estimating transition probabilities for ignorable intermittent missing data in a discrete-time Markov chain. *Communications in Statistics-Simulation and Computation* 39, 2 (2010), 433–448.

[19] Yanxu Zheng, Sutharshan Rajasegarar, and Christopher Leckie. 2015. Parking availability prediction for sensor-enabled car parks in smart cities. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2015 IEEE Tenth International Conference on*. IEEE, 1–6.

Chapter 8

Quantifying the Potential of Data-Driven Mobility Support Systems

Publication

Lukas Rottkamp and Matthias Schubert. “Quantifying the Potential of Data-Driven Mobility Support Systems”. In: *Proceedings of the 13th ACM SIGSPATIAL International Workshop on Computational Transportation Science*. IWCTS ’20. Seattle, Washington: Association for Computing Machinery, 2020. ISBN: 9781450381666. DOI: 10.1145/3423457.3429366

Contribution

Lukas Rottkamp is the first author of this publication. In a discussion with Matthias Schubert, he realized that cost savings through better data can be estimated but depend on the use-case. Lukas Rottkamp thus devised a method to determine the potential benefits of mobility support systems. He ran experiments to compare different parking search methods against the optimal strategy, going by taxi and the case of going without assistance system, respectively. He then produced the manuscript. Matthias Schubert contributed to this publication through regular discussion of the ongoing work. Matthias Schubert also reviewed drafts of the manuscript and helped polishing it before submission.

Quantifying the Potential of Data-Driven Mobility Support Systems

Lukas Rottkamp
Audi AG
Ingolstadt, Germany
lukas.rottkamp@audi.de

Matthias Schubert
LMU Munich
Munich, Germany
schubert@dbs.ifi.lmu.de

ABSTRACT

When traveling it is often necessary to take a detour, for example to find an on-street parking opportunity or a charging station. Numerous systems intending to reduce time or other resources spent on such detours have been presented. An example are methods guiding drivers to free on-street parking opportunities. However, the question of how much can actually be saved by using such solutions when compared to the status quo remains largely unanswered. Often, the cost attached to these detours is unclear. In this work, we present a generalized approach to answer these questions: A methodology consisting of an evaluation environment powered by real-world data and implementations of different scenarios. We then illustrate our proposal by using it to quantify the potential of an optimal assistant for finding on-street parking opportunities. We further show how to generate synthetic but realistic parking data when real-world data is not available.

CCS CONCEPTS

• **Applied computing** → **Transportation**; • **Information systems** → **Spatial-temporal systems**; • **Computing methodologies** → *Model development and analysis.*

KEYWORDS

smart city, mobility, transportation, routing, spatial resources, parking, spatio-temporal simulation

ACM Reference Format:

Lukas Rottkamp and Matthias Schubert. 2020. Quantifying the Potential of Data-Driven Mobility Support Systems. In *13th International Workshop on Computational Transportation Science (IWCTS'20), November 3, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3423457.3429366

1 INTRODUCTION

Mobility is a key element of societies which enables commercial endeavors such as trade and meeting in workplaces, as well as direct social interaction. Traveling always comes with a cost: Primarily the time spent for the trip but often also expenditures for fuel or parking fees. Even worse, it is often not possible to take the best-possible path to the destination: When taking the car, it has to be parked before the remainder of the way to the destination is completed on foot. Thus, we do not just waste time on the actual search, but also on the walk from the eventually found parking bay to the destination. Another example regards the vehicle itself: Its energy storage has to be replenished in certain intervals, which becomes especially relevant with the current rise of battery electric vehicles. Parking and recharging necessitates detours, which increase total trip times. More examples can be found, such as detours caused by blocked roads or when sharing a ride with others.

As these detours cost time and money, evaluating their length and how to reduce these costs, are of great interest for travelers as well as for cities, transport agencies and car manufacturers. Thus, the first question addressed in this paper is: **How large is the impact of these detours when compared to the baseline case?** The baseline case is defined as the situation in which the detour in question is not necessary. For example, taking a taxi gets rid of the need of parking and a vehicle with a conventional combustion engine is not affected by the recharging problem as cruising ranges are still significantly larger and refueling still takes much less time than recharging a battery.

To minimize these detours considerable effort is undertaken. For example, the search for parking is optimized by parking guidance systems. The detour needed for recharging batteries is reduced by intelligent placement of charging opportunities, such as near points of interest as supermarkets or at certain highway stops. This leads us to the second question: **How big is the potential of the used strategies for minimizing these detours?** Answering this question is arguably even more important as it is mandatory to justify the cost associated to engineer, buy and maintain potential solutions. For example, several solutions for Smart Cities accrue considerable costs when sensors need to be installed throughout the street network. Of course, the service quality depends on the quality of this data, but it is often unclear to which extent. It is important to note that these savings are depending on the current setting–the best parking guidance system for example does not provide much value if the destination area is guaranteed to provide an almost inexhaustible pool of available parking opportunities.

This paper presents an approach to answer both questions: We determine the cost and potential savings by simulating different scenarios in an evaluation environment powered by realistic data. To get insights into the minimally obtainable cost, we propose to include the theoretically **optimal scenario** (e.g. having full information about the environment). We also include a **baseline scenario** to determine the cost when the necessity of a detour is absent. Third, a **status quo scenario** is evaluated to obtain estimates for the cost currently incurred. It is of course advantageous

IWCTS'20, November 3, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-8166-6/20/11…$15.00
https://doi.org/10.1145/3423457.3429366

to also include other scenarios which are expected to give results somewhere in-between. These are then compared to optimal and baseline scenarios to determine the possible advantage (and therefore value) of a strategy for reducing the cost of the necessary detour.

Relevant metrics for quantifying cost of a detour may include the congestion level of roads, spent fuel, pollution and time used. Time wasted is one of the most important metrics because various other metrics directly depend on it. It is also often the immediate source of frustration, as in our case study of hunting for a free parking spot and then walking from there to the actual destination. Thus, we focus on time, keeping in mind that a reduction in trip time will also decrease traffic, pollution and energy wasted.

Related work using simulations often aims to analyze the collective dynamics and effects of a multitude of drivers in a city, for example, a swarm of drivers searching for parking opportunities. As such, these simulations may involve thousands of agents, hoping to recreate (or approximate) realistic distributions. Because of the difficulty of building a simulator which adequately recreates these, for example, parking availability distributions as observed in the real world, we do not attempt to do so but use prerecorded data. This data follows realistic distributions because it was recorded in reality or at least generated using generative models estimated from real observations. This allows us to use only a single agent in each simulation run–the other agents' effects are covered by real-world data. In this sense, we basically replay a time span in history. The combination of prerecorded agents with simulated agents can be problematic because the simulated agents cannot influence the recorded agents' actions or state. While it is true that the simulated agent would in reality slightly change the state of the system, e.g. the traffic flow, we argue that the influence of just one agent is negligible with regard to the overall outcome.

If no real-world data is available, our methodology can still be carried out by using synthetic data. Such data can be generated with models of the domain in question. In this case, only a few parameters are required which may be easier procured than real-world data. Section 5 describes how to generate such data in our on-street parking use case. It should however be noted that realistic behavior is often not accurately reproduced by artificial models, especially when these are only comparatively simple approximations of the real processes. Let us note that it is outside of the scope of this paper to examine more accurate methods to estimate real-world distributions. We argue that real-world data is strongly preferable as necessary assumptions when using synthetic data may lead to unrealistic simulation behavior and ultimately wrong conclusions.

To summarize, our approach enables the quantification of cost attached to a necessary detour, as well as the potential of methods to decrease the cost of these detours. This enables decision-makers to more accurately determine if cost associated with a service (or certain effort for implementation) will be justified by the value the method is expected to provide. In the following, we first discuss related work and then present our methodology in detail. We then illustrate our approach by applying it to the use case of on-street parking, with both real-world and synthetic data, before we end with concluding remarks.

2 RELATED WORK

We first discuss existing work related to our general methodology, i.e. regarding our two questions of determining impact of use cases and the potential of solutions lessening this impact. Afterwards, we review work regarding our on-street parking case study.

2.1 Use cases' impact and solutions' potential

Cici et al. [9] evaluate the potential of ride-sharing systems using a data-driven analysis of possible shared rides. They assume that individual trips of neighbors to the same work area can in principle be shared. Based on cellphone data, they identify the homes and workplaces of users and determined shared rides. A 52% reduction of the amount of cars is reported, which the authors take as an upper bound for the potential of car pooling. As not everybody is able or willing to participate in a car pooling scheme, the real reduction is expected to be lower. This is similar of our approach of also including an optimal scenario in our evaluation.

Turečková and Nevima [20] discuss approaches to determine the efficiency of smart city solutions not restricted to transportation including a cost benefit analysis (CBA) done before the solution is implemented. This allows to estimate potential benefit before funds are allocated, which is also one of our goals. It should be noted that they evaluate efficiency with regard to numerous factors including societal, economic end environmental ones. Our methodology can be used for determining the value (not only measured in money but also other resources as time and environmental ones) of a solution which can then be used alongside these other factors in a CBA.

The placement of gateways in 5G cellular networks is analyzed by Kiess and Khan [13] with regard to the evaluating the potential of using more than one gateway in a nation-wide network. A scenario consisting of a central gateway is taken for comparison, as is the theoretically best scenario of a direct transmission without gateways. As with our examples, this best scenario is not feasible in the real world according to the authors. They describe and evaluate further scenarios consisting of more than one gateway to determine the potential of this strategy. It is shown that a strategy of four gateways could yield a significant reduction in transmission cost. While the placement of gateways in cellular networks does not directly relate to transportation, this application affirms our general methodology. It should be noted that their model-based approach may have benefited from a simulative approach using real world data, as we propose in this paper.

Zhan et al. [24] present an efficiency analysis of the New York taxi system: They use real-world data of taxi trips to calculate the total empty trip cost of these trips. Because empty trips are not earning money, they are inefficient although necessary. For comparison, the authors algorithmically determine a second mapping of taxis to passengers which minimizes empty trip cost while at the same time retaining the same trips as in the original data. Such a comparison between status quo and an optimal scenario is very much in the spirit of our paper–it relates to our second question regarding the potential of a solution.

2.2 Simulation of on-street parking

Parking is an important topic in most cities and thus a well-studied research area. Assigning parking bays to individual vehicles[1] or pricing parking events[2] is often of particular interest, other studies include side-effects such as road congestion or pollution caused by vehicles looking for a parking opportunity[18]. For this paper, determining the time lost by searching for the parking lot is especially relevant. One strategy is to analyze real-world data regarding average time spent looking for parking.

Though it is beneficial to analyze parking as it occurs in reality through surveys or data analysis, research focuses on evaluations based on artificial scenarios by devising appropriate models or simulations. These have the advantage that parameters as parking demand, street graphs, traffic patterns etc. can be modified to see how certain measures or future developments would influence the relevant metrics. For instance, Belloche [3] use mathematical models for estimating parking search time: Based on a survey in Lyon including 896 usable parking approaches and their respective search times, they argue that parking search time can to some degree be approximated by an exponential model based solely on the congestion ratio. Taking a different approach, Zhang et al. [25] use timed petri nets to model driving towards parking opportunities based on availability information given by parking providers.

Especially relevant to our on-street parking use case are approaches making use of a simulation. Benenson et al. [4] describe the PARKAGENT model to simulate residential parking in Tel Aviv. Their agents use a parking search algorithm inferred by observing human drivers, described below in Section 4.2. Their simulator is applied to obtain insights into the effects of a hypothetical new parking garage. Steenberghen et al. [19] simulate agents looking for parking opportunities to determine effects on traffic in Leuven (SUSTAPARK). Waraich et al. [22] propose a parking search model for the MATSim toolkit[1]. This model chooses from different strategies, such as on-street parking, off-street parking or parking on private grounds, based on their predicted utility with regard to factors as trip time, walking distance and price.

In our case study, we apply our methodology to determine the potential of on-street parking guidance systems. The existence of this potential is of course well known and thus route guiding system for parking have been proposed before:

Boehlé et al. [6] present a route guidance system which aims to reduce both time wasted in traffic and time wasted for parking search. Information about traffic and parking availability is shared and parking bays are reserved in advance. Their simulations show that a drastic reduction of overall travel time is already achieved when only 10% of cars use the reservation system. It is however unclear which extend these improvements carry over into the real world, as the simulation is done in artificial cities and is not stated how many parking opportunities were present and how routes were set-up.

Savings in travel time are determined by Waterson et al. [23] in a simulation of a simplified city plan of Southampton. Only off-street parking (i.e. parking garages) is considered, while we focus on on-street parking. Their simulation shows that Parking Guidance and Information (PGI) signs (directing drivers to free parking garages) bring almost no reduction in travel times and may therefore not be justified in this regard.

[1] https://matsim.org/

Friedrich et al. [10] present two algorithms to search for a best route in a probabilistic graph: A branch and bound algorithm used when having probabilistic data and a heuristic used when only map data is available. They also describe a field study in different areas of Berlin, comparing the performance of a human driver with their search algorithms. This study used 87 runs in 36 situations over four days. 15 of these situations were selected for evaluation ("difficult situations") as the others did not pose a challenge for finding a suitable parking opportunity. Results indicate that their approach can beat a human driver with regard to total trip time.

Liu et al. [16] propose and on-street parking guidance system and evaluate it using real-world parking data from the city of Melbourne. They use a receding-horizon approach in a multi-agent setting. Routes are generated based on predictions of parking availability. The authors report significant improvement compared to the "Smart Parking" algorithm given by Geng and Cassandras [11].

All approaches for route guidance systems mentioned above aim to decrease time spent for parking and substantiate their respective proposals with good arguments such as quantitative comparisons against previously published approaches. However, none considers the question of how much time could actually be saved in the first place (the "second question" we state in the introduction). It is also not stated how grave their respective parking problem is when compared to the baseline of not needing to park at all (our "first question"). We argue that answering those questions would have improved the value of their evaluations, as readers would be able to better appraise their contributions. As it is in fact not technically difficult to obtain answers to these questions when a simulation environment is available, we hope that future publications take the extra mile to give answers to these questions according to our proposed methodology. Of course this hope is not restricted to use cases regarding parking guidance system but all use cases onto which our general approach can be mapped.

3 METHODOLOGY

As motivated in the introduction, we aim to tackle two questions: First, how large is the cost of necessary detours when compared to the baseline case. Second, how large is the potential of any strategy for minimizing the detours. We answer both questions by simulating the use case in question in a realistic environment. Table 1 shows an overview of terms used in our methodology which are detailed in the remainder of this section.

3.1 Use case

The use case always includes going from origin to destination under a predefined set of constraints. These constraints are central in our approach as they necessitate a detour. In the following, we will take a use case related to charging stations as our running example: We suppose that a vehicle needs recharging between starting the journey at the origin and reaching the destination. Therefore, a detour is mandatory, which not only consists of additional distance traveled but also additional time needed because it might be necessary to wait for a free charging station. In our scenario, we want to determine suitable locations for charging station given a data set of regularly traveled vehicle trajectories for which the detour over the charging station should be minimized.

Table 1: Overview of terms used in our methodology

Term	Definition
Use case	Task of vehicle with necessity causing detour
Scenario	Approach being evaluated which defines type of agents and their configuration (e.g. amount of knowledge/data provided)
Agent	Entity moving through the environment to its destination
Environment	Infrastructure to use by the agent, i.e. road graph, parking bay locations
Setting	Parametrization of the environment, i.e. traffic, destinations, availability of resources (e.g. charging stations)
Metric	Used for measuring cost, usually total trip time

Table 2: Definitions of main scenarios

Name	Definition	Example
Baseline	Detour not necessary	No charging necessary
Optimal	Detour minimized perfectly	Optimal placement of charging stations
Status Quo	Detour as is now	Current placement of charging stations

3.2 Scenario

A scenario describes a setting to solve the given mobility task. Thus, scenarios might vary with respect to constraints and available information. We propose three main scenarios summarized in Table 2:

The **Baseline** scenario is used to provide a baseline for a particular use case, i.e., the detour is not required. It is used to determine a lower bound on the cost of the use case. In our charging station example, the baseline would be that no charging is necessary and thus no cost is incurred by the detour of any considered trip. The **Optimal** scenario includes detours but assumes optimal preconditions to minimize detours. It is used to obtain a lower bound for all methods that require a detour. Thus, the optimal scenario often will not be practicable or even possible in the real world. In our running example, it would mean that charging stations can be placed optimally. Thus, we assume free placement of stations and do not consider waiting times. As a third scenario, we propose to include a **Status Quo** scenario describing the preconditions the use case has to be tackled by the current state-of-the-art methods. Thus, it sets the preconditions for solutions which can be practically applied. In our charging station example, real charging stations with their properties such as location, amount of individual charging bays, available charging voltages etc. are used. Evaluation of this scenario will provide estimates of currently observed detour costs.

Note that additional scenarios describing the status quo are possible. For example, it may be of interest to evaluate the currently used strategy of placing new charging stations, e.g. at supermarkets or existing fuel stations. The strategy can than be evaluated against the theoretically best strategy determined through the optimal scenario described above.

Apart from these three main scenarios, it is usually beneficial to add more scenarios according to the use case and connected questions. These typically consist of different approaches to minimize the detour, e.g. different algorithms or cost functions for placing charging stations. In our experience, it is especially interesting to also include scenarios with varying quality of data as for example different types of sensor networks (static sensors, data collection by vehicles themselves and so on). Many such scenarios can be thought of, for example varying in the way of data is acquired: data may be recorded by infrastructure sensors (e.g. cameras or in-ground sensors), the vehicles itself, smartphone apps etc. These can typically be modeled quite easily when an simulation environment is present by for example implementing virtual in-ground sensors which provide additional data to the agents.

Depending on the scenario, the status quo may be already observable in the real-world. In this case, it may only be necessary to evaluate the new solution if the scenario can be reproduced with sufficient accuracy, ideally in the same environment, i.e. the real world. If not, both may be evaluated in a simulation–with the advantage that the simulation can directly be verified using the real-world data.

3.3 Agent

The entity going from origin to destination is called an agent. It may be designed to reflect human behavior or to represent a user using a new service to be evaluated. In other words, we do not model a given service explicitly but implicitly through the behavior of agents. Agents take actions which may or may not advance them to their destination. Different agents may be provided with different sets of actions depending on the concept they are representing. For example, an action might be to take a turn at an intersection or to occupy a particular charging station.

For evaluating the baseline scenario and the optimal scenario, we require that agents act optimal with respect to the respective constraints. Since both scenario are used to generate lower bounds, sub optimal behavior would lead to losing the bounding property here. Fortunately, both scenarios are simplified settings which allow for deterministic solutions under best case assumptions.

For all other scenarios, agents act according to the configuration of the given scenario, such as real-time sensor data or data about charging station reservations. The scenario also defines the constraints which must be followed by the agents, e.g. whether the environment is completely or partially observable. For example, an agent might only have access to current occupancy of some of the charging stations.

In our example of evaluating the placement of charging stations, individual electric vehicles represent agents. Additional to the origin and destination as described above, each has a battery charge state which decreases during the trip according to a realistic model of energy use while driving. Actions of the agents include choosing the route taken and the charging stations used. Depending on the scenario, agents may be informed of the availability state of charging stations and/or other agents' planned uses of charging stations.

Let us note that the quality of the agent policy plays an important role when evaluating scenarios. Thus, evaluations can only measure how beneficial a certain scenario is in combination with a given agent policy. For example, if the scenario would provide information which the agent cannot exploit, we have to consider that a better policy exists which could exploit the provided information.

3.4 Environment

The most realistic evaluation environment is arguably the real world itself–a real city with real entities. However, properly executed field studies are typically expensive while data obtained in a passive way is often lacking the requirements for determining comparable results regarding the metric of interest. In some settings, field studies are virtually impossible, e.g. when covering large time-frames or a huge number of entities (vehicles/persons). Simulations also bring the advantage of enabling a multitude of scenarios under varying settings for evaluation. It is generally possible to repeat simulations while changing only parts of the parameters and thus, excluding unintended side-effects. A simulation can also be repeated multiple times to obtain average results, which increases the reliability of results when random effects are present (for example caused by non-deterministic agents). The environment in this simulation typically mirrors a real one. In this case, street graph, speed limits and further properties can be copied from digital maps of real transportation networks. For instance, the location of existing charging stations and/or possible locations for new charging stations would be included in our charging station placement example.

3.5 Setting

When using a simulation, realistic distributions of entities relevant to the application (such as traffic patterns or availability of charging stations) are needed, which may be acquired from the real world or generated synthetically. Of course, it is best to use data of the real-world, to minimize the divergence of reality and simulation. Luckily, numerous data providers provide real-world datasets, often at no cost and non-restrictive license terms. If no fitting real-world dataset can be obtained, it may be possible to build a synthetic dataset with realistic properties. For example, when the exact occupancy of individual parking bays is not available, but aggregate distributions are, these aggregates may be used to generate the target distributions. Of course a careful modeling is needed to not diverge too much from reality.

To obtain reliable values, settings should be initialized extensively enough to cover the range of external distributions (such as traffic patterns changing by time of day). This is especially important in use cases sensitive to changes in these distributions. For instance, we expect our charging station example to be especially sensitive to the time of the day, as charging demand significantly varies over the course of the day and charging a vehicle takes a significant amount of time during which the used charging bay is not available to other agents. It is of course beneficial if real data is available enough so that multiple sets of different times of day and/or dates can be examined. Furthermore, it is generally useful to include a variety of settings during an evaluation. For example, it is often interesting to use traffic patterns not present in reality now but expected in projections of the future. If random influences are present, especially when non-deterministic algorithms are involved, simulation runs should be repeated appropriately to obtain good approximations of the expected values.

3.6 Metric

In transport applications, various metrics are related to time. For example, time spent on the road can be less productive or enjoyable and professional drivers need to be paid per hour. Other candidate resources include fuel spent or a reduction of emissions. Time can be measured directly or indirectly through related metrics such as traffic flow speed or traffic throughput. Depending on the application, other metrics may be better suited. For example, when car occupants aren't strong walkers or when unwieldy objects are being transported, the walking distance after parking the vehicle may be critical. However, the total trip time is often the most important metric. This also applies to our example of charging station placement, as time spent for charging is perceived as one of the top disadvantages of electric vehicles [14].

3.7 Evaluation

After obtaining the results of running the simulations, differences between certain scenarios are analyzed: The difference of *Baseline* and *Optimal* scenarios indicates the gravity of the use case, i.e. if it necessitates big detours even in the best-possible way or has little impact on the overall trip cost. This targets the first question of our paper. The difference of *Baseline* and *Status Quo* scenarios also provides an answer to it, in a way providing a high bound for the cost which should be approached with suitable measures. In our example of charging station placement, this cost represents the additional amount of time a battery electric vehicle currently needs when compared to a vehicle with combustion engine. The difference between *Optimal* and *Status Quo* scenarios aims at the second question: It directly indicates the potential of those measures. If the potential turns out to be very small, even expensive measures would not lead to a significant cost reduction. If the potential is in contrast very large, it can be warranted to invest large sums in strategies to alleviate the cost. Of course these strategies can itself be implemented as scenarios according to our methodology, to determine if those investments will lead to the anticipated cost reduction. When looking at our charging station placement example, the low cost of the optimal scenario likely comes with a prohibitively high investment. Limiting certain parameters in further scenarios, e.g. the amount of charging stations to be placed or the amount of money invested, would be interesting to determine the cost of realistic approaches which should of course again be compared against the main scenarios given above. It may be especially interesting to determine which scenario represents a turning point, i.e. is the first to provide a significant benefit. It is for example likely that even an advanced parking guidance assistant yields no improvement when parking opportunities are abundant in the destination area. Now, different scenarios with increasing percentages of occupied parking bays can be evaluated: It is likely that the parking assistant will start to outperform the status quo approach at a certain occupancy value. This value would be interesting to know in order to determine if it makes sense to deploy the service (which may be associated with a certain cost) in a certain district or not.

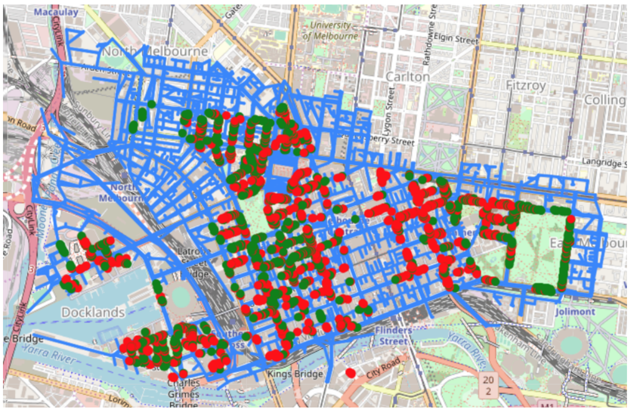


Figure 1: Snapshot of the street graph and on-street parking bays during a simulation run. Street graph shown in blue, free parking bays as green dots, occupied bays as red dots.

4 CASE STUDY: ON-STREET PARKING

We now illustrate the general approach described above by evaluating the use-case of on-street parking. Here, a traveler has the task of parking at a free parking bay and walk from there to the destination. Our goal is to obtain insights into the potential of parking guidance services but also into the general use case and costs of associated detours. Detours consist of the necessary search for a parking bay and the time incurred for walking to the actual destination.

4.1 Evaluation Environment

For simulating parking search, a modification of the COMSET simulator created by van Barlingen et al. [21][2] is used. While COMSET was originally built to evaluate taxi search time while looking for customers, its event-based engine and well-structured code-base allow for straight-forward adaption to other domains. COMSET uses street network data from OpenStreetMap[3], which is made available under an open data license. This makes it convenient to use street graphs other than the originally included street graph of New York City.

In our modification, on-street parking bays are added to the street graph at their real-world location. Additionally, two new events are introduced: Parking bays change their occupation state based on prerecorded data and agents are notified when they pass a resource during their trip, at which point they may record the observed occupation state and, if the bay is free, may decide to park there. Parking bay locations and their respective state-change events are loaded from real-world data sets, which will be described in Section 4.3. Since we aim to also measure the time needed for walking from the parking position to the actual destination, a method for determining walking time from parking bay to destination was added: We determine air-line distance and then multiply it with a constant walking speed also addressed below in Section 4.3. While we systematically underestimate walking distance this way, we decided against implementing a more complex calculation as pedestrians can often take unmapped shortcuts through buildings or open areas and walking routes in this use case usually turn out to be quite simple, such as down a road.

[2]COMSET on Github (MIT license): https://github.com/Chessnl/COMSET-GISCUP
[3]https://www.openstreetmap.org - Data ©OpenStreetMap contributors

Table 3: Scenarios in the on-street parking use case.

Name	Description
Baseline	No parking necessary (passenger takes taxi)
Optimal	Omniscient agent knows future
Status Quo	Human-like agent searches without help

4.2 Scenarios and Agents

We implement the three main scenarios as shown in Table 3:

The **Baseline** scenario consists of going by taxi, i.e. the agent finds the best (with regard to total trip time) drop-off point, stops there and the passenger then walks from that point to the destination. Thus no detour is necessary as required.

In the **Optimal** scenario, the agent has full knowledge of the future and parks at parking bay b_{opt}. This bay is defined as the bay having the minimal total trip time $t_{\text{total}}(b) = t_{\text{drive}}(b) + t_{\text{wait}}(b) + t_{\text{walk}}(b)$ of all bays b in the evaluation environment. Here, $t_{\text{drive}}(b)$ is the driving time from the agent's current position to b, determined according to the street graph (taking the fastest route). If b is free at arrival time, $t_{\text{wait}}(b)$ will be zero. Otherwise, the agent cannot park and must keep driving, as stopping and waiting for a bay to become free is forbidden because it would likely impede traffic and thus not be socially acceptable. In this case, the agent however returns to b on the fastest route and checks the bay's occupancy state again, adding the time of the detour to $t_{\text{wait}}(b)$. This is repeated until the bay is free and thus can be occupied. The walking time $t_{\text{walk}}(b)$ is the time needed to walk from b to the destination. Note that the agent will always reach the destination after time $t_{\text{total}}(b_{\text{opt}})$ as no random effects are present in our simulation and the agent's knowledge is flawless. As motivated above, we include this unrealistic (as perfect knowledge about the future is not available in the real-world) scenario to gain insights in the optimal performance for reasons of comparison.

For the **Status Quo** scenario, we use an agent approximating the parking strategy of a human driver who has no knowledge about the parking opportunities in the destination environment nor about their current or future availability states. In Benenson et al. [4] (PARKAGENT), such an agent was described: The agent drives towards the destination using the fastest route. When the agent is less than an *awareness distance* D_a (250 m) away from the destination, it slows down to 25 km/h and starts counting occupied and unoccupied parking places while still driving towards the destination. As soon as the agent is less than a second, smaller, *parking distance* D_p (100 m) away from the destination, it further reduces its speed to 12 km/h. On each passed parking opportunity, a formula is evaluated to decide if this opportunity should be taken. If the agent has not parked until the destination is passed, the third stage is entered: The agent takes the first parking opportunity encountered while cruising in an area given by radius D_p around the destination[15]. D_p is increased by 30 m each minute as agents become increasingly desperate[4].

It should be noted that, alternatively, other agents could be used for the status quo scenario. For example, Bischoff and Nagel [5] describe an on-street parking strategy they call "simple random search logic": The agent drives towards the destination using the fastest route and then takes the first parking opportunity it encounters. If no parking opportunity is available in the remainder of the destination street, a random sequence of neighboring streets is likewise searched until a free parking bay is found. We choose the PARKAGENT model because its authors specifically designed it for recreating the behavior of real drivers searching for an on-street parking opportunity, a goal we share.

We record the individual total trip times, being the sum of driving time and walking time, for later evaluation.

4.3 Evaluation Setup

We will now describe the simulator setup used for obtaining measurements for our on-street parking example.

4.3.1 Street graph. The street graph of the Melbourne Central Business District (CBD) is obtained from the OpenStreetMap project mentioned above. It contains 3185 nodes (intersections) and 6384 directed edges (roads) with a combined length of about 305 kilometers over an area of approximately nine square kilometers.

4.3.2 Parking bays. The City of Melbourne, Australia, maintains an open data platform[4] where it provides various datasets under creative commons license.

This data platform provides the dataset "On-street Parking Bays"[5], of which we extract the location of on-street parking bays. These bays are then added to the nearest road in our street graph. We are interested in a typical parking setting, so we do not include bays for short-time parking or bays which require extra permissions, such as bays for people with special needs.

4.3.3 Parking data. To obtain realistic simulation results, we use prerecorded parking bay occupancy state changes i.e. the parking bays in our simulation behave in the same way as their real counterparts did at a certain day in the past.

The data platform just mentioned provides the dataset "On-street Car Parking Sensor Data - 2017"[6] which contains such data corresponding to the parking bays: Occupancy state changes were recorded throughout the year 2017 by permanently installed inground sensors. The temporal resolution is one second, which allows a realistic replay. It should be noted that the dataset contains artifacts such as overlapping occupancy events which were corrected prior to using it.

A snapshot of the simulation's parking bay occupation states can be seen in Figure 1.

4.3.4 Speed of travel. Though the maximum allowed speed for a given street is provided by OpenStreetMap, the speed observed in reality is generally lower due to traffic lights, congestion and similar influences. It would be ideal if actual historic travel speed values could be used during our simulations. Unfortunately it was not possible to obtain these for same time spans as covered by our parking data. Thus we take the same approach as the original authors of COMSET did: They determine a global "calibration" factor by comparing real trip times with trip times obtained by simulating the same routes assuming the maximum allowed speed for each street segment[21]. Taking the same approach, we arrive at a factor of 1.35 for trips in our street network, i.e. a given trip takes approximately 35% longer in reality than the shortest path according to the speed limit. Agents include this factor in their calculations and it is also reflected in simulated driving speeds.

When measuring walking time, we assumed no delays due to traffic and thus set the walking speed to a constant value. We use a walking speed of 1.42 m/s as determined by Browning et al. [7].

4.3.5 Simulated trips. As we want to determine the time an agent needs to find a parking opportunity, we need to determine start and destination locations for these trips. For our evaluation, we determine one destination location for each road longer than 20 m and shorter than 200 m randomly using a uniform distribution in this range. This range restriction is included to not over-represent areas with lots of small connecting roads and exclude long road segments not relevant to parking as those belonging to highways. For each of those destination locations, we randomly determine a start location so that the shortest road distance between start and destination equals 750 m (using a uniform distribution over all locations satisfying the constraint). This distance was chosen experimentally to be long enough to allow the agent some variation in approaching the destination (e.g. from the north or south), while not restricting the set possible starting locations too much. This leads to 4085 pairs of start and destination location.

To be able to analyze parking search times with regard to time of day, trips were generated for each of those start-destination-pairs at multiple starting times. According to the data, demand for parking is typically high from morning to afternoon in our evaluation area. As we are interested in challenging situations, we choose evaluation time segments accordingly: Hourly from 8:30 a.m. to 3:30 p.m. at a random day in our data (August 9, 2019), resulting in 32680 trips.

Each of those trips was simulated independently for each agent type. Agent types involving a random component were simulated three times and then averaged to account for random fluctuations. This number was chosen empirically to be large enough to reduce variance due to random effects while still keeping computation times low. While three is not a large value for such attempts, the large amount of trips helps to smooth out random effects well.

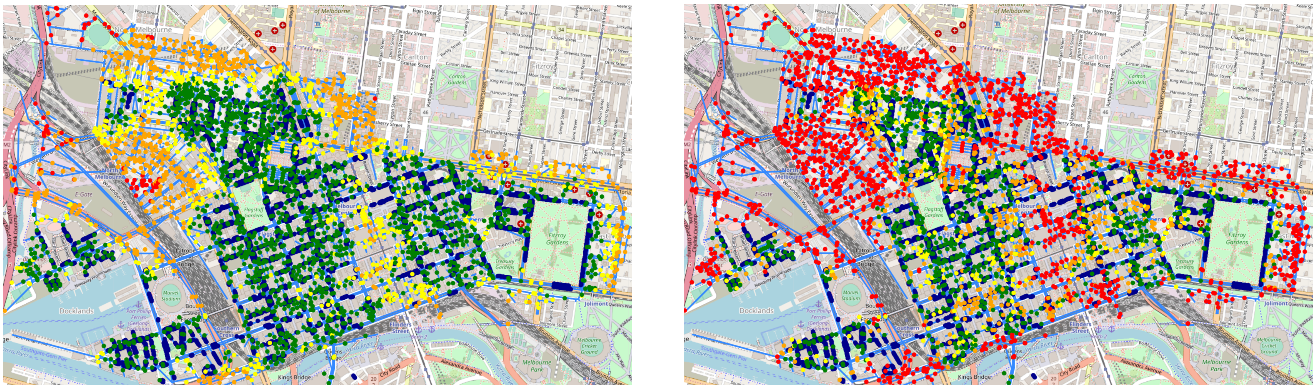
4.4 Results

4.4.1 Hardness of parking. The difference *Baseline* and *Optimal* scenarios indicates the "hardness" of parking with regard to total trip time, i.e. it answers the question "How much time does the optimal parking agent lose compared to arriving by taxi?". A visualization of these differences can be seen in Figure 2a. Quantitative results are summed up over all agent deploy times in Figure 3a. Those results state that taking a taxi does not decrease total trip time by more than one minute approximately half of the time (56%). Interestingly, total trip time is rarely increased by more than five

[4]City of Melbourne open data platform: https://data.melbourne.vic.gov.au; Data licensed under "Creative Commons Attribution 3.0 Australia": https://creativecommons.org/licenses/by/3.0/au/

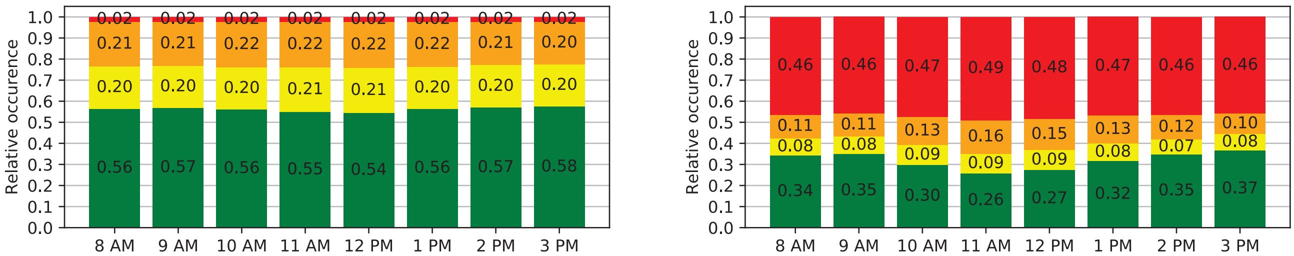
[5]https://data.melbourne.vic.gov.au/Transport-Movement/On-street-Parking-Bays/crvt-b4kt

[6]https://data.melbourne.vic.gov.au/Transport/On-street-Car-Parking-Sensor-Data-2017/u9sa-j86i



(a) Disadvantage of optimal parking agent compared to taking a taxi. (b) Disadvantage of human-like agent compared to optimal agent.

Figure 2: Results for real-world parking occupancy data. Blue dots represent parking bays. A green dot indicates that a trip ended here for which the optimal/human agent did not need more than one minute longer than the taxi/optimal agent. Yellow 1-2, Orange 2-5, Red more than five minutes. All agents in these visualizations started their trips at half past noon.



(a) ***Taxi*** **agent vs.** ***Optimal*** **agent by time of day.** (b) ***Optimal*** **agent vs.** ***Human*** **agent by time of day.**

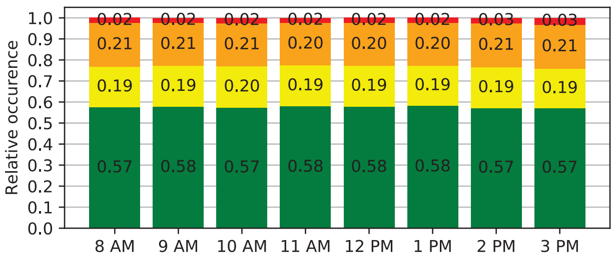
Figure 3: Results for real-world parking occupancy data. The green bars indicate the proportion of trips in which the optimal/taxi agent did not need more than one minute longer than the human/optimal agent. Yellow 1-2, Orange 2-5, Red 5+ min.

minutes. This amounts to the walking time of the bays most far away from the nearest group of parking bays, as represented by the red dots in Figure 2a. A good example of this is the cluster of red dots in the left center. Here are no parking bays (blue dots), so the optimal agent needs much walking time while the taxi can just park at the destination. When parking bays are near, as in the center of the map, the optimal agent generally finds a parking bay with only little extra time when compared to the taxi agent.

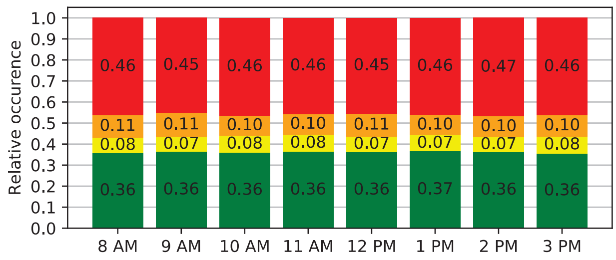
4.4.2 Advantage of optimal agent. The additionally needed time of the human-like agent used in the *Status Quo* scenario when compared to the omniscient agent in the *Optimal* scenario gives an indication of the disadvantage an uninformed human driver may have when compared to an optimal agent. Looking at this time answers the question "How much time can a human driver save using optimal assistance?"–an exemplary visualization can be seen in Figure 2b. This figure shows that the human-like agent suffers especially when no parking bays are near its destination. As it does not know the location of parking bays, it has to search them, thus accumulating a large extra time. But also when parking bays are present in the immediate surroundings, the agent is often not able to find one without some delay. This can be seen in the center where lots of orange dots are present, representing an extra time of two to five minutes when compared to the optimal agent which knows the location and occupancy of parking lots. Figure 2a quantifies these delays: Only approx. 32% of all trips were disadvantaged by less than one minute when compared to the optimal agent.It also becomes apparent that this disadvantage gets worse around noon. This can be explained by the heightened demand for parking bays during this time of day.

5 CASE STUDY WITHOUT REAL-WORLD DATA

In our on-street parking case study presented above, real-world parking data was available for evaluation. Such data is however often not readily available. We argue that in this case, synthetic data can be used. In this section we will give an example of how to generate such data based on only a few parameters for a generative model. Otherwise, our example remains unchanged, i.e. the street graph, agents etc. are not modified.



(a) *Taxi* agent vs. *Optimal* agent by time of day.



(b) *Optimal* agent vs. *Human* agent by time of day.

Figure 4: Results for synthetic parking occupancy data. The green bars indicate the proportion of trips in which the optimal/taxi agent did not need more than one minute longer than the human/optimal agent. Yellow 1-2, Orange 2-5, Red 5+ min.

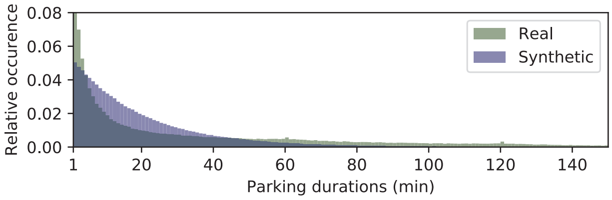


Figure 5: Parking durations of real vs. synthetic dataset.

5.1 Generation of synthetic data

To generate synthetic parking data similar to realistic observations, some knowledge about the behavior of real bays is needed. Often, an average occupation rate ω can be estimated, for example that 70% of parking bays are occupied at a given time. Additionally, a mean parking duration γ can be estimated, such as 45 minutes. In our case, these estimates are based on our experience gained while working with the Melbourne dataset, however they can also be gained by analyzing data from ticket vending machines or recorded trajectories of cars [8]. Estimates ω and γ can now be used to parametrize a model for generating parking events.

One such model for parking events is called Continuous Time Markov Chain (CTMC)[12]. It assumes exponentially distributed sojourn times in each of the two possible states *available* and *occupied* of a single parking bay. The model is quite simple which brings advantages and disadvantages. The advantage is its elegance and the requirement of only two parameters (state change rates λ and μ as described in [12]). The disadvantage is that while availability durations indeed seem to be approximately exponentially distributed, it may be argued that this does not generally apply to occupation durations due to parking time restrictions or other factors. However, Figure 5 shows that while data generated by the model does not perfectly reflect the data from our Melbourne dataset, the general trend does fit, which in our experience is good enough as an approximation. Note that we did not choose the parameters to get the highest possible similarity of resulting distributions as this would not be possible if a real dataset was not available.

To parameterize the CTMC model, we assume to have estimates for the overall occupation rate ω and mean parking duration γ. The CTMC's sojourn time in the state *occupied* ($\frac{1}{\mu}$) is thus already specified by γ. The sojourn time in the state *available* ($\frac{1}{\lambda}$) can be calculated via the model's stationary distribution given in [12]:

$$\pi_1 = \frac{\mu}{\lambda + \mu} \Rightarrow \lambda = \frac{\mu(1 - \pi_1)}{\pi_1} \tag{1}$$

In this equation, π_1 is the stationary probability of the state *available*, for which we know that $\pi_1 = 1 - \omega$. Together with $\mu = \frac{1}{\gamma}$ we arrive at:

$$\lambda = \frac{\mu\omega}{1 - \omega} = \frac{\omega}{\gamma(1 - \omega)} \tag{2}$$

In our example, we use a mean parking duration γ of 45 minutes, thus $\mu = \frac{1}{45\text{min}}$, and an average occupation rate $\omega = 0.7$. Using the equation above, we arrive at $\lambda = \frac{7}{135\text{min}}$. These parameters are then used to sample parking events according the CTMC model, i.e. via its exponential distributions. This process is quite straight-forward: As a parking bay's states *available* and *occupied* alternate, one draws in an alternating way from each exponential distribution to obtain the respective duration, until enough data is generated. This is done for each parking bay. Note that this implies independent behavior of parking bays as given by the model. If different parameters are available for different parking bays (e.g. because they are located in areas with different average parking occupancy), data can of course be sampled from differently parametrized distributions.

5.2 Results

Results obtained using the synthetic dataset can be seen in Figure 4. As the CTMC model for creating these data did not take time-of-day changes into account, fluctuations between the different starting times are minimal–those are caused by the randomness introduced while sampling the data. It should be noted that the human-like agent of the *Status Quo* setting is not deterministic. Therefore, it is also introducing a slight random variation. However, compared to the example with real-world data in Figure 3, it can be seen that the overall results are similar when excluding time-of-day influences. This indicates that synthetic data can in this case be a good substitute when real-world parking data is not available.

It should be noted that the CTMC model's limitation of assuming a constant occupation rate throughout the day can be remedied by using a more complex model. This may be beneficial when a more realistic representation is required, as we have seen in Section 4.4 that this rate does indeed vary depending on the time of day. More complex models covering such time-of-day variations, as the Cyclic

Time-Inhomogeneous Markov Model proposed in [17], however require more parameters, which may not be available.

6 CONCLUSION

In this paper, we present an approach to quantify the cost of necessary detours and the potential of solutions for reducing this cost. In contrast to related literature, we focus on comparisons with the status quo and a best-case scenario instead of just giving absolute numbers of a solution's performance or comparing against competing solutions. Our method can be applied under various preconditions, such as in the real world or using a simulator. Real-world data is advantageous in regard to realism, but synthetic data can also be used for evaluation if it is carefully constructed to resemble distributions observed in the real world. Our approach enables decision makers to better appraise the potential of mobility solutions. We also argue that comparing against the introduced benchmark scenarios can indicate which research or business opportunities are most promising, or which are already "solved" for practical purposes. Our general method is applied to a case study on searching on-street parking opportunities. To quantify the cost of necessary detours caused by searching for a free parking bay, we compare going by taxi (baseline) with having full knowledge of the future (optimal scenario). Results show that taking a taxi saves less than one minute of total trip time in 56% of the simulation runs as the optimal agent is typically able to find a parking space near to the destination. This indicates that if a human driver would be using a park-routing service approaching the performance of the optimal agent, driving your own car would not take much longer than taking a taxi. Secondly, we compared the omniscient agent with a human-like agent to get an insight in the potential for strategies attempting to shorten the time needed for finding a suitable parking bay. Our results indicate that even when free parking bays are not far from the destination (which the omniscient agent knows about), the human agent often needs one to five minutes longer to search for a parking bay and walk from there to the destination. While in reality, it won't be possible to reach the performance of the omniscient agent because complete and correct information about the future is not available, this comparison shows the potential of methods for reducing time spent for parking search. When applying our methodology with synthetic parking occupancy data generated by an artificial model of parking bay behavior, we observe similar simulation results as obtained using real-world data. This indicates that synthetic data can be used as a substitute if no real-world data is available. This broadens the potential application areas of our approach, as real-time data is often not readily available but parameters for suitable generative models can often be estimated or even measured with good confidence.

ACKNOWLEDGMENTS

We thank the City of Melbourne, Australia, for providing the parking datasets used in this paper under Creative Commons license. We also thank the authors of the COMSET simulator[21] for releasing it under MIT license. Further thanks extend to Niklas Strauß for testing and improving our simulator, and Shanki Berger and Andre Koch for providing a cleansed version of the parking event dataset.

REFERENCES

[1] Daniel Ayala, Ouri Wolfson, Bo Xu, Bhaskar Dasgupta, and Jie Lin. 2011. Parking slot assignment games. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 299–308.

[2] Daniel Ayala, Ouri Wolfson, Bo Xu, Bhaskar DasGupta, and Jie Lin. 2012. Pricing of parking for congestion reduction. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*. 43–51.

[3] Sylvain Belloche. 2015. On-street parking search time modelling and validation with survey-based data. *Transportation Research Procedia* 6 (2015), 313–324.

[4] Itzhak Benenson, Karel Martens, and Slava Birfir. 2008. PARKAGENT: An agent-based model of parking in the city. *Computers, Environment and Urban Systems* 32, 6 (2008), 431–439.

[5] Joschka Bischoff and Kai Nagel. 2017. Integrating explicit parking search into a transport simulation. *Procedia computer science* 109 (2017), 881–886.

[6] JL Boehlé, LJM Rothkrantz, and Marcel van Wezel. 2008. Cbprs: A city based parking and routing system. (2008).

[7] Raymond C Browning, Emily A Baker, Jessica A Herron, and Rodger Kram. 2006. Effects of obesity and sex on the energetic cost and preferred speed of walking. *Journal of applied physiology* 100, 2 (2006), 390–398.

[8] Oded Cats, Chen Zhang, and Albania Nissan. 2016. Survey methodology for measuring parking occupancy: Impacts of an on-street parking pricing scheme in an urban center. *Transport Policy* 47 (2016), 55–63.

[9] Blerim Cici, Athina Markopoulou, Enrique Frías-Martínez, and Nikolaos Laoutaris. 2013. Quantifying the potential of ride-sharing using call description records. In *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications*. 1–6.

[10] Tobias Friedrich, Martin S Krejca, Ralf Rothenberger, Tobias Arndt, Danijar Hafner, Thomas Kellermeier, Simon Krogmann, and Armin Razmjou. 2019. Routing for on-street parking search using probabilistic data. *AI Communications* 32, 2 (2019), 113–124.

[11] Yanfeng Geng and Christos G Cassandras. 2013. New smart parking system based on resource allocation and reservations. *IEEE Transactions on intelligent transportation systems* 14, 3 (2013), 1129–1139.

[12] Gregor Jossé, Klaus Arthur Schmid, and Matthias Schubert. 2015. Probabilistic Resource Route Queries with Reappearance.. In *EDBT*, Vol. 15. 445–456.

[13] Wolfgang Kiess and Ashiq Khan. 2014. Centralized vs. distributed: On the placement of gateway functionality in 5G cellular networks. In *2014 IEEE Global Communications Conference*. IEEE, 4788–4793.

[14] Kenneth Lebeau, Joeri Van Mierlo, Philippe Lebeau, Olivier Mairesse, and Cathy Macharis. 2013. Consumer attitudes towards battery electric vehicles: a large-scale survey. *International Journal of Electric and Hybrid Vehicles* 5, 1 (2013), 28–41.

[15] Nadav Levy, Marc Render, and Itzhak Benenson. 2015. Spatially explicit modeling of parking search as a tool for urban parking facilities and policy assessment. *Transport Policy* 39 (2015), 9–20.

[16] Kin Sum Liu, Jie Gao, Xiaobing Wu, and Shan Lin. 2018. On-street parking guidance with real-time sensing data for smart cities. In *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 1–9.

[17] Lukas Rottkamp and Matthias Schubert. 2018. A time-inhomogeneous markov model for resource availability under sparse observations. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 460–463.

[18] Donald C Shoup. 2006. Cruising for parking. *Transport Policy* 13, 6 (2006), 479–486.

[19] Thérèse Steenberghen, Karel Dieussaert, Sven Maerivoet, and Karel Spitaels. 2012. SUSTAPARK: An Agent-based Model for Simulating Parking Search. *Journal of the Urban & Regional Information Systems Association* 24, 1 (2012).

[20] Kamila Turečková and Jan Nevima. 2020. The Cost Benefit Analysis for the Concept of a Smart City: How to Measure the Efficiency of Smart Solutions? *Sustainability* 12, 7 (2020), 2663.

[21] Robert van Barlingen, João Ferreira, Jeroen Klimovic, Tijana Schols, Wouter de Vries, and Bo Xu. 2020. *COMSET*. https://github.com/Chessnl/COMSET-GISCUP

[22] Rashid A Waraich, Christoph Dobler, and Kay W Axhausen. 2012. Modelling parking search behaviour with an agent-based approach. In *13th International Conference on Travel Behaviour Research (IATBR 2012)*. IVT, ETH Zurich.

[23] BJ Waterson, NB Hounsell, and Kiron Chatterjee. 2001. Quantifying the potential savings in travel time resulting from parking guidance systems—a simulation case study. *Journal of the Operational Research Society* 52, 10 (2001), 1067–1077.

[24] Xianyuan Zhan, Xinwu Qian, and Satish V Ukkusuri. 2016. A graph-based approach to measuring the efficiency of an urban taxi service system. *IEEE Transactions on Intelligent Transportation Systems* 17, 9 (2016), 2479–2489.

[25] Xiaolu Zhang, Demin Li, Jiacun Wang, Guanglin Zhang, and Xiaoyin Jiang. 2016. Faster parking and less cruise for public parking spot discovery: Modeling and analysis based on timed petri nets. In *2016 IEEE 13th International Conference on Networking, Sensing, and Control (ICNSC)*. IEEE, 1–6.

Chapter 9

Concluding Remarks

This chapter gives a summary of contributions and indicates directions for future research.

9.1 Summary

This thesis highlights four propositions regarding mobility problems: 1) Mobility tasks such as the parking search problem or the ambulance redeployment problem can be addressed by developing computer systems that recommend actions based on a known or estimated environment state. 2) Data about the environment state can be obtained by sensors. 3) Post-processing of obtained data can improve data quality in case of imperfections such as sensor gaps. 4) The effects of different data qualities should be determined and compared in order to decide on implementation details such as the number of sensors to be installed.

Advances in each of these four fields potentially increase the overall quality of the eventual actions or plans recommended by the computer system. Therefore, each is covered in this thesis: In the first field, a variation of the resource routing problem is introduced: A fleet of agents with individual resource routing tasks improves travel times by sharing data obtained by the same agents while passing resources. The problem is formalized and methods are given to solve it in various settings, including a partially-observable setting in which no stationary sensors are present. The thesis also considers the ambulance redeployment problem for electric ambulances by introducing the dynamic electric ambulance redeployment problem (DEAR). A method to solve the problem is given and evaluated against existing solutions for the non-electric case. The second field, obtaining data, is especially interesting when not all resources can be observed by sensors due to budget or other restrictions. This thesis therefore compares existing data-driven and data-agnostic sensor placement methods for a limited number of stationary sensors and presents a method for placing sensors more cost efficiently. In the third field, improving data, the estimation of current or future resource states is especially interesting if only sparse measurements are available. Here, this thesis presents a way to obtain estimates by exploiting the cyclic nature of state change dynamics usually present in environmental data such as parking availability data. Finally, the thesis underlines the importance of evaluating and comparing different data qualities with regard to the user-observable quality of the mobility solution. This is beneficial when different data qualities come with different cost, e.g., additional sensors increase data quality but require further investment. Knowing the effects of different methods for obtaining, improving and using data enables decision makers to implement cost-effective solutions to mobility problems.

9.2 Outlook

Solving real-world mobility problems is difficult because of the complex dynamic environment that includes numerous independent actors, side effects and external influences. Therefore, many opportunities for future work on topics covered in this thesis exist: First, methods for solving mobility problems necessarily include assumptions or abstractions of the real world. For example, approaches presented in this thesis assume that traffic flow speed is not affected by actions taken. This may lead to inaccuracies in some instances, e.g., simultaneously routing multiple vehicles through a narrow street can actually cause congestion and thus increase average driving time. Future work should quantify the effects of such feedback loops and improve methods if necessary. External influences, e.g., current or predicted weather, can often be integrated into existing methods in a straightforward way. It seems promising to determine the significance of various external effects and include relevant contributors. Further, methods can be improved by integrating anomalies such as temporary street closures or sudden spikes in demand caused by events such as large concerts. The electric ambulance location problem introduced in this thesis assumes a fixed number of ambulances. As ambulance demand varies throughout the day, future work could improve the state of the art by additionally planning crew schedules. This is especially relevant for electric ambulances due to their charging time requirements.

Regarding real-time data, future work may evaluate heterogeneous sensor networks, especially the combination of stationary and mobile sensors, as benefiting from their respective advantages may increase efficiency. Further, approaches given in this thesis assume accurate sensor readings. In real-world deployments, sensors can fail and certain environmental conditions may lead to elevated error rates. Extensions considering such sensor anomalies may provide more robust solutions. This thesis discusses how the benefit of certain data can be determined with regard to a mobility application. A next step would be to assign a monetary value to data points, e.g., by determining users' willingness to pay for a certain benefit. Knowing the actual value of data may enable a more efficient placement of sensors and facilitate the trade of data, resulting in improved service quality.

Bibliography

[1] Carole A Aldrich, John C Hisserich, and Lester B Lave. "An analysis of the demand for emergency ambulance service in an urban area." In: *American Journal of Public Health* 61.6 (1971), pp. 1156–1169. DOI: 10.2105/AJPH.61.6.1156.

[2] Joshua S Apte et al. "High-resolution air pollution mapping with Google street view cars: exploiting big data". In: *Environmental science & technology* 51.12 (2017), pp. 6999–7008. DOI: 10.1021/acs.est.7b00891.

[3] Tobias Arndt et al. "Probabilistic routing for on-street parking search". In: *24th Annual European Symposium on Algorithms (ESA 2016)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik. 2016. DOI: 10.4230/LIPIcs.ESA.2016.6.

[4] Richard Arnott and Kenneth Small. "The economics of traffic congestion". In: *American scientist* 82.5 (1994), pp. 446–455.

[5] Kay W Axhausen, John W Polak, and Manfred Boltze. "Effectiveness of parking guidance and information systems: recent evidence from Nottingham and Frankfurt am Main". In: *63rd Annual Meeting: Compendium of Technical Papers*. Vol. 37. Institute of Transportation Engineers. 1993. DOI: 10.3929/ethz-b-000024872.

[6] Davide Bacciu. "Unsupervised feature selection for sensor time-series in pervasive computing applications". In: *Neural Computing and Applications* 27 (2016), pp. 1077–1091. DOI: 10.1007/s00521-015-1924-x.

[7] Antoine Bagula, Lorenzo Castelli, and Marco Zennaro. "On the design of smart parking networks in the smart cities: An optimal sensor placement model". In: *Sensors* 15.7 (2015), pp. 15443–15467. DOI: 10.3390/s150715443.

[8] Rajan Batta, June M Dolan, and Nirup N Krishnamurthy. "The maximal expected covering location problem: Revisited". In: *Transportation science* 23.4 (1989), pp. 277–287. DOI: 10.1287/trsc.23.4.277.

[9] Itzhak Benenson, Karel Martens, and Slava Birfir. "PARKAGENT: An agent-based model of parking in the city". In: *Computers, Environment and Urban Systems* 32.6 (2008), pp. 431–439. DOI: 10.1016/j.compenvurbsys.2008.09.011.

[10] Fabian Bock, Sergio Di Martino, and Antonio Origlia. "A 2-Step Approach to Improve Data-driven Parking Availability Predictions". In: *Proceedings of the 10th ACM SIGSPATIAL Workshop on Computational Transportation Science*. ACM. 2017, pp. 13–18. DOI: 10.1145/3151547.3151550.

[11] Fabian Bock, Sergio Di Martino, and Antonio Origlia. "Smart parking: Using a crowd of taxis to sense on-street parking space availability". In: *IEEE Transactions on Intelligent Transportation Systems* 21.2 (2019), pp. 496–508. DOI: 10.1109/TITS.2019.2899149.

[12] Fabian Bock, Jiaqi Liu, and Monika Sester. "Learning on-street parking maps from position information of parked vehicles". In: *Geospatial Data in a Changing World: Selected papers of the 19th AGILE Conference on Geographic Information Science*. Springer. 2016, pp. 297–314. DOI: 10.1007/978-3-319-33783-8_17.

[13] Fabian Bock and Monika Sester. "Improving parking availability maps using information from nearby roads". In: *Transportation Research Procedia* 19 (2016), pp. 207–214. DOI: 10.1016/j.trpro.2016.12.081.

[14] Geoff Boeing. "OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks". In: *Computers, Environment and Urban Systems* 65 (2017), pp. 126–139. DOI: 10.1016/j.compenvurbsys.2017.05.004.

[15] Luce Brotcorne, Gilbert Laporte, and Frederic Semet. "Ambulance location and relocation models". In: *European journal of operational research* 147.3 (2003), pp. 451–463. DOI: 10.1016/S0377-2217(02)00364-8.

[16] Andreas Bürger et al. "The effect of ambulance response time on survival following out-of-hospital cardiac arrest: an analysis from the German resuscitation registry". In: *Deutsches Ärzteblatt International* 115.33-34 (2018), p. 541. DOI: 10.3238/arztebl.2018.0541.

[17] Eric Lucas dos Santos Cabral et al. "Response time in the emergency services. Systematic review". In: *Acta cirurgica brasileira* 33.12 (2018), pp. 1110–1121. DOI: 10.1590/s0102-865020180120000009.

[18] Murat Caliskan, Andreas Barthels, Bjorn Scheuermann, and Martin Mauve. "Predicting parking lot occupancy in vehicular ad hoc networks". In: *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*. IEEE. 2007, pp. 277–281. DOI: 10.1109/VETECS.2007.69.

[19] Murat Caliskan, Daniel Graupner, and Martin Mauve. "Decentralized discovery of free parking places". In: *VANET - Proceedings of the Third ACM International Workshop on Vehicular Ad Hoc Networks* 2006 (2006), pp. 30–39. DOI: 10.1145/1161064.1161070.

[20] Huajun Chai, Rui Ma, and H Michael Zhang. "Search for parking: A dynamic parking and route guidance system for efficient parking and traffic management". In: *Journal of Intelligent Transportation Systems* 23.6 (2019), pp. 541–556. DOI: 10.1080/15472450.2018.1488218.

[21] Richard Church and Charles R Velle. "The maximal covering location problem". In: *Papers in regional science* 32.1 (1974), pp. 101–118. DOI: 10.1111/j.1435-5597.1974.tb00902.x.

[22] Blerim Cici, Athina Markopoulou, Enrique Frías-Martínez, and Nikolaos Laoutaris. "Quantifying the potential of ride-sharing using call description records". In: *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications*. 2013, pp. 1–6. DOI: 10.1145/2444776.2444799.

[23] Intergovernmental Panel on Climate Change (IPCC). *Climate Change 2021 – The Physical Science Basis: Working Group I Contribution to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, 2023. DOI: 10.1017/9781009157896.

[24] Mark S Daskin. "A maximum expected covering location model: formulation, properties and heuristic solution". In: *Transportation science* 17.1 (1983), pp. 48–70. DOI: 10.1287/trsc.17.1.48.

[25] Mark S Daskin and Edmund H Stern. "A hierarchical objective set covering model for emergency medical service vehicle deployment". In: *Transportation Science* 15.2 (1981), pp. 137–152. DOI: 10.1287/trsc.15.2.137.

[26] Stephen F Dean. "Why the closest ambulance cannot be dispatched in an urban emergency medical services system". In: *Prehospital and disaster medicine* 23.2 (2008), pp. 161–165. DOI: 10.1017/S1049023X00005793.

[27] Thierry Delot, Nicolas Cenerario, Sergio Ilarri, and Sylvain Lecomte. "A cooperative reservation protocol for parking spaces in vehicular ad hoc networks". In: *Proceedings of the 6th International Conference on Mobile Technology, Application & Systems*. 2009, pp. 1–8. DOI: 10.1145/1710035.1710065.

[28] Edsger W Dijkstra. "A note on two problems in connexion with graphs". In: *Edsger Wybe Dijkstra: his life, work, and legacy*. 2022, pp. 287–290. DOI: 10.1145/3544585.3544600.

[29] Chase Dowling, Tanner Fiez, Lillian Ratliff, and Baosen Zhang. "How much urban traffic is searching for parking". In: *arXiv preprint arXiv:1702.06156* (2017), pp. 1–20. DOI: 10.48550/arXiv.1702.06156.

[30] Marc Eckstein and Linda S Chan. "The effect of emergency department crowding on paramedic ambulance availability". In: *Annals of emergency medicine* 43.1 (2004), pp. 100–105. DOI: 10.1016/S0196-0644(03)00747-9.

[31] Shakiba Enayati, Maria E Mayorga, Hari K Rajagopalan, and Cem Saydam. "Realtime ambulance redeployment approach to improve service coverage with fair and restricted workload for EMS providers". In: *Omega* 79 (2018), pp. 67–80. DOI: 10.1016/j.omega.2017.08.001.

[32] Bradley M Estochen, Reginald R Souleyrette, and Tim Strauss. *An assessment of emergency response vehicle pre-deployment using gis identification of high-accident density locations*. Center for Transportation Research and Education, Iowa State University, 1998.

[33] Ross J Fleischman et al. “Predicting ambulance time of arrival to the emergency department using global positioning system and Google maps”. In: *Prehospital Emergency Care* 17.4 (2013), pp. 458–465. DOI: 10.3109/10903127.2013.811562.

[34] Jeffrey Goldberg et al. “A simulation model for evaluating a set of emergency vehicle base locations: Development, validation, and usage”. In: *Socio-economic planning sciences* 24.2 (1990), pp. 125–141. DOI: 10.1016/0038-0121(90)90017-2.

[35] Giulio Grassi, Kyle Jamieson, Paramvir Bahl, and Giovanni Pau. “Parkmaster: An in-vehicle, edge-based video analytics service for detecting open parking spaces in urban environments”. In: *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*. 2017, pp. 1–14. DOI: 10.1145/3132211.3134452.

[36] Chenjuan Guo et al. “Ecosky: Reducing vehicular environmental impact through eco-routing”. In: *2015 IEEE 31st International Conference on Data Engineering*. IEEE. 2015, pp. 1412–1415. DOI: 10.1109/ICDE.2015.7113389.

[37] Karla L Hoffman, Manfred Padberg, Giovanni Rinaldi, et al. “Traveling salesman problem”. In: *Encyclopedia of operations research and management science* 1 (2013), pp. 1573–1578. DOI: 10.1007/978-1-4419-1153-7_1068.

[38] Sebastian Houben et al. “On-vehicle video-based parking lot recognition with fish-eye optics”. In: *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE. 2013, pp. 7–12. DOI: 10.1109/ITSC.2013.6728595.

[39] Mohd Yamani Idna Idris et al. “Car park system: A review of smart parking system and its technology”. In: *Information Technology Journal* 8.2 (2009), pp. 101–113. DOI: 10.3923/itj.2009.101.113.

[40] Caroline Jagtenberg, Sandjai Bhulai, and Rob van der Mei. “An efficient heuristic for real-time ambulance redeployment”. In: *Operations Research for Health Care* 4 (2015), pp. 27–35. DOI: 10.1016/j.orhc.2015.01.001.

[41] Caroline Jagtenberg, Sandjai Bhulai, and Rob van der Mei. “Optimal ambulance dispatching”. In: *Markov decision processes in practice* (2017), pp. 269–291. DOI: 10.1007/978-3-319-47766-4_9.

[42] SACS Jayasooriya and Yapa Mahinda Bandara. “Measuring the Economic costs of traffic congestion”. In: *2017 Moratuwa Engineering Research Conference (MERCon)*. IEEE. 2017, pp. 141–146. DOI: 10.1109/MERCon.2017.7980471.

[43] Shenggong Ji, Yu Zheng, Zhaoyuan Wang, and Tianrui Li. “A deep reinforcement learning-enabled dynamic redeployment system for mobile ambulances”. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3.1 (2019), pp. 1–20. DOI: 10.1145/3314402.

[44] Yijuan Jiang and Xiang Li. “Travel time prediction based on historical trajectory data”. In: *Annals of GIS* 19.1 (2013), pp. 27–35. DOI: 10.1080/19475683.2012.758173.

[45] Gregor Jossé, Klaus Arthur Schmid, and Matthias Schubert. “Probabilistic Resource Route Queries with Reappearance.” In: *EDBT*. Vol. 15. 2015, pp. 445–456. DOI: 10.5441/002/edbt.2015.39.

[46] Ricardo D Kamenetzky, Larry J Shuman, and Harvey Wolfe. “Estimating need and demand for prehospital care”. In: *Operations research* 30.6 (1982), pp. 1148–1167. DOI: 10.1287/opre.30.6.1148.

[47] Nick Klausner, Mahmood R Azimi-Sadjadi, and Louis L Scharf. “Detection of spatially correlated time series from a network of sensor arrays”. In: *IEEE transactions on signal processing* 62.6 (2014), pp. 1396–1407. DOI: 10.1109/TSP.2014.2298833.

[48] Olive C Kobusingye et al. “Emergency medical services”. In: *Disease Control Priorities in Developing Countries. 2nd edition* (2006).

[49] Andreas Krause, Ajit Singh, and Carlos Guestrin. “Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies.” In: *Journal of Machine Learning Research* 9.2 (2008).

[50] Jan Karel Lenstra and AHG Rinnooy Kan. “Complexity of vehicle routing and scheduling problems”. In: *Networks* 11.2 (1981), pp. 221–227. DOI: 10.1002/net.3230110211.

[51] Jan Karel Lenstra and AHG Rinnooy Kan. “Some simple applications of the travelling salesman problem”. In: *Journal of the Operational Research Society* 26.4 (1975), pp. 717–733. DOI: 10.1057/jors.1975.151.

[52] Ilias Leontiadis and Cecilia Mascolo. “Opportunistic spatio-temporal dissemination system for vehicular networks”. In: *Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking*. 2007, pp. 39–46. DOI: 10.1145/1247694.1247702.

[53] Mengyu Li et al. “Determining ambulance destinations when facing offload delays using a Markov decision process”. In: *Omega* 101 (2021), p. 102251. DOI: 10.1016/j.omega.2020.102251.

[54] Adrian Xi Lin et al. “Leveraging machine learning techniques and engineering of multi-nature features for national daily regional ambulance demand prediction”. In: *International journal of environmental research and public health* 17.11 (2020), p. 4179. DOI: 10.3390/ijerph17114179.

[55] Kin Sum Liu, Jie Gao, Xiaobing Wu, and Shan Lin. “On-street parking guidance with real-time sensing data for smart cities”. In: *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE. 2018, pp. 1–9. DOI: 10.1109/SAHCN.2018.8397113.

[56] Junsheng Ma et al. “A Bayesian Approach in Estimating Transition Probabilities of a Discrete-time Markov Chain for Ignorable Intermittent Missing Data”. In: *Communications in Statistics-Simulation and Computation* 45.7 (2016), pp. 2598–2616. DOI: 10.1080/03610918.2014.911895.

[57] Shuo Ma, Ouri Wolfson, and Bo Xu. "UPDetector: Sensing parking/unparking activities using smartphones". In: *Proceedings of the 7th ACM SIGSPATIAL international workshop on computational transportation science*. ACM. 2014, pp. 76–85. DOI: 10.1145/2674918.2674929.

[58] Krithika Manohar, Bingni W Brunton, J Nathan Kutz, and Steven L Brunton. "Data-driven sparse sensor placement for reconstruction: Demonstrating the benefits of exploiting known patterns". In: *IEEE Control Systems Magazine* 38.3 (2018), pp. 63–86. DOI: 10.1109/MCS.2018.2810460.

[59] Vladimir Marianov and Charles Revelle. "The queuing probabilistic location set covering problem and some extensions". In: *Socio-Economic Planning Sciences* 28.3 (1994), pp. 167–178. DOI: 10.1016/0038-0121(94)90003-5.

[60] Samir El-Masri and Basema Saddik. "An emergency system to improve ambulance dispatching, ambulance diversion and clinical handover communication—A proposed model". In: *Journal of medical systems* 36 (2012), pp. 3917–3923. DOI: 10.1007/s10916-012-9863-x.

[61] Suhas Mathur et al. "Parknet: drive-by sensing of road-side parking statistics". In: *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM. 2010, pp. 123–136. DOI: 10.1145/1814433.1814448.

[62] Matthew S Maxwell, Mateo Restrepo, Shane G Henderson, and Huseyin Topaloglu. "Approximate dynamic programming for ambulance redeployment". In: *INFORMS Journal on Computing* 22.2 (2010), pp. 266–281. DOI: 10.1287/ijoc.1090.0345.

[63] Laura A McLay and Maria E Mayorga. "Evaluating emergency medical service performance measures". In: *Health care management science* 13 (2010), pp. 124–136. DOI: 10.1007/s10729-009-9115-x.

[64] Lubos Mitas and Helena Mitasova. "Spatial interpolation". In: *Geographical information systems: principles, techniques, management and applications* 1.2 (1999), pp. 481–492.

[65] Mohamed Mokbel et al. "Mobility Data Science: Perspectives and Challenges". In: *ACM Transactions on Spatial Algorithms and Systems* (2024). DOI: 10.1145/3652158.

[66] Donald E Myers. "Spatial interpolation: an overview". In: *Geoderma* 62.1-3 (1994), pp. 17–28. DOI: 10.1016/0016-7061(94)90025-6.

[67] S Deepak Narayanan, Zeel B Patel, Apoorv Agnihotri, and Nipun Batra. "A toolkit for spatial interpolation and sensor placement". In: *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. 2020, pp. 653–654. DOI: 10.1145/3384419.3430407.

[68] Brian K Nelson. "Time series analysis using autoregressive integrated moving average (ARIMA) models". In: *Academic emergency medicine* 5.7 (1998), pp. 739–744. DOI: 10.1111/j.1553-2712.1998.tb02493.x.

[69] Marcus EH Ong et al. “Geographic-time distribution of ambulance calls in Singapore: utility of geographic information system in ambulance deployment (CARE 3)”. In: *Annals Academy of Medicine Singapore* 38.3 (2009), p. 184. DOI: 10.47102/annals-acadmedsg.v38n3p184.

[70] Sharon L Padula and Rex K Kincaid. *Optimization strategies for sensor and actuator placement*. Tech. rep. 1999.

[71] Jill P Pell et al. “Effect of reducing ambulance response times on deaths from out of hospital cardiac arrest: cohort study”. In: *Bmj* 322.7299 (2001), pp. 1385–1388. DOI: 10.1136/bmj.322.7299.1385.

[72] Sherisha Pullola, Pradeep K Atrey, and Abdulmotaleb El Saddik. “Towards an intelligent GPS-based vehicle navigation system for finding street parking lots”. In: *2007 IEEE International Conference on Signal Processing and Communications*. IEEE. 2007, pp. 1251–1254. DOI: 10.1109/ICSPC.2007.4728553.

[73] Hari K Rajagopalan, Cem Saydam, and Jing Xiao. “A multiperiod set covering location model for dynamic redeployment of ambulances”. In: *Computers & Operations Research* 35.3 (2008), pp. 814–826. DOI: 10.1016/j.cor.2006.04.003.

[74] John F Repede and John J Bernardo. “Developing and validating a decision support system for locating emergency medical vehicles in Louisville, Kentucky”. In: *European journal of operational research* 75.3 (1994), pp. 567–581. DOI: 10.1016/0377-2217(94)90297-6.

[75] Charles ReVelle and Kathleen Hogan. “The maximum availability location problem”. In: *Transportation science* 23.3 (1989), pp. 192–200. DOI: 10.1287/trsc.23.3.192.

[76] Sheldon M Ross. *Introduction to probability models*. Academic press, 2014.

[77] Lukas Rottkamp and Matthias Schubert. “A Time-Inhomogeneous Markov Model for Resource Availability under Sparse Observations”. In: *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. SIGSPATIAL ’18. Seattle, Washington: Association for Computing Machinery, 2018, pp. 460–463. ISBN: 9781450358897. DOI: 10.1145/3274895.3274945.

[78] Lukas Rottkamp and Matthias Schubert. “A Time-Inhomogeneous Markov Model for Resource Availability under Sparse Observations”. In: *arXiv* (2024). DOI: 10.48550/arXiv.2404.12240. URL: https://doi.org/10.48550/arXiv.2404.12240.

[79] Lukas Rottkamp and Matthias Schubert. “Quantifying the Potential of Data-Driven Mobility Support Systems”. In: *Proceedings of the 13th ACM SIGSPATIAL International Workshop on Computational Transportation Science*. IWCTS ’20. Seattle, Washington: Association for Computing Machinery, 2020. ISBN: 9781450381666. DOI: 10.1145/3423457.3429366.

[80] Lukas Rottkamp, Matthias Schubert, and Niklas Strauß. "Efficient On-Street Parking Sensor Placement". In: *Proceedings of the 15th ACM SIGSPATIAL International Workshop on Computational Transportation Science*. IWCTS '22. Seattle, Washington: Association for Computing Machinery, 2022. ISBN: 9781450395397. DOI: 10.1145/3557991.3567796.

[81] Lukas Rottkamp, Niklas Strauß, and Matthias Schubert. "DEAR: Dynamic Electric Ambulance Redeployment". In: *Proceedings of the 18th International Symposium on Spatial and Temporal Data*. SSTD '23. Calgary, AB, Canada: Association for Computing Machinery, 2023, pp. 11–20. ISBN: 9798400708992. DOI: 10.1145/3609956.3609959.

[82] Sebastian Schmoll, Sabrina Friedl, and Matthias Schubert. "Scaling the Dynamic Resource Routing Problem". In: *Proceedings of the 16th International Symposium on Spatial and Temporal Databases*. 2019, pp. 80–89. DOI: 10.1145/3340964.3340983.

[83] Sebastian Schmoll and Matthias Schubert. "Dynamic resource routing using real-time dynamic programming". In: *IJCAI International Joint Conference on Artificial Intelligence* 2018-July.Vi (2018), pp. 4822–4828. ISSN: 10450823. DOI: 10.24963/ijcai.2018/670.

[84] Sebastian Schmoll and Matthias Schubert. "Dynamic resource routing using real-time information". In: *Advances in Database Technology - EDBT* 2018-March (2018), pp. 501–504. ISSN: 23672005. DOI: 10.5441/002/edbt.2018.57.

[85] Hubert Setzler, Cem Saydam, and Sungjune Park. "EMS call volume predictions: A comparative study". In: *Computers & Operations Research* 36.6 (2009), pp. 1843–1851. DOI: 10.1016/j.cor.2008.05.010.

[86] Xavier Sevillano, Elena Màrmol, and Virginia Fernandez-Arguedas. "Towards smart traffic management systems: Vacant on-street parking spot detection based on video analytics". In: *17th International Conference on Information Fusion (FUSION)*. IEEE. 2014, pp. 1–8.

[87] Donald Shepard. "A two-dimensional interpolation function for irregularly-spaced data". In: *Proceedings of the 1968 23rd ACM national conference*. 1968, pp. 517–524. DOI: 10.1145/800186.810616.

[88] Donald C Shoup. "Cruising for parking". In: *Transport Policy* 13.6 (2006), pp. 479–486. ISSN: 0967070X. DOI: 10.1016/j.tranpol.2006.05.005.

[89] Brian M de Silva et al. "PySensors: A Python package for sparse sensor placement". In: *arXiv preprint arXiv:2102.13476* (2021). DOI: 10.48550/arXiv.2102.13476.

[90] Thérèse Steenberghen, Karel Dieussaert, Sven Maerivoet, and Karel Spitaels. "Sustapark: An agent-based model for simulating parking search". In: *URISA Journal* 24.1 (2012), pp. 63–76. ISSN: 10458077.

[91] Krisjanis Steins, Niki Matinrad, and Tobias Granberg. “Forecasting the demand for emergency medical services”. In: *Proceedings of the Annual Hawaii International Conference on System Sciences*. Hawaii International Conference on System Sciences. 2019.

[92] Niklas Strauß, Max Berrendorf, Tom Haider, and Matthias Schubert. “A Comparison of Ambulance Redeployment Systems on Real-World Data”. In: *Proceedings of the 1st Workshop on Urban Internet-of-Things Intelligence (UNIT 2022) co-located with the 22nd IEEE International Conference on Data Mining (ICDM 2022)*. 2022. DOI: 10.1109/ICDMW58026.2022.00010.

[93] Niklas Strauß, Lukas Rottkamp, Sebastian Schmoll, and Matthias Schubert. “Efficient Parking Search using Shared Fleet Data”. In: *2021 22nd IEEE International Conference on Mobile Data Management (MDM)*. IEEE, 2021, pp. 115–120. ISBN: 9781665428453. DOI: 10.1109/MDM52706.2021.00026.

[94] Niklas Strauß, Lukas Rottkamp, Sebastian Schmoll, and Matthias Schubert. “Efficient Parking Search using Shared Fleet Data”. In: *arXiv* (2024). DOI: 10.48550/arXiv.2404.10646. URL: https://doi.org/10.48550/arXiv.2404.10646.

[95] Chenxi Sun, Victor OK Li, Jacqueline CK Lam, and Ian Leslie. “Optimal citizen-centric sensor placement for air quality monitoring: a case study of city of Cambridge, the United Kingdom”. In: *IEEE Access* 7 (2019), pp. 47390–47400. DOI: 10.1109/ACCESS.2019.2909111.

[96] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction.* MIT press, 2018.

[97] Geert Tasseron, Karel Martens, and Rob van der Heijden. “The potential impact of vehicle-to-vehicle and sensor-to-vehicle communication in urban parking”. In: *IEEE Intelligent Transportation Systems Magazine* 7.2 (2015), pp. 22–33. DOI: 10.1109/MITS.2015.2390918.

[98] Yogesh Tayade and MD Patil. “Advance Prediction of Parking Space Availability and Other Facilities for Car Parks in Smart Cities”. In: *International Research Journal of Engineering and Technology* 3.5 (2016), pp. 2225–2228.

[99] Constantine Toregas, Ralph Swain, Charles ReVelle, and Lawrence Bergman. “The location of emergency service facilities”. In: *Operations research* 19.6 (1971), pp. 1363–1373. DOI: 10.1287/opre.19.6.1363.

[100] Kamila Turečková and Jan Nevima. “The Cost Benefit Analysis for the Concept of a Smart City: How to Measure the Efficiency of Smart Solutions?” In: *Sustainability* 12.7 (2020), p. 2663. DOI: 10.3390/su12072663.

[101] Thije Van Barneveld, Sandjai Bhulai, and Rob van der Mei. “A dynamic ambulance management model for rural areas: Computing redeployment actions for relevant performance measures”. In: *Health care management science* 20 (2017), pp. 165–186. DOI: 10.1007/s10729-015-9341-3.

[102] Zhaonan Wang et al. “Forecasting ambulance demand with profiled human mobility via heterogeneous multi-graph neural networks”. In: *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE. 2021, pp. 1751–1762. DOI: 10.1109/ICDE51399.2021.00154.

[103] Ben Waterson, Nick Hounsell, and Kiron Chatterjee. “Quantifying the potential savings in travel time resulting from parking guidance systems—a simulation case study”. In: *Journal of the Operational Research Society* 52.10 (2001), pp. 1067–1077. DOI: 10.1057/palgrave.jors.2601207.

[104] Ho-Ting Wong and Poh-Chin Lai. “Weather factors in the short-term forecasting of daily ambulance calls”. In: *International journal of biometeorology* 58 (2014), pp. 669–678. DOI: 10.1007/s00484-013-0647-x.

[105] Xiaofei Ye et al. “Short-term prediction of available parking space based on machine learning approaches”. In: *IEEE Access* 8 (2020), pp. 174530–174541. DOI: 10.1109/ACCESS.2020.3025589.

[106] Hung-Wen Yeh, Wenyaw Chan, Elaine Symanski, and Barry R Davis. “Estimating transition probabilities for ignorable intermittent missing data in a discrete-time Markov chain”. In: *Communications in Statistics-Simulation and Computation* 39.2 (2010), pp. 433–448. DOI: 10.1080/03610910903480800.

[107] Sung Wook Yoon, Alan Fern, Robert Givan, and Subbarao Kambhampati. “Probabilistic planning via determinization in hindsight.” In: *AAAI*. 2008, pp. 1010–1016. DOI: 10.5555/1620163.1620229.

[108] Jing Yuan et al. “T-drive: driving directions based on taxi trajectories”. In: *Proceedings of the 18th SIGSPATIAL International conference on advances in geographic information systems*. 2010, pp. 99–108. DOI: 10.1145/1869790.1869807.

[109] Yanxu Zheng, Sutharshan Rajasegarar, and Christopher Leckie. “Parking availability prediction for sensor-enabled car parks in smart cities”. In: *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2015 IEEE Tenth International Conference on*. IEEE. 2015, pp. 1–6. DOI: 10.1109/ISSNIP.2015.7106902.

[110] Hanwei Zhu, Sid Chi-Kin Chau, Gladhi Guarddin, and Weifa Liang. “Integrating IoT-sensing and crowdsensing with privacy: Privacy-preserving hybrid sensing for smart cities”. In: *ACM Transactions on Internet of Things* 3.4 (2022), pp. 1–30. DOI: 10.1145/3549550.

[111] Carol Zimmerman et al. *San Francisco urban partnership agreement: national evaluation report*. Tech. rep. United States. Department of Transportation. Intelligent Transportation Systems Joint Program Office, 2014.

Acknowledgements

The idea of writing a thesis on mobility problems came to life in 2016 during my work at Audi. While working on mobility applications processing spatio-temporal data, the need for further research soon became obvious. I am thankful that my colleagues and managers welcomed my ambition to carry out this research as a PhD project in cooperation with LMU Munich, and shared their experience and advice. I am especially grateful to Gerhard Stanzl, my manager at the time, for his trusting approval of this endeavor.

The greatest thanks extend to Matthias Schubert, who helped to shape the initial research idea, agreed to supervise me during this thesis, facilitated organizational matters, and always found time to engage with my research ideas, questions and drafts. His knowledge and precise feedback were instrumental in improving my publications and this thesis. Further thanks to him, Niklas Strauß and Sebastian Schmoll for their discussions and contributions as co-authors of publications presented in this thesis. I am also thankful to Dimitris Sacharidis and Kristian Torp who agreed to act as reviewers of this thesis.

Without attempting an exhaustive enumeration of names, I thank everyone who helped with organizational matters, engaged in scientific discussion, suggested improvements or provided encouragement. Standing on the shoulders of giants, I hope the contributions presented in this thesis improve the quality of future mobility solutions.