TUM School of Management
Technische Universität München

Balanced Electric Vehicle Charging under Uncertainty

From User-Centric to System-Centric Approaches

Marianne Marie-Pierre Guillet

Vollständiger Abdruck der von der TUM School of Management der Technischen Universität München zur Erlangung des akademischen Grades einer Doktorin der Wirtschafts- und Sozialwissenschaften (Dr. rer. pol.) genehmigten Dissertation.

Vorsitz : Prof. Dr. Gudrun Kiesmüller

Prüfende der Dissertation:

1. Prof. Dr. Maximilian Schiffer
2. Prof. Dr. Dario Paccagnan

Die Dissertation wurde am 06.09.2022 bei der Technischen Universität München eingereicht und durch die TUM School of Management am 15.11.2022 angenommen.

Contents

Acknowledgments

I would like to first sincerely thank my supervisor Prof. Maximilian Schiffer for his guidance, trust, patience, and considerable support over the past years. This thesis would have never taken on its final shape without his broad knowledge of our research field and highly-valuable feedback during our countless discussions. Moreover, I would like to thank Prof. Dr. Gudrun P. Kiesmüller and Prof. Dr. Dario Paccagnan for being in the assessment committee of my thesis. I am also very grateful to Dr. Gerhard Hiermann for the many always constructive technical discussions, especially at the beginning of my Ph.D. journey. While I am very thankful to the whole research group for the many opportunities to discuss my research, I owe special thanks to Patrick, Christina, and Julius who helped me finalize the writing of my dissertation.

I would also like to thank Dr. Alexander Kroëller who created this research opportunity by making the collaboration between academia and industry possible, no matter the barriers, and helped me to set my research direction. I am also grateful to my colleagues at TomTom who helped me to shape my research and to sharpen the resulting insights. In particular, I would like to thank Dr. Falk Hüffner for our discussions and the time he spent to provide me with constructing feedback.

On a personal level, I would like to thank my family, and especially my parents, for their unconditional support and for having encouraged me to pursue scientific studies. I would also like to thank all my friends who made my life much brighter during this time. I finally would like to thank Clément for standing by me all these years, being there in the better but also worst times of this journey.

Abstract

While electric vehicle (EV) adoption is crucial to reducing the carbon footprint of future mobility systems, many drivers remain reluctant to purchase an EV if they cannot rely on an at-home charging solution. Limited public charging station availability and reliability cause the so-called charge anxiety, i.e., the fear of not being able to rely on the public charging infrastructure when needed to recharge a vehicle, due to non-usable or missing public charging stations. This thesis aims to mitigate the users' perceived charge anxiety by maximizing the chances of quickly finding an available and usable public charging station, i.e., with a non-broken, reachable, and compatible charging connector.

This thesis consists of an introduction, a state-of-the-art chapter, three methodological chapters, and a conclusion. The introduction presents background information on vehicle electrification developments and principal barriers against a faster EV adoption. The state-of-the-art chapter comprehensively reviews related applications and methodology papers, and identifies existing research gaps. The first two methodological chapters address the problem from a short-term user perspective. Here, the objective is to provide reliable in-car guidance instructions that allow a driver to seamlessly find a suitable charging station by explicitly accounting for a charging station's availability uncertainty. The first methodological chapter studies a single-agent setting and introduces a comprehensive algorithmic framework composed of a rollout algorithm and a multi-label setting algorithm with novel dominance criteria, which allows to save up to 44% of a driver's search time. The second methodological chapter studies a multi-agent setting and focuses on identifying the best possible driver coordination strategy, via information-sharing and possible charging requests centralization. Numerical results show that advanced charging station search methods, including driver coordination, outperform greedy charging station searches by decreasing the multi-agent search cost by up to 38%. The third methodological chapter addresses the problem from a long-term system perspective and aims to minimize potential station utilization conflicts due to uncoordinated charging demand between navigation service platforms. Here, the objective is to synchronize the public charging infrastructure utilization of several drivers that get guidance instructions through self-interested navigation platforms. A mechanism design approach is developed that allows to gear the behavior of such platforms towards an outcome that benefits all EV drivers, by decreasing the social cost up to 42% in an online problem setting. Finally, a conclusion summarizes and critically discusses the main methodological contributions and managerial insights of this thesis, and highlights possible avenues for further research.

1 Introduction

1 Background

The transportation sector contributed to 26% of worldwide carbon emissions in 2019 (IEA 2019) and is the second biggest carbon emitter, with more than 90% of the sector's energy demand being oil-based (EIA 2021). Accordingly, decarbonizing future mobility systems is crucial to achieving climate change mitigation targets. To this end, mitigation policies from the international panel on climate change (IPCC) report urge to foster a modal shift toward low-carbon mobility systems (e.g., public transportation, shared mobility, walking, cycling), to enhance communication technology (e.g., in order to facilitate home-office), and to improve urban planning (e.g., to lower traveling distances) (Sims et al. 2014). While these measures increase the use of non-motorized mobility, it remains essential to cut down the prevailing carbon emissions caused by the remaining motorized mobility modes.

In this context, battery electric vehicles (BEVs) constitute an essential component of future mobility systems, which allows to cut emissions if accompanied by strong measures on energy mix decarbonization, battery technology improvement, and vehicle weight reductions. In countries with intensive coal-based electricity, electric vehicles (EVs) show little lifetime emissions savings compared to conventional vehicles. In contrast, studies report significant lifetime emissions reductions in countries with a low-carbon energy mix, e.g., a Nissan EV Leaf in the UK emits 70% fewer emissions than an average conventional vehicle. Whereas the EV uptake may increase the stress on electricity distribution and transmission networks, smart-charging strategies can limit the additional energy consumption to meet the charging demand, e.g., by no more than 10% in the UK (Cambridge Econometrics 2017). Studies further show that electrification helps fleet operators to cut down emissions and operational costs (cf. Schmidt et al. 2021). Over the last years, field experiments verified theoretical expectations: the experimental electric car-sharing network rollout in the city of Stuttgart resulted in 12 064 kg savings of CO2 per 100 000 km driven, given a total of over 2 000 active users (cf. Unterwegs 2020). In practice, existing commercial services, e.g., Weshare, already operate free-floating electric car-sharing services; major logistics service providers unveiled their plans to electrify their fleets (DHL 2019, Amazon 2019), and municipalities increasingly operate electric buses.

To promote EV adoption for private users, the EU committed to invest 20 million euros to foster clean, i.e., electric and hydrogen, vehicle sales, as well as one million euros in clean charging stations (CSs) by 2025 (Virta 2021). Such investments support the EU target of having 30 million clean vehicles on the road by 2030. Governments complementary aim to support the shift from conventional vehicles to EVs for private vehicle owners through different national incentivization mechanisms. Most western Europe countries offer tax exemption (e.g., Belgium, Denmark, Norway) complemented by EV and charger purchase grants (e.g., Italy, France, Germany, The Netherlands, Luxembourg, Sweden, Spain). At the local level, municipalities can also implement incentive mechanisms (e.g., free parking space, charger

purchase grant), or enforce punitive measures (e.g., eco-tax, conventional vehicles ban in cities). Moreover, the automotive industry significantly matured its production of BEVs in recent years, announcing more than 100 new pure EV models to be ready by 2024 and accessible for all budgets. As a result, the market of privately owned vehicles is showing an encouraging growth trend, with BEV sales doubling in 2021 and the share of BEVs increasing in the EV mix, i.e., BEV and plug-in hybrid EV, in 2021.

However, an in-depth look at the adoption rates reveals significant geographical disparities worldwide. In 2021, the EV share of new vehicle sales reached 75% in Norway but only 2% in the US. The BEV share of the total vehicle volumes attained 52% in Norway but only 7% in France (cleantechnica 2021). In general, public charging infrastructure coverage in Europe is highly uneven and correlates with the EV market share and the GDP per capita of the country (ACEA 2021). For example, the Netherlands possesses 47.5 EV chargers (EVC)/100km, but Lithuania only reports 0.2 EVC/100km (Intertraffic 2021). Besides lacking significant charging infrastructure interoperability, the infrastructure quality can be heterogeneous even within a single country. In recent years, highway-charging service quality has increased, but public charging infrastructure can still be largely insufficient in remote areas with limited connection to larger cities or highways (cf. Hsu and Fingerman 2021, Zap-Map 2022). Furthermore, a user survey (Brückmann et al. 2021) reveals that house-ownership correlates with a higher EV adoption rate contrary to flat owning or renting. In the former case, users can easily install and rely on their home-charging solution, whereas in the latter case, users depend on the charging and parking capabilities of their building. If home-charging is impossible, then the likelihood to adopt an EV is much lower, as users must solely rely on the public charging infrastructure, which might be missing or deficient.

In fact, the major remaining obstacles to a faster private EV adoption are the users' perceived anxieties related to recharging at public infrastructure. The *range anxiety* describes the fear of not being able to reach the next CS with the current battery' state of charge in a sparse charging network. Furthermore, the *charge anxiety* describes the fear of relying on a public CS to recharge an EV, due to the heterogeneous quality and availability of existing charging infrastructure. Map and EV-navigation services attempt to mitigate these anxieties by providing driving range or CS locations, along with – if available – additional charging station characteristics, e.g., plug-types, opening hours, and availability status. However, such services still proved insufficient to cope with all existing infrastructure limitations, e.g., unreliable or missing availability information.

Not being able to find an available charging station (US) or a non-broken charging station (France) is the most cited pain point and source of frustration caused by missing, poorly maintained, or hardly accessible CSs (TomTom internal study 2020). The - sometimes unexpected - long charging time comes as second most cited source of frustration. In addition, a crucial limitation factor is the availability and reliability of CSs' availability status data. While 14% of the CSs do not report dynamic availability in the Netherlands, this number

may rise to 80% in Berlin, Germany (TomTom Internal study 2020). When dynamic data is available, discrepancies between reported and actual physical availabilities may still be observed due to physical limitations, e.g., broken or unreachable connectors, but also due to ICEing, i.e., conventional vehicles parking next to charging stations and blocking the access for EVs. Up to 34% of observed CSs report being available while actually being blocked by a non-charging (electric) vehicle in Berlin, in areas with low parking availability (Guillet et al. 2022). Similarly, french EV drivers mention planning their routes and charging stops by considering fallback stations in the vicinity of the initially planned CSs, to cope with availability uncertainty (ACOZE 2018).

Uncoordinated charging represents another limiting factor. Today's navigation service APIs serve stateless charging requests, i.e., recommend identical CSs for users with similar search attributes, e.g., same request location, which may create station utilization conflicts. A survey realized by the Norwegian EV users association points out that 88% of users have experienced queuing at a fast-CS (Frydenlund 2016). In line with these users' observations, facilitating a reliable EV charging process by ensuring that drivers can easily find and seamlessly use an available CS is critical to reducing EV drivers' anxieties (Bonges and Lusk 2016, McKinsey 2020).

Accordingly, multiple measures may help to improve the users' charging experience. At the strategic planning level, increasing the CS coverage or the capacity of existing stations, and detecting accurate availability through sensor detection is necessary to increase the public charging infrastructure usability. Additional measures to enable a seamless charging experience include improved interoperability between charging service providers but also better price transparency (Bonges and Lusk 2016). As such strategic solutions require long lead times, complementary solutions are necessary at the operational level to improve today's charging experience. Cooperative strategies leveraging vehicle-to-vehicle and vehicle-to-grid technology may free up some charging capacities to cope with the currently undersized charging infrastructure (You et al. 2016). However, such solutions remain unimplemented at the moment due to immature technology, uncertain economic values, or low user acceptance (Lauinger et al. 2017).

Against this background, we identify three levers to address today's and future charging infrastructure deficiencies, with a high positive impact potential on the user's charging experience. From a short-term and operational perspective, easily implementable levers are

i) *Providing a reliable charging experience* by delivering reliable navigation guidance to EV drivers through in-car or mobile navigation apps while accounting for the CS's availability uncertainty.

ii) *Enabling local charging coordination* by centralizing charging requests and enabling information-sharing between vehicles and navigation services besides accounting for CS's availability uncertainty.

From a long-term strategical perspective, the development of vehicle electrification may reduce some of today's issues, such as charging station availability uncertainty, e.g., with sensors, new parking regulations. However, the expected charging demand increase may lead to new problems. In particular, misaligned charging demand and charging capacity supply with respect to charging stations or energy availability can lead to station use congestion and users' dissatisfaction, even within a properly dimensioned charging infrastructure. Expected infrastructure bottlenecks can be addressed in the future as follows

iii) *Providing system-level charging demand coordination* by aligning locally coordinated charging demand with available CSs, via incentivization mechanisms controlled by non-profit entities, e.g., municipalities.

2 Aim and scope of the thesis

This thesis aims to address existing but also upcoming challenges of high-practical relevance related to EV adoption, at both the individual and the system level. Specifically, the scope of this thesis is twofold. First, the thesis provides an optimization-based framework and contributes novel models and novel broadly applicable solution methods that solve the following problems

i) *Single-agent CS search under uncertainty:* In a single-agent setting, an EV driver aims to find a sequence of stations that minimizes her expected search cost to reach an available station, considering heterogeneous CSs with stochastic availability.
ii) *Multi-agent CS search under uncertainty:* This multi-agent setting extends the single-agent setting described above by accounting for possible charging requests centralization, and for information-sharing between drivers.
iii) *Competitive CS allocation:* Assuming deterministic CS availability, with selfish navigation platforms that optimally allocate their respective charging demand to stations, a non-profit entity aims to best align the overall charging demand with all available CSs.

Second, managerial and technical insights that validate and evaluate the benefits of the proposed solutions are derived from extensive numerical experiments, based on real-world data.

3 Structure and contribution of the thesis

The remainder of this thesis is organized as follows.

Chapter 2 reviews the literature related to the three optimization problems described above. Accordingly, path planning problems, as well as search problems under uncertainty are reviewed, both in single-agent and multi-agent settings. In addition, resource allocation

problems in game-theoretical settings are discussed. Besides the analysis of application papers and the used methodology, this chapter reviews purely methodological papers and identifies existing research gaps.

Chapter 3 introduces the stochastic charging pole search (SCPS) problem and formalizes it as a finite-horizon Markov decision-process (MDP). To solve this problem, this chapter derives a comprehensive algorithmic framework composed of a novel label-setting algorithm that provides an exact and a heuristic solution method best suited for small to medium-sized test instances, and a rollout algorithm that allows to solve large-sized instances. An extensive computational study outlines the benefit of a stochastic search method, compared to naive EV users' search strategies. Specifically, the numerical results highlight that such advanced search methods may help an EV driver save up to 44% of her search time while increasing the likelihood of successfully terminating the search within a given time budget. Results further show that accounting for time-dependent recovering probabilities has a negligible impact on the driver's search experience.

Chapter 4 introduces the multi-agent SCPS problem (MASCPS), by extending the SCPS problem to a multi-agent setting. The planning problem is first formalized as an MDP, that can be solved from a driver or a system perspective. Then, several online algorithms are developed, that allow solving various information-sharing scenarios: drivers can share their planned visits, their observations of a charging station's occupancy, or both. Practical and managerial insights are derived from an extensive simulation-based computational analysis, emphasizing the best search coordination strategies, considering quality performance and implementability, i.e., real-world requirements to implement the solution in practice. The results show that a decentralized coordination strategy that provides static recommendations to drivers, i.e., a fixed sequence of stations to visit, decreases the system cost obtained without coordination by 26%, as long as visit intentions are shared. Additional experiments show, that from a user perspective, a coordinated search strategy dominates an uncoordinated search strategy.

Chapter 5 introduces the fleet CSs allocation (FCSA) game, in a setting with deterministic CS availability, in which multiple commercial navigation service platforms, modeled as players, aim to best allocate their charging demand to existing available CSs. First, it is shown that no pure Nash equilibrium guarantee exist. Then, centralized coordination is enforced by implementing the Vickrey–Clarke–Groves (VCG) mechanism in both offline and online settings. In the online setting, a data-driven station allocation policy is derived for the mechanism's principal. Numerical studies analyze the benefit of centralized coordination over purely selfish agents' behaviors. The results show that VCG-based coordination can achieve on average a 52% social cost decrease in perfect information settings, and a 42% social cost decrease in online settings. Results further highlight that a data-driven online policy outperforms a myopic online policy.

Chapter 6 concludes the thesis, summarizes its contribution and results, and provides a

brief research outlook.

References

ACEA. 2021. Making the transition to zero-emissions mobility. Tech. rep., European Automobile Manufacturers' Association, Belgium.

ACOZE. 2018. Pour un système efficace et simple de recharge des véhicules électriques à batterie. Tech. rep., Association des conducteurs de vehicles zéro emission.

Amazon. 2019. Amazon co-founds the climate pledge, setting goal to meet the paris agreement 10 years early. https://www.businesswire.com/news/home/20190919005609/en/ (last accessed: 2019-12-14). Last accessed: 11.27.2019.

Bonges, H. A., A. C. Lusk. 2016. Addressing electric vehicle (ev) sales and range anxiety through parking layout, policy and regulation. *Transportation Research Part A: Policy and Practice* **83** 63–73.

Brückmann, B., F. Willibald, V. Blanco. 2021. Battery electric vehicle adoption in regions without strong policies. *Transportation Research Part D: Transport and Environment* **90** 102615.

cleantechnica. 2021. 16 countries now over 10% plugin vehicle share, 6 over 20%. Url: https://cleantechnica.com/2021/09/05/16-countries-now-over-10-plugin-vehicle-share-6-over-20/ (last accessed: 04.04.22).

DHL. 2019. Dhl electro mobility press release. https://www.dpdhl.com/en/media-relations/specials/electro-mobility.html. Last accessed: 11.27.2019.

Econometrics, Cambridge. 2017. 2040 government ban positive: but significant challenges remain. Tech. rep. Url: https://www.camecon.com/news/2040vehicleban/# (last accessed: 04.04.22).

EIA. 2021. International energy outlook 2021. Tech. rep., U.S. Energy Information Administration.

Frydenlund, S. 2016. The norwegian ev success story: Reaching mass market. Tech. rep., Norwegian EV Association.

Guillet, M., G. Hiermann, A. Kröller, M. Schiffer. 2022. Electric vehicle charging station search in stochastic environments. *Transportation Science* **56**(2) 483–500.

Hsu, C., K. Fingerman. 2021. Public electric vehicle charger access disparities across race and income in california. *Transport Policy* **100** 59–67.

IEA. 2019. Global co2 emissions by sector, 2019. Tech. rep., International Energy Agency, Paris.

Intertraffic. 2021. Electric vehicle charging infrastructure in europe 2021. Url: https://www.intertraffic.com/news/infrastructure/electric-vehicle-charging-2021-europe/# (last accessed: 04.04.22).

Lauinger, D., F. Vuille, D. Kuhn. 2017. A review of the state of research on vehicle-to-grid (v2g): Progress and barriers to deployment. *Working paper, Ecole Polytechnique Fédérale de Lausanne* .

McKinsey. 2020. The road ahead for e-mobility. https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/the-road-ahead-for-e-mobility. Last accessed: 02.25.2021.

Schmidt, M., P. Staudt, C. Weinhardt. 2021. Decision support and strategies for the electrification of commercial fleets. *Transportation Research Part D: Transport and Environment* **97** 102894.

Sims, R., R. Schaeffer, F. Creutzig, X. Cruz-Núñez, M. D'Agosto, D. Dimitriu, M.J. Figueroa Meza, L. Fulton, S. Kobayashi, O. Lah, A. McKinnon, P. Newman, M. Ouyang, J.J. Schauer, D. Sperling, G. Tiwari. 2014. Transport. *Climate Change 2014: Mitigation of Climate Change. Contribution of Working Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA.

Unterwegs, Zeozweifrei. 2020. Electric car sharing programme in germany cuts carbon footprint and traffic. Url: https://ec.europa.eu/regional_policy/en/projects/Germany/electric-car-sharing-programme-in-germany-cuts-carbon-footprint-and-traffic (last accessed: 04.04.22).

Virta. 2021. Here's how eu legislation accelerates the ev revolution. Url: https://www.virta.global/blog/this-is-how-eu-regulation-accelerates-the-electric-vehicle-revolution (last accessed: 04.04.22).

You, P., Z. Yang, M. Chow, Y. Sun. 2016. Optimal cooperative charging strategy for a smart charging station of electric vehicles. *IEEE Transactions on Power Systems* **31**(4) 2946–2956.

Zap-Map. 2022. Ev charging stats 2022. Url: https://www.zap-map.com/statistics/#points (last accessed: 13.06.22).

2 State of the art

1 Introduction

Electric vehicle (EV) adoption raises new operational challenges, due to limited vehicle driving range, hard-to-predict energy consumption, and long recharging times. Accordingly, EV fleet operators must plan their vehicles' routes and charging schedules but can mitigate operational uncertainties if relying on their own private charging infrastructure and centralized coordination. In contrast, private EV users are additionally facing road and charging networks uncertainties, due to road and station congestion, public charging infrastructure unreliability, and uncertain behaviors of other road and charging station (CS) users. Accordingly, increasing private EV adoption necessitates reducing users' anxieties related to unreliable charging (Bonges and Lusk 2016, Myersdorf 2020). Besides enhancing charging tariffs transparency and up-scaling the charging infrastructure, readily usable solutions that can provide a seamless charging experience to the drivers will be necessary (McKinsey 2020).

Against this background, this thesis focuses on developing algorithmic solutions that allow to provide reliable guidance to private EV users via navigation services to address infrastructure and charging network congestion-related uncertainties. The work presented in this thesis belongs to several research streams, covering different methodological concepts, e.g., Markov decision process (MDP), graph theory, mechanism design (MD), and contributes to different application areas, e.g., EV routing, and multi-agent system (MAS). As the thesis focus lies on private users, work centered around logistics operations that only considers private depot recharging, is excluded from this literature review. However, work that addresses uncertainties related to non-depot charging in logistics is included. The remainder of the review divides into three EV focused topic areas that best relate to the optimization problems analyzed in the subsequent chapters,

- *En-route charging under uncertainty:* addresses work on road network and CS capacity and availability uncertainties, focusing on single-agent settings.
- *Coordinated charging under uncertainty:* addresses work on users, road and charging network uncertainties, focusing on multi-agent settings, with controllable agents.
- *Non-cooperative charging:* addresses work on user-related uncertainties, focusing on self-interested multi-agent settings.

For each topic area, the following sections review related problems that may also extend beyond EV applications and provide an overview of possible solution approaches.

2 En-route charging under uncertainty

This section focuses on optimizing charging station selection under uncertainty in a single-agent setting. Section 2.1 reviews path planning under uncertainty, first in general settings, and then in the context of EV path planning with intermediate charging stops. Then,

Section 2.2 focuses on related resource search problems, including but not limited to CS search.

2.1 Path planning under uncertainty

Path planning under uncertainty was first analyzed within the problem of pursuit (Eaton and Zadeh 1962), which has been defined as the problem of catching a target randomly moving between a finite set of states. Frank (1969) mentioned the path planning problem under uncertainty as the shortest path problem in probabilistic graphs, while Derman (1970) referred to it as the first passage problem to name a few. Bertsekas and Tsitsiklis (1991) defined the stochastic shortest path problem that allows both positive and negative arc costs, as the stochastic variant of its deterministic counterpart. The authors further showed that the problem variant with non-negative or non-positive arc costs can be formalized as an MDP (cf. Bellman 1954, Puterman 1994). Considering arc weights to be independent, discrete, and time-varying random variables, Miller-Hooks and Mahmassani (1998) studied the least travel time path problem, while Miller-Hooks and Mahmassani (2000) solved the least expected time path problem using a label-correcting algorithm. Nie and Wu (2009) addressed the problem of finding a-priori non-dominated paths in stochastic time variant networks, with a guaranteed likelihood of on-time arrival. Focusing on an optimal routing policy (ORP), i.e., a set of routing decisions depending on realized arc weights, rather than an optimal path (i.e., a set of vertices), Andreatta and Romeo (1988) studied the ORP problem in a stochastic network. Polychronopoulos and Tsitsiklis (1996) extended Andreatta and Romeo (1988)'s work to account for correlated arc weights. The first time-dependent variant of the ORP problem has been studied by Hall (1988). Assuming dependent random arc weights, Gao and Chabini (2006) proposed the first framework for ORP problems, in stochastic time-dependent networks. They derived an exact algorithm, as well as approximation strategies, to solve the online problem variant with perfect information, i.e., when a traveler knows at a given time t about all edges' travel times realizations prior to t. Gendreau et al. (2015) provided a comprehensive overview of recent works on time-dependent arc traversal times.

In the context of path planning problems for EVs, Jafari and Boyles (2017) studied the problem of finding minimal-cost EV paths under stochastic arc travel times and energy consumption. They proposed a label-setting algorithm (cf. Martins 1984) based on generalized dynamic programming (cf. Carraway et al. 1990). While Jafari and Boyles (2017) focused on uncertain arc travel costs, De Weerdt et al. (2016) and Sweda et al. (2017) additionally modeled uncertain CS availability due to station congestion in the realm of long-distance EV path planning. De Weerdt et al. (2016) solved a stochastic time-dependent routing problem modeled as an MDP that accounts for waiting time at CSs, based on the work of Gao and Chabini (2006). The authors implemented a system that centralizes historical and reported EV drivers routing intentions to derive probabilistic waiting times. Each EV driver benefits from the centralized information to optimally plan her routing policy. In Sweda et al. (2017),

the authors explicitly modeled stochastic CS availability for the first time, and assumed that occupied charging stations become available again after some expected amount of time. Besides deriving a-priori routing strategies based on dynamic programming in a stochastic grid network, they further derived dynamic re-planning heuristics to obtain en-route adaptive routing and charging strategies. Gareau et al. (2019) extended the work of Sweda et al. (2017), by solving real-world instances, and accounting for dynamically selected alternative paths upon an EV driver's journey. Focusing on charging operations for EV fleets instead of private users, Kullman et al. (2021) were the first to consider charging strategies that use both an operator's private and public charging infrastructure, while accounting for the CS availability uncertainty. Specifically, they represented the problem of deciding where to recharge the fleet's vehicles as a finite-horizon MDP, allowing multiple vehicles to queue at a CS if already in use. As can be seen, there has been a very limited number of publications that explicitly modeled stochastic CS availability for EV path planning.

2.2 Resource search

Resource search problems deal with finding resources under availability and location uncertainty in grid or graph networks. In contrast to path planning problems, the process of searching for resources may terminate after one or many resources have been collected independent of a target destination. While practical solutions in this context may only rely on real-time information sharing, e.g., using wireless embedded sensors, we only review work that explicitly models resource stochasticity. Such problems can be either viewed as variants of the traveling salesman problem (TSP) or as sequential decision-making processes. In the former case, nodes that contain resources may represent customers to visit, weighted according to the resource availability probability. In the latter case, deciding which resources to visit next occurs after each visit to an occupied resource.

Focusing on the first category, Verroios et al. (2011), and Arndt et al. (2016) addressed the public parking spot search problem, considering probabilistic parking spot availability. Verroios et al. (2011) assumed availability probabilities to depend on the traveled time between the origin and the potential parking spot, located on the graph's nodes. They derived edge weights based on a cost combining the time-dependent availability probability and the deterministic edge travel times, and implemented an exact solution based on the time-dependent TSP for small-sized instances. Arndt et al. (2016) defined the problem on a probabilistic road network graph and associated each edge with a probability representing the likelihood of finding a free spot while traversing it. Their probabilistic model built upon the work of Jossé et al. (2015), and assumed time-independent and a-priori known probabilities. The authors showed the NP-completeness through reduction to the Hamiltonian path problem, and used a branch-and-bound based procedure to find a path that minimizes a generalized cost, corresponding to the expected arrival time at a free charging spot. Focusing on similar applications, Shao et al. (2018) modeled the traveling officer problem that aims to find a tour

maximizing the probability of issuing infringement notices for vehicles that violate parking restrictions, and proposed an ant-colony optimization based heuristic solution.

Modeling the resource search as a decision-making process, Tang et al. (2013) formalized the taxi customer pick-up problem as an MDP, modeling for each (clustered) pickup area the probability of finding a lucrative passenger, and solving small-sized instances with a policy-iteration algorithm. Guo and Wolfson (2018) solved the problem of collecting generic resources with uncertain availability located on a graph's vertices as a dynamic program. They forbade agents to wait at an occupied resource similar to Arndt et al. (2016) but allowed the availability probability of a resource observed as occupied to recover over time. More recently, Schmoll and Schubert (2018) formalized the dynamic resource routing problems as an MDP, modeling probabilities of resources allowed to reappear (Jossé et al. 2015). Their approach built on real-time dynamic programming to provide routing decisions to agents that can account for up-to-date resource state information. Schmoll and Schubert (2020) extended Schmoll and Schubert (2018) to a resource collection problem, in which an agent aims to collect as many resources as possible within a limited time span (cf. Shao et al. 2018). Here, resource availability is stochastic but cannot be explicitly represented, as they used a deep Q-network-based solution method, to solve their problem formalized as a Semi-MDP (S-MDP) (see Sutton et al. 1999) with unknown transition functions.

None of the work that modeled stochastic resource search processes as sequential decision-making processes has considered both the heterogeneity of the resources and the possibility to wait for an occupied resource to become available again. Furthermore, no exact algorithms solve the underlying MDP by exploiting the problem structure. In our specific application case, the decision process can be represented on a road network and adapted multi-criteria shortest path algorithms can be used as computationally efficient solution approaches.

3 Coordinated charging under uncertainty

In this section, we survey work that relates to the problem of coordinating the utilization of CSs across multiple EV users, under uncertainty. Section 3.1 reviews resource search problems under uncertainty in a multi-agent setting, while Section 3.2 details work focused on multi-agent path planning. We analyze both research streams in the context of MAS, focused on controllable agents. A MAS describes a self-organized system composed of multiple agents whose actions interfere with each other. The degree of information, centralization, and cooperation, i.e., whether agents share a common goal, help each other, or pursue their individual goals, may significantly vary according to the underlying problem.

3.1 Multi-agent resource search

Multi-agent resource search problems often arise in the context of robotics and system control theory. A commonly analyzed problem is the target search problem via multiple

sensor-detection, where uncertainty may relate to both the target location and the sensor observation (Polycarpou et al. 2001, Bourgault et al. 2003, Song and Teneketzis 2004, Wong et al. 2005). Song and Teneketzis (2004) assumed that sensors are located at each potential target location and aimed to find an optimal sensor utilization strategy, given that sensors can be used only once at a time. Focusing on moving vehicles, Polycarpou et al. (2001) dealt with the problem of cooperatively detecting a target using vehicles with sensors capabilities. They proposed a two-stage solution framework, that i) maps an environment using the vehicle's own sensing capabilities and observation sharing, and ii) computes the optimal trajectory accordingly. Similarly, a problem setting with several moving sensing agents searching for a resource with no a-priori information on its location has been studied in Bourgault et al. (2003). Here, decision-making was decentralized, i.e., took place at the agent level. However, agents shared observations via a decentralized Bayesian data fusion technique, allowing agents to build equivalent representations of their environment. Wong et al. (2005) extended this work to a multi-target search problem with independent targets, such that several instances of a Bayesian filter can be used. In addition to Bourgault et al. (2003)'s setting, Chung and Burdick (2008) leveraged belief distribution feedback to dynamically adjust agents' search strategies. Following Wong et al. (2005), Dai and Sartoretti (2020) analyzed a multi-target environment in a decentralized-decision making setting, leveraging local coordination via an information exchange procedure between nearby agents. Using a deep reinforcement learning (RL) solution approach, their results highlighted the benefit of collaboration with information-sharing compared to a non-collaborative setting.

More recently, related search problems were studied in the context of taxi or ride-hailing order dispatching for vehicle fleets in large-scale stochastic settings (see Xu et al. 2018, Wang et al. 2018b, Tang et al. 2019, Kullman et al. 2022). Here, the customer to be picked up constitutes the resource, and uncertainty relates to the location of future customer demand but not the availability of the resource at a given location. Xu et al. (2018) analyzed the problem of matching drivers and on-demand orders on a ride-hailing platform. Their approach accounted for future customer orders by formalizing the sequential planning process as an MDP. They obtained the matching action from a min-cost assignment problem solution, deriving the assignment cost from the temporal difference error between the driver's state at picking and its current state, using approximated evaluation of the value function. Tang et al. (2019) extended the work of Xu et al. (2018), by formalizing the sequential process of matching customers and vehicles, as a S-MDP. The authors extended the RL approach to a deep-RL approach, using the Cerebellar Value Network (CVnet), and adopting the transfer learning solutions described in Wang et al. (2018b). Similarly using deep-RL but with dynamic DQNs, Kullman et al. (2022) studied the order dispatching problem for ride-hailing platforms operating EVs, accounting for battery depletion and vehicle recharging.

So far, there has been no work on the multi-agent search problem in the context of EV CS search under uncertainty. Most of the work on multi-agent search problems treated agents

as fully cooperative, and synchronously searching for one or multiple resources. In none of the settings, multiple agents can visit several occupied resources in parallel, and stop their search at the first available resource. In large-scale ride-hailing settings with unseen future customer demand, most publications integrate a learning component as uncertainty only relates to future demand. In the case of multi-agent CS searches, uncertainty relates to both future demand and resources availability, but can be explicitly represented for resources' availability,

3.2 Multi-agent path planning

Finding one (or many) resource(s) for multiple agents can be analyzed from a multi-agent routing perspective, which minimizes the possible collusion between agents when visiting or heading towards the resources. In the context of robot navigation in MAS, the multi-agent pathfinding (MAPF) problem (cf. Erdmann and Lozano-Perez 1987) aims to find non-colluding routes for several agents that simultaneously traverse a transportation network. Researchers have solved a variety of MAPF problem variants, with different sets of assumptions and objectives. Stern et al. (2019) gave a recent overview of MAPF problems in the literature, besides existing benchmarks to evaluate the performances of MAPF algorithms.

Unlike non-cooperative path planning, agents have full knowledge about other agents' plans in cooperative path planning, and do not use this knowledge to prevent other agents from reaching their destination unlike in antagonistic path-finding. Focusing on the former, planning can be either coupled, i.e., realized by a centralized planner, or decoupled, i.e., decomposed into tasks that can be individually solved by each agent. Coupled planning can be highly intractable (Hopcroft et al. 1984) but solves the problem to optimality (cf. Hopcroft et al. 2008, Standley 2010, Standley and Korf 2011, Sharon et al. 2011). Sharon et al. (2011) however showed that there do not exist any dominant optimal algorithms, due to the high sensitivity of the optimal algorithms' performances to the test instances characteristics, e.g., graph sparsity. Decoupled planning (Erdmann and Lozano-Perez 1987) allows agents to individually search for a path while avoiding collision given other agents' states, which yields competitive computational times but no optimal solution. In Zelinsky (1992), the authors developed the local repair A* algorithm to account for the re-planning of an agent's initial route when the risk of collision with another agent increases. Silver (2005) extended the local repair A* by hierarchically pre-planning agents' routes but also re-planning routes at regular intervals to avoid a collision. In Sharon et al. (2015), the authors presented an optimal conflict-based search algorithm that leveraged both coupled and decoupled planning, using a two-level solution approach. In this work, a high-level constraints tree is maintained based on potential route conflicts for all agents, while at the agent level, each agent's path is planned such that it satisfies all upper-level constraints. As such traditional multi-agent path planners can fail to scale, modern artificial intelligence (AI)-based approaches may solve larger-scale settings, e.g., Sartoretti et al. (2019)'s work

on large factory robot navigation. In this work, the authors combined distributed RL (cf. Foerster et al. 2016, Gupta et al. 2017, Lowe et al. 2017) and imitation learning to train a high number of agents to cooperatively find non-colluding routes. Dai and Sartoretti (2020) extended Sartoretti et al. (2019)'s work by identifying conventions and agent behaviors that can be locally learned to improve the implicit coordination and solve the MAPF problem in dense and more constrained environments. In a similar setting, Li et al. (2020) used graph neural networks to improve agents' knowledge sharing.

None of the existing work has studied cooperative path planning problems in the larger context of EVs. Here, some exploration of the value of information-sharing and planning centralization in the multi-agent EV path planning problem may yield interesting insights, especially focusing on autonomous mobility-on-demand systems. In particular, it may be crucial for system designers to implement the right-trade off between system operational performance and data-efficiency.

4 Non-cooperative EV charging

This section details work that relates to EV charging with self-interested agents. Focusing on a game-theoretical perspective, Section 4.1 reviews resource allocation games, before Section 4.2 briefly details work that focuses on congestion reduction. Finally, Section 4.3 reviews MD techniques to align agents and system interests, and related application works.

4.1 Resource allocation games

Resource allocation games model the allocations of pooled resources among non-cooperative agents. Congestion games were introduced as a framework to model resource allocation games (Rosenthal 1973) and were referred to as atomic selfish routing games in contrast to non-atomic selfish routing games (see Pigou 1920, Wardrop 1952). In congestion games, there exists a common set of resources that any player can use. A player's strategy consists of a subset of the resources, while its payoff amounts to the sum of the selected resources costs. A resource cost depends on its utilization congestion, i.e., is a function of the number of players simultaneously selecting this resource. Monderer and Shapley (1996) showed that congestion games belong to the broader category of potential games, which admit a player-independent function – a potential function– that reflects a player payoff variation between two different strategies. Such games have essential properties, such as the guaranteed existence of a pure Nash equilibrium (PNE) or the finite improvement property, i.e., best response dynamics convergence. Milchtaich (1996) proved that congestion games can extend to player-specific payoffs without losing the PNE existence guarantee for singleton strategies. While Fotakis et al. (2002) showed that weighted players games with singleton strategies similarly possess a PNE, this property does not hold anymore for non-linear resource utilization cost functions (Fotakis et al. 2005). Ackermann et al. (2009) completed these works by showing that

the PNE existence guarantee holds when strategies are defined as the bases of a game's resources matroid. Airiau (2014) studied a multi-agent resource allocation problem that allows for resource synergies, by defining a player's payoff as the difference between its items bundle valuation and the resource congestion –or delay– cost. By possessing a PNE, this class of games is of particular interest to studying real-world applications, such as load-balancing (Even-Dar et al. 2003, Goemans et al. 2006), network design (Anshelevich et al. 2008), wireless network selection (Cesana et al. 2008, Malanchini et al. 2013), or tolling mechanisms (Paccagnan et al. 2021).

Game-theoretical work focusing on EV charging applications generally studied the equilibrium of energy prices or charging strategies that best align available energy supply with existing charging demand. Tushar et al. (2012) modeled the problem of defining profitable energy prices in presence of self-interested EV drivers as a Stackelberg game. Here, the electricity grid operator constitutes the leader, while EV drivers constitute the followers. Yoon et al. (2016) followed a similar approach but additionally considered constraints on the minimum energy amount to recharge. In Ma et al. (2013) or Karfopoulos and Hatziargyriou (2013), a large population of EV drivers must define the amount of energy to recharge at each point in time during a time-limited interval, based on energy prices that depend on the total charging demand. The authors iteratively computed a pure ϵ-equilibrium that achieves a near-optimal social cost for heterogeneous agents, and the social optimum for homogeneous agents. Zhang and Li (2016) extended Ma et al. (2013)'s work by assuming maximal individual drivers' charging rates and limited station transformers, for local charge scheduling coordination at charging parks, while Beaude et al. (2016) extended Ma et al. (2013)'s work by considering rectangular energy profiles and an accordingly discrete strategy action space. Wu et al. (2012) considered bidirectional charging in the vehicle-to-grid (V2G) context, allowing vehicles to act as additional energy storage devices. Here, the goal was to derive energy prices that incentivize EV owners to participate in the system. More recently, Sohet et al. (2021) discussed a game that mitigates the impact of EV charging in both energy and transportation networks by deriving energy prices that depend on the EV flow in the transportation network. Related to the latter case but not focused on EV charging, Ayala et al. (2011) introduced the competitive parking slot assignment problem, and an algorithm that computes equilibrium parking slot assignment strategies. Here, the successful assignment of a slot to a driver depends on its driving distance. Ayala et al. (2018) added a time component to Ayala et al. (2011)'s work, by allowing vehicles to enter and leave the system.

Most congestion game applications focused on singleton strategies. Usually, the value of resource utilization costs depends on the number of participants selecting the resource, the players' weights, or the player itself, but not on characteristics depending on the strategy profile, e.g., numeric distance from each participant to the resource. Furthermore, most game-theoretical settings of EV-related applications dealt with efficient charging strategies, in the context of smart grids. None of the aforementioned works solved the optimal distri-

bution of EV drivers across stations, realized by self-interested intermediate players, e.g., navigation service platforms.

4.2 Congestion reduction

Besides in system control and robotics, congestion reduction naturally arises in traffic management on highly-decentralized road transportation networks. To analyze congestion reduction strategies, game theory offers an adequate framework to manipulate drivers interested in minimizing their individual travel times. In this context, route guidance information systems (RGIS) aim to provide route choice recommendations or additional network information to their users, which benefit both the overall system but also each individual driver. If carefully designed, such systems proved to reduce total travel times (see Koutsopoulos and Lotan 1989, Mahmassani and Jayakrishnan 1991, Al-Deek and Kanafani 1993, Hall 1996), but also on-time arrival reliability (Wunderlich et al. 2001), while in the opposite case, they can lead to exacerbated congestion, due to an increased driver concentration on a restricted set of routes or system overreaction (Ben-Akiva et al. 1996). Predictive guidance accounts for network state forecasts, such as expected traffic. While it may achieve similar outcomes as non-predictive guidance under certain conditions, e.g., in densely meshed networks (see Pavlis and Papageorgiou 1999, Ben-Akiva et al. 1996), it can better reflect actual drivers' decisions. Predictive guidance must also be consistent, i.e., anticipating the impact of guidance on future network conditions, since a high number of system participants will likely affect network flows and conditions (cf. Barnhart and Laporte 2007). In contrast to such guidance-oriented traffic network models (Ben-Akiva et al. 1991), dynamic traffic assignment (DTA) models (Janson 1991) compute assignments that approximate a dynamic user-equilibrium based on the class of non-atomic selfish routing games. In this case, DTA does not account for the impact of available travel information on users' behaviors. Wang et al. (2018a) provided a comprehensive overview of more recent advances in DTA applications. Little work has been done so far to adapt traffic assignment models to EVs, i.e., accounting for range-constrained vehicles and limited charging infrastructure availability. A notable exception is the work of Zhang et al. (2019). The authors adapted the traffic assignment problem (TAP) under user equilibrium (UE) by considering EV's limited driving range and additional charging times. Enforced traffic control strategies complement these approaches, using, e.g., traffic signaling (Farges et al. 1990, Donati et al. 1984), or reactive ramp metering strategies (Masher 1975).

4.3 Market-based resource allocation

The social outcome that results from non-cooperative participants might be far from the social optimum, but market-based techniques can help mitigate the negative impact of ignored coordination among self-interested agents.

To reduce high prizes of anarchy, e.g., due to strategic drivers on a congested road network

(cf. Koutsoupias and Papadimitriou 1999, Roughgarden and Tardos 2002), MD approaches aim to design games – payoff structure and strategy space – that gear participants' behavior toward socially desirable outcomes (Nisan et al. 2007). Focusing on a resource allocation problem, the principal computes an allocation, i.e., the social outcome, based on participants' resources valuations, and a payment based on the received allocation. Assuming that a player's utility amounts to its allocation valuation and payment, the generic VCG pricing scheme (cf. Vickrey 1961, Clarke 1971, Groves 1973) ensures truthful revelations of a player's resource valuation, i.e., a player has no incentive to lie on its preferences. Roberts (1979) showed that truthfulness holds even for weighted players and allocations outcome for so-called affine maximizer social choice functions. The payment rule guarantees truthfulness by aligning a participant's interest with the overall system interest if the computed social outcome is optimal. To cope with suboptimal outcomes, Nisan and Ronen (2007) introduced the second-chance mechanism that gives participants the possibility to improve suboptimal social outcomes and accordingly provides a rationale for truth-telling behaviors. In practice, offline resource allocation applications might be limited, if participants dynamically enter or leave the system, or resources availability changes. In Parkes and Singh (2003), the authors introduced two mechanisms *delayed VCG* and *online VCG* mechanisms for online resource allocation. The underlying online allocation problem was formalized as an MDP to compute minimal in-expectation cost allocation and Bayesian-truthful payment rules, and found direct applications for WiFi pricing (Parkes et al. 2004a). Since the truthfulness relies on an optimal policies argument, Parkes et al. (2004b) discussed an approximately efficient mechanism extension, based on ϵ-greedy policies, similar to the work of Nisan and Ronen (2007) on offline VCG mechanisms. Parkes and Wellman (2015) placed Parkes and Singh (2003)'s work in more general settings, connecting AI and economic reasoning. Additional work that coupled mechanism design and dynamic resource allocation include Bi et al. (2019) and Stein et al. (2020). In Bi et al. (2019), the authors derived a price-based mechanism that guarantees dominant-strategy incentive compatibility and individually rationality to assign resource allocation in fog computing. In Stein et al. (2020), the authors solved a generic dynamic task allocation problem using a RL-based mechanism with guaranteed strategy-proofness.

Focusing on coordinating EV charging operations of selfish agents, Gerding et al. (2011) developed an online model-free mechanism ensuring that EV drivers truthfully report their energy demand and charging time window preferences, to achieve a socially optimal charge schedule. In Rigas et al. (2020), the authors formalized an offline charge scheduling problem that accounts for demand imbalance and energy cost and defined the charging prices based on the standard VCG pricing scheme. In Flath et al. (2014), the authors relied on dynamic electricity pricing to coordinate EV charging but argued that prices must, besides a temporal component, integrate a spatial component, to mitigate energy peak loads. Limmer and Rodemann (2019) derived a dynamic pricing scheme for a charging station operator

(CSO), based on deadline differentiated pricing (see Bitar and Low 2012), which requires drivers to announce their energy demand and their charging deadline for completed charging. Compared to earlier work on dynamic pricing for EV charging (see Limmer and Rodemann 2017, Ghosh and Aggarwal 2018), the authors accounted for extra energy peak demand fees, such that the pricing and charge scheduling strategy of a profit maximization CSO will contribute to reducing energy peak load caused by an increased charging demand. Foster and Caramanis (2013) designed a participation policy in an hour-ahead power market for a load aggregator managing the charging schedule of EVs, shown to outperform charge scheduling strategies based on forecasted electricity prices.

Similar to game-theoretic settings discussed in Section 4.2, most MD approaches focused on EV application dealt with efficient charging strategies that mitigated the discrepancy between possibly high charging demand and low capacity of energy infrastructure. In a broader context, none of the work that focused on static or dynamic resource allocations considered the allocation of resources to end consumers via intermediate self-interested agents.

5 Conclusion

Concerning operational problems addressed in the context of EVs, a large body of recent work focuses on mitigating the impact of increased charging demand on power networks. Another significant stream of research aims to adapt traditional routing and navigation algorithms for EVs, focusing on planning efficient charging stops during long-distance trips, and improved predictions of the car's energy consumption. In both cases, related publications usually consider the charging infrastructure to be reliable. From an application perspective, few studies have emphasized the uncertain charging infrastructure reliability or CS congestion, and the potential of improving the charging experience of the EV user through well-orchestrated CS visit recommendations from modern navigation services.

From a methodology perspective, a limited number of publications model stochastic resource search problems as sequential decision-making processes. None of the works discussed previously proposed efficient exact solution methods, at least in single-agent settings, but also practical heuristic solutions that can be readily deployed in navigation systems. In multi-agent settings, the problem of several collaborative agents asynchronously searching for a single stochastic resource and stopping their search upon discovery of one of the available resources has been rarely discussed. Most multi-agent search applications arise in robotics or ride-hailing problems, where no or limited information on the location and the availability probabilities may be available. Furthermore, only a very limited number of works compare performances of resource search with respect to the degree of decisions centralization, or the amount of information shared between agents, or between agents and a central decision-maker. Focusing on resource allocation games, none of the previous work considered the allocation of resources to users through intermediate self-interested agents, that a

higher-level orchestrator, e.g., a municipality, can coordinate.

A few last remarks are in order. Multi-agent settings significantly raise the problem complexity compared to their single-agent counterpart, and recent work on multi-agent resource allocation or search problems increasingly addresses very large-scale settings. Accordingly, most recent papers use deep-RL frameworks to leverage the enhanced capabilities of modern data storage and management. However, in multi-agent settings, such papers assume trustful agents and ignore possible strategic manipulation. Mechanism design approaches aim to optimize systems composed of unreliable agents that may miscommunicate some of their private information, but such systems are usually limited to myopic settings. In this context, there has been a recent effort to couple MD and RL approaches and, more generally, couple AI and economics (see Parkes and Wellman 2015).

References

Ackermann, H., H. Röglin, B. Vöcking. 2009. Pure nash equilibria in player-specific and weighted congestion games. *Theoretical Computer Science* **410**(17) 1552–1563.

Airiau, Endriss U., S. 2014. Multiagent resource allocation with sharable items. *Autonomous Agents and Multi-Agent Systems* **28** 956–985.

Al-Deek, H., A. Kanafani. 1993. Modeling the benefits of advanced traveler information systems in corridors with incidents. *Transportation Research Part C: Emerging Technologies* **1**(4) 303–324.

Andreatta, G., L. Romeo. 1988. Stochastic shortest paths with recourse. *Networks* **18** 193–204.

Anshelevich, E., A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, T. Roughgarden. 2008. The price of stability for network design with fair cost allocation. *SIAM Journal on Computing* **38**(4) 1602–1623.

Arndt, T., D. Hafner, T. Kellermeier, S. Krogmann, A. Razmjou, M. S. Krejca, R. Rothenberger, T. Friedrich. 2016. Probabilistic routing for on-street parking search. *24th Annual European Symposium on Algorithms (ESA 2016)*, vol. 57. 6:1–6:13.

Ayala, D., W. Ouri, D. Bhaskar, L. Jie, X. Bo. 2018. Spatio-temporal matching for urban transportation applications. *ACM Transactions on Spatial Algorithms and Systems (TSAS)* **3** 1–39.

Ayala, D., O. Wolfson, B. Xu, B. Dasgupta, J. Lin. 2011. Parking slot assignment games. *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*. 299–308.

Barnhart, C., G. Laporte. 2007. *Handbooks in Operations Research & Management Science: Transportation*. Amsterdam, The Netherlands: Elsevier.

Beaude, O., S. Lasaulce, M. Hennebel, I. Mohand-Kaci. 2016. Reducing the impact of ev charging operations on the distribution network. *IEEE Transactions on Smart Grid* **7**(6) 2666–2679.

Bellman, R. 1954. The theory of dynamic programming. *Bulletin of the American Mathematical Society* **60**(6) 503–515.

Ben-Akiva, M., A. De Palma, Kaysi I. 1991. Dynamic network models and driver information systems. *Transportation Research Part A: General* **25**(5) 251–266.

Ben-Akiva, M, A. de Palma, I. Kaysi. 1996. The impact of predictive information on guidance efficiency: An analytical approach. *Advanced Methods in Transportation Analysis*. 413–432.

Bertsekas, D. P., J. N. Tsitsiklis. 1991. An analysis of stochastic shortest path problems. *Mathematics of Operations Research* **16**(3) 580 – 595.

Bi, F., S. Stein, E. Gerding, N. Jennings, T. La Porta. 2019. A truthful online mechanism for resource allocation in fog computing. *PRICAI 2019: Trends in Artificial Intelligence*. 363–376.

Bitar, E., S. Low. 2012. Deadline differentiated pricing of deferrable electric power service. *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. 4991–4997.

Bonges, H. A., A. C. Lusk. 2016. Addressing electric vehicle (ev) sales and range anxiety through parking layout, policy and regulation. *Transportation Research Part A: Policy and Practice* **83** 63–73.

Bourgault, F., T. Furukawa, H. F. Durrant-Whyte. 2003. Coordinated decentralized search for a lost target in a bayesian world. *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*.

Carraway, R. L., T. L. Morin, H. Moskowitz. 1990. Generalized dynamic programming for multicriteria optimization. *European Journal of Operational Research* **44**(1) 95–104.

Cesana, M., I. Malanchini, A. Capone. 2008. Modelling network selection and resource allocation in wireless access networks with non-cooperative games. *2008 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*. 404–409.

Chung, T. H., J. W. Burdick. 2008. Multi-agent probabilistic search in a sequential decision-theoretic framework. *2008 IEEE International Conference on Robotics and Automation*.

Clarke, E. H. 1971. Multipart pricing of public goods. *Public Choice* **11** 17–33.

Dai, W., G. Sartoretti. 2020. Multi-agent search based on distributed deep reinforcement learning. Tech. rep., National University of Singapore.

De Weerdt, M. M., S. Stein, E. H. Gerding, V. Robu, N. R. Jennings. 2016. Intention-aware routing of electric vehicles. *IEEE Transactions on Intelligent Transportation Systems* **17**(5) 1472–1482.

Derman, C. 1970. *Finite State Markovian Decision Processes*. Academic Press, New York.

Donati, F., F. Mauro, G. Roncolini, M. Vallauri. 1984. A hierarchical-decentralized traffic light control system. the first realization: "progetto torino". *IFAC Proceedings Volumes* **17**(2) 2853–2858.

Eaton, J. H., L. Zadeh. 1962. Optimal pursuit strategies in discrete state probabilisti systems. *Trans. ASME Ser. D, J. Basic Eng* **84** 23 – 29.

Erdmann, M., T. Lozano-Perez. 1987. On multiple moving objects. *Algorithmica* **2** 477 – 521.

Even-Dar, E., A. Kesselman, Y. Mansour. 2003. Convergence time to nash equilibria. *Automata, Languages and Programming*.

Farges, J.L., I. Khoudour, J.B. Lesort. 1990. Prodyn: on site evaluation. *Third International Conference on Road Traffic Control, 1990.*. 62–66.

Flath, C. M., J. P. Ilg, S. Gottwalt, H. Schmeck, C. Weinhardt. 2014. Improving electric vehicle charging coordination through area pricing. *Transportation Science* **48**(4) 619–634.

Foerster, J., I. A. Assael, N. de Freitas, S. Whiteson. 2016. Learning to communicate with deep multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, vol. 29.

Foster, J. M., M. C. Caramanis. 2013. Optimal power market participation of plug-in electric vehicles pooled by distribution feeder. *IEEE Transactions on Power Systems* **28**(3) 2065–2076.

Fotakis, D., S. Kontogiannis, E. Koutsoupias, M. Mavronicolas, P. Spirakis. 2002. The structure and complexity of nash equilibria for a selfish routing game. *Automata, Languages and Programming*. 123–134.

Fotakis, D., S. Kontogiannis, P. Spirakis. 2005. Selfish unsplittable flows. *Theoretical Computer Science* **348**(2) 226–239.

Frank, H. 1969. Shortest paths in probabilistic graphs. *Operations Research* **17**(4) 583–599.

Gao, S., I. Chabini. 2006. Optimal routing policy problems in stochastic time-dependent networks. *Transportation Research Part B* **40**(2) 93–122.

Gareau, J. C, E. Beaudry, V. Makarenkov. 2019. An efficient electric vehicle path-planner that considers the waiting time. *27th ACMSIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '19)*. 389–397.

Gendreau, M., G. Ghiani, E. Guerriero. 2015. Time-dependent routing problems: A review. *Computers & Operations Research* **64** 189–197.

Gerding, E. H., V. Robu, S. Stein, David C. Parkes, A. Rogers, N. R. Jennings. 2011. Online mechanism design for electric vehicle charging. *10th International Conference on Autonomous Agents and Multiagent Systems 2011*. 761–768.

Ghosh, A., V. Aggarwal. 2018. Control of charging of electric vehicles through menu-based pricing. *IEEE Transactions on Smart Grid* **9**(6) 5918–5929.

Goemans, M.X., L. Li, V.S. Mirrokni, M. Thottan. 2006. Market sharing games applied to content distribution in ad hoc networks. *IEEE Journal on Selected Areas in Communications* **24**(5) 1020–1033.

Groves, T. 1973. Incentives in teams. *Econometrica* **41**(4) 617–631.

Guo, Q., O. Wolfson. 2018. Probabilistic spatio-temporal resource search. *GeoInformatica* **22**(1) 75–103.

Gupta, J K., M. Egorov, M. Kochenderfer. 2017. Cooperative multi-agent control using deep reinforcement learning. *Autonomous Agents and Multiagent Systems*. 66–83.

Hall, R. W. 1988. The fastest path through a network with random time-dependent travel times. *Transportation Science* **20**(3) 182–188.

Hall, R. W. 1996. Route choice and advanced traveler information systems on a capacitated and dynamic network. *Transportation Research Part C: Emerging Technologies* **4**(5) 289–306.

Hopcroft, J.E., J.T. Schwartz, M. Sharir. 1984. On the complexity of motion planning for multiple independent objects; pspace- hardness of the "warehouseman's problem". *The International Journal of Robotics Research* **3**(4) 76–88.

Hopcroft, J.E., J.T. Schwartz, M. Sharir. 2008. Exploiting subgraph structure in multi-robot path planning. *JAIR* **31** 497–542.

Jafari, E., S. D. Boyles. 2017. Multicriteria stochastic shortest path problem for electric vehicles. *Networks and Spatial Economics* **17**(3) 1043–1070.

Janson, Bruce N. 1991. Dynamic traffic assignment for urban road networks. *Transportation Research Part B: Methodological* **25**(2) 143–161.

Jossé, G, K. A. Schmid, M. Schubert. 2015. Probabilistic resource route queries with reappearance. *Proceedings of the 18th International Conference on Extending Database Technology*. 445–456.

Karfopoulos, E. L., N. D. Hatziargyriou. 2013. A multi-agent system for controlled charging of a large population of electric vehicles. *IEEE Transactions on Power Systems* **28**(2) 1196–1204.

Koutsopoulos, H.N., T. Lotan. 1989. Effectiveness of motorist information systems in reducing traffic congestion. *Conference Record of papers presented at the First Vehicle Navigation and Information Systems Conference (VNIS '89)*. 275–281.

Koutsoupias, E., C. Papadimitriou. 1999. Worst-case equilibria. *STACS 99*. 404–413.

Kullman, N., M. Cousineau, J. Goodson, J. Mendoza. 2022. Dynamic ridehailing with electric vehicles. *Transportation Science* **56**(3) 775–794.

Kullman, N., J. C. Goodson, J. E. Mendoza. 2021. Electric vehicle routing with public charging stations. *Transportation Science* **55**(3) 637–659.

Li, Q., F. Gama, A. Ribeiro, A. Prorok. 2020. Graph neural networks for decentralized multi-robot path planning. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 11785–11792.

Limmer, S., T. Rodemann. 2017. Multi-objective optimization of plug-in electric vehicle charging prices. *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. 1–8.

Limmer, S., T. Rodemann. 2019. Peak load reduction through dynamic pricing for electric vehicle charging. *International Journal of Electrical Power & Energy Systems* **113** 117–128.

Lowe, R., Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, I. Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. *Autonomous Agents and Multiagent Systems*. 6382–6393.

Ma, Z., D. S. Callaway, I. A. Hiskens. 2013. Decentralized charging control of large populations of plug-in electric vehicles. *IEEE Transactions on Control Systems Technology* **21**(1) 67–78.

Mahmassani, H. S., R. Jayakrishnan. 1991. System performance and user response under real-time information in a congested traffic corridor. *Transportation Research Part A: General* **25**(5) 293–307.

Malanchini, I., M. Cesana, N. Gatti. 2013. Network selection and resource allocation games for wireless access networks. *IEEE Transactions on Mobile Computing* **12**(12) 2427–2440.

Martins, E. Q. V. 1984. On a multicriteria shortest path problem. *European Journal of Operational Research* **16**(2) 236–245.

Masher, Ross D.W. Wong P.J. Tuan P.L. Zeidler A. Peracek S., D.P. 1975. Guidelines for design and operating of ramp control systems. Tech. rep., NCHRP 3-22, SRI Project 3340, Standford Research Institute, SRI, Menid Park, California, USA.

McKinsey. 2020. The road ahead for e-mobility. https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/the-road-ahead-for-e-mobility. Last accessed: 02.25.2021.

Milchtaich, I. 1996. Congestion games with player-specific payoff functions. *Games and Economic Behavior* **13**(1) 111–124.

Miller-Hooks, E. D., Hani S. Mahmassani. 1998. Least possible time paths in stochastic,time-varying networks. *Computers & Operations Research* **25**(12) 1107 – 1125.

Miller-Hooks, E. D., Hani S. Mahmassani. 2000. Least expected time paths in stochastic, time-varying transportation networks. *Transportation Science* **34**(2) 198–215.

Monderer, D., L. S. Shapley. 1996. Potential games. *Games and Economic Behavior* **14**(1) 124–143.

Myersdorf, D. 2020. Ultra-fast charging batteries and the cure for ev charge anxiety. https://www.intelligenttransport.com/transport-articles/112928/charging-anxiety/. Last accessed: 02.25.2021.

Nie, Y., X. Wu. 2009. Shortest path problem considering on-time arrival probability. *Transportation Research Part B* **43**(6) 597–613.

Nisan, N., A. Ronen. 2007. Computationally feasible VCG mechanisms. *Journal of Artificial Intelligence Research* **29** 19–47.

Nisan, N., T. Roughgarden, E. Tardos, V. V. Vazirani. 2007. *Algorithmic Game Theory*. Cambridge University Press.

Paccagnan, D., R. Chandan, B. L. Ferguson, J. R. Marden. 2021. Optimal taxes in atomic congestion games. *ACM Transactions on Economics and Computation (TEAC)* **9**(3) 1–33.

Parkes, D. C., S. Singh, D. Yanovsky. 2004a. Approximately efficient online mechanism design. *Advances in Neural Information Processing Systems*.

Parkes, D. C., M. P. Wellman. 2015. Economic reasoning and artificial intelligence. *Science* **349**(6245) 267–272.

Parkes, D. C, D. Yanovsky, S. Singh. 2004b. Approximately efficient online mechanism design. L. Saul, Y. Weiss, L. Bottou, eds., *Advances in Neural Information Processing Systems*, vol. 17.

Parkes, David C, Satinder Singh. 2003. An MDP-based approach to online mechanism design. *Advances in Neural Information Processing Systems*.

Pavlis, Y., M. Papageorgiou. 1999. Simple decentralized feedback strategies for route guidance in traffic networks. *Transportation Science* **33**(3) 264–278.

Pigou, A. 1920. *The Economics of Welfare*. Macmillan.

Polycarpou, M.M., Yang Y., K. M. Passino. 2001. A cooperative search framework for distributed agents. *Proceeding of the 2001 IEEE International Symposium on Intelligent Control (ISIC '01) (Cat. No.01CH37206)*.

Polychronopoulos, G. H., J.N. Tsitsiklis. 1996. Stochastic shortest path problems with recourse. *Networks* **27** 133– 143.

Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. 1st ed. John Wiley & Sons, Ltd. doi:10.1002/9780470316887.

Rigas, E. S., E. Gerding, S. Stein, S. D. Ramchurn, N. Bassiliades. 2020. Mechanism design for efficient allocation of electric vehicles to charging stations. *11th Hellenic Conference on Artificial Intelligence*. 10–15.

Roberts, K. 1979. The characterization of implementable choice rules. JJ-Laffont (ed.), ed., *Aggregation and Revelation of Preferences*. Noth Holland Publishing Company.

Rosenthal, R. W. 1973. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory* **2**(1) 65–67.

Roughgarden, T., E. Tardos. 2002. How bad is selfish routing? *J. ACM* **49**(2) 236–259.

Sartoretti, G., J. Kerr, Y. Shi, G. Wagner, T. K. S. Kumar, S. Koenig, H. Choset. 2019. Primal: Pathfinding via reinforcement and imitation multi-agent learning. *IEEE Robotics and Automation Letters* **4**(3) 2378–2385.

Schmoll, S., M. Schubert. 2018. Dynamic resource routing using real-time dynamic programming. *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. 4822–4828.

Schmoll, S., M. Schubert. 2020. Semi-markov reinforcement learning for stochastic resource collection. *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*.

Shao, W., F. D. Salim, T. Gu, N. Dinh, J. Chan. 2018. Traveling officer problem: Managing car parking violations efficiently using sensor data. *IEEE Internet of Things Journal* **5**(2) 802–810.

Sharon, G., R. Stern, A. Felner, N. R. Sturtevant. 2015. Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence* **219** 40–66.

Sharon, G., R. Stern, M. Goldenberg, A. Felner. 2011. Complete algorithms for cooperative pathfinding problems. *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume One*. IJCAI, 668–673.

Silver, D. 2005. Cooperative pathfinding. *Proceedings of the First AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. AAAI Press, 117–122.

Sohet, B., Y. Hayel, O. Beaude, A. Jeandin. 2021. Coupled charging-and-driving incentives design for electric vehicles in urban networks. *IEEE Transactions on Intelligent Transportation Systems* **22**(10) 6342–6352.

Song, N., D. Teneketzis. 2004. Discrete search with multiple sensors. *Mathematical Methods of Operations Research* **60** 1–13.

Standley, T. 2010. Finding optimal solutions to cooperative pathfinding problems **24** 173–178.

Standley, T., R. Korf. 2011. Complete algorithms for cooperative pathfinding problems. *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*. AAAI Press, 668–673.

Stein, S., M. Ochal, I. A. Moisoiu, E. Gerding, R. Ganti, T. He, T. L. Porta. 2020. Strategyproof reinforcement learning for online resource allocation. *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*. 1296–1304.

Stern, R., N. R. Sturtevant, A. Felner, S. Koenig, H. Ma, T. T. Walker, J. Li, D. Atzmon, L. Cohen, T. K. S. Kumar, R. Bartak, E. Boyarski. 2019. Multi-agent pathfinding: Definitions, variants, and benchmarks. *Twelfth Annual Symposium on Combinatorial Search*. 151–159.

Sutton, R. S., D. Precup, S. Singh. 1999. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence* **112**(1) 181–211.

Sweda, T. M., I. S. Dolinskaya, D. Klabjan. 2017. Adaptive routing and recharging policies for electric vehicles. *Transportation Science* **51**(4) 1326–1348.

Tang, H., M. Kerber, Q. Huang, L. Guibas. 2013. Locating lucrative passengers for taxicab drivers. *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*.

Tang, X., Z. Qin, F. Zhang, Z. Wang, Z. Xu, Y. Ma, H. Zhu, J. Ye. 2019. A deep value-network based approach for multi-driver order dispatching. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1780–1790.

Tushar, W., W. Saad, H. V. Poor, D. B. Smith. 2012. Economics of electric vehicle charging: A game theoretic approach. *IEEE Transactions on Smart Grid* **3**(4) 1767–1778.

Verroios, V., V. Efstathiou, A. Delis. 2011. Reaching available public parking spaces in urban environments using ad hoc networking. *2011 IEEE 12th International Conference on Mobile Data Management*.

Vickrey, W. 1961. Counter speculation, auctions, and competitive sealed tenders. *The Journal of finance* **16**(1) 8–37.

Wang, Y., W.Y. Szeto, K. Han, T. L. Friesz. 2018a. Dynamic traffic assignment: A review of the methodological advances for environmentally sustainable road transportation applications. *Transportation Research Part B: Methodological* **111** 370–394.

Wang, Z., Z. Qin, X. Tang, J. Ye, H. Zhu. 2018b. Deep reinforcement learning with knowledge transfer for online rides order dispatching. *2018 IEEE International Conference on Data Mining (ICDM)*. 617–626.

Wardrop, J. G. 1952. Road paper. some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers* **1**(3) 325–362.

Wong, E., F. Bourgault, T. Furukawa. 2005. Multi-vehicle bayesian search for multiple lost targets. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*.

Wu, C., H. Mohsenian-Rad, J. Huang. 2012. Vehicle-to-aggregator interaction game. *IEEE Transactions on Smart Grid* **3**(1) 434–442.

Wunderlich, K.E., M.H. Hardy, J.J. Larkin, V.P. Shah. 2001. On-time reliability impacts of advanced traveler information services (atis): Washington, dc case study. Tech. rep., Mitretek Systems.

Xu, Z., Z. Li, Q. Guan, D. Zhang, Q. Li, J. Nan, C. Liu, W. Bian, J. Ye. 2018. Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 905–913.

Yoon, S., Y. Choi, J. Park, S. Bahk. 2016. Stackelberg-game-based demand response for at-home electric vehicle charging. *IEEE Transactions on Vehicular Technology* **65**(6) 4172–4184.

Zelinsky, A. 1992. A mobile robot exploration algorithm. *IEEE Transactions on Robotics and Automation* **8**(6) 707–717.

Zhang, L., Y. Li. 2016. A game-theoretic approach to optimal scheduling of parking-lot electric vehicle charging. *IEEE Transactions on Vehicular Technology* **65**(6) 4068–4078.

Zhang, X., D. Rey, S.T. Waller, C. Nathan. 2019. Range-constrained traffic assignment with multi-modal recharge for electric vehicles. *Networks and Spatial Economics* **19** 633–668.

3 Electric Vehicle Charging Station Search in Stochastic Environments

This chapter is based on an article published as:

Guillet M., Hiermann G., Kroëller A., Schiffer M. (2022). Electric Vehicle Charging Station Search in Stochastic Environments. *Transportation Science* Vol. 56(2), 483-500. 483–500. Doi: 10.1287/trsc.2021.1102

Abstract

Electric vehicles are a central component of future mobility systems as they promise to reduce local noxious and fine dust emissions and CO_2 emissions, if fed by clean energy sources. However, the adoption of electric vehicles so far fell short of expectations despite significant governmental incentives. One reason for this slow adoption is the drivers' perceived range anxiety, especially for individually owned vehicles. Here, bad user-experiences, e.g., conventional cars blocking charging stations or inconsistent real-time availability data, manifest the drivers' range anxiety. Against this background, we study stochastic search algorithms, that can be readily deployed in today's navigation systems in order to minimize detours to reach an available charging station. We model such a search as a finite horizon Markov decision process and present a comprehensive framework that considers different problem variants, speed-up techniques, and three solution algorithms: an exact labeling algorithm, a heuristic labeling algorithm, and a rollout algorithm. Extensive numerical studies show that our algorithms significantly decrease the expected time to find a free charging station while increasing the solution quality robustness and the likelihood that a search is successful compared to myopic approaches.

1. Introduction

All around the world, governments and companies try to foster the adoption of electric vehicles (EVs), which are seen as a central component of future sustainable mobility systems that can play a major role in reducing noise, noxious, and carbon emissions, if fed by clean energy sources. While governments introduced national programs to support individual EV purchase, e.g., by means of ecological bonuses (France), tax exemption (Germany), or subsidies (Netherlands), private companies invested in public charging infrastructure (Bensasson 2019, Fröhlich 2019), and major logistics service providers progressively integrated EVs to realize sustainable transportation (Amazon 2019, DHL 2019). Yet the market uptake for EVs fell short of expectation in most countries. One of the main obstacles remaining, in particular for privately-owned EVs, is the customers' perceived range-anxiety, triggered by missing charging infrastructure and incomplete or non-standardized information about charging services available. Hence, facilitating a reliable EV charging process is crucial to decrease range anxiety in order to invigorate the adoption of EVs (Bonges and Lusk 2016).

Different stakeholders focus on different measures to facilitate a reliable charging experience for users. At strategic level, municipalities try to facilitate reliable charging options by improving the overall charging infrastructure, e.g., by building new charging stations or increasing the capacity of existing ones. Further, enforcing new pricing schemes that support a higher turnover rate of (re)charged vehicles may help to save additional capacities (Bonges and Lusk 2016). However, infrastructure investments that require long planning lead times and pricing schemes can hardly be realized without standardization and agreements between charging station operators. At operational level, some online map services exist and aim to help EV drivers to locate available charging stations. Unfortunately, such services struggle with data inaccuracy, e.g., an incomplete coverage of real-time status data. Even worse, charging stations can be blocked by non-charging vehicles (so-called "ICEing") without reflecting such inaccessibility in status data. An empirical study in Berlin revealed a high correlation between such inaccessibility and parking availability, showing that in areas without available parking it is three times more likely for a charging station to be illegitimately blocked than legitimately used (see Table 1). Drivers may find a selected and seemingly free station

Table 1: Parking availability on data accuracy

Legal parking spots available in immediate vicinity	≥ 2	1	0
Blocked by charging EV	5%	14%	10%
Blocked without connection (ICEing)	1%	9%	34%
Available	94%	77%	56%

Internal TomTom study in Berlin, 2019. The study shows how often charge points were legitimately used (i.e.,"blocked" and actually used), actually available (correct "free"), or illegitimately blocked by a vehicle. The data relates to the amount of free parking spots found in the immediate vicinity. For most of the stations, real-time availability data was not available.

to be blocked when arriving there. Such negative user experiences may require additional detours with an already depleted battery in order to find a suitable charging station and thus may even increase a driver's range anxiety. To this end, a reliable stochastic search algorithm that helps to route drivers to an available charging station as convenient as possible constitutes a valuable algorithmic component for today's map and navigation services and may help to reduce the drivers' range anxiety.

The goal of this paper is to develop such an algorithmic solution that can be readily deployed as a built-in implementation in today's navigation systems and map services. In the remainder of this section, we first review the related literature, before we detail our contribution and elaborate the organization of this paper.

1.1. Literature review

We now survey literature on EV routing with uncertain charging stations before we focus on related search problems with stochastic resources.

Sweda et al. (2017) were the first to study probabilistic charging station availability in a multi-stop shortest path problem under uncertainty. They used a dynamic programming approach on a grid network and considered waiting times at unavailable stations. Kullman et al. (2021) extended this work by modeling an electric vehicle routing problem with uncertain charging at public charging stations as a Markov decision process (MDP). Given the large action and state spaces, they proposed an approximate stochastic dynamic programming approach with a look-ahead procedure (Goodson et al. 2017). Jafari and Boyles (2017) studied a multicriteria stochastic shortest path problem for EVs, in which stochasticity relates to travel time and energy consumption.

Focusing on related stochastic search problems, Arndt et al. (2016) studied a probabilistic routing problem for on-street parking search with parking spots as stochastic resources and incorporated user preferences for stops closer to their destination. Besides showing the problem's NP-completeness, they proposed a branch-and-bound algorithm for small problem spaces. More generally, Guo and Wolfson (2018) studied a probabilistic spatio-temporal resource search problem, in which resources have a general usage cost but the resource seeker is not allowed to wait for an occupied resource to become available again. Contrary to Arndt et al. (2016) where resource observations are persistent during the search, stations can become available again after a defined time threshold. Guo and Wolfson (2018) proposed a value-iteration solution procedure that remains tractable by making a fast recovery assumption at the instance level, which keeps the state space small. Schmoll and Schubert (2018) studied a dynamic resource routing problem under reliable real-time information, which requires fast (re)computations as resources frequently change their occupancy state. Assuming a large problem space, they used real-time dynamic programming to maintain utility values for likely states to determine the best action on-the-fly.

As can be seen, related research on stochastic charging station search is still scarce.

Approaches that are specifically tailored to EV charging station search focus on finding best cost paths between an origin and a destination rather than on an open search with an a-priori unknown destination. Further these approaches lack the consideration of sufficient real-world problem variants and heterogeneous charging station characteristics. Literature on stochastic resource search focuses on an open search concept but lacks EV and charging station specific characteristics. Moreover, the algorithmic solutions that have been proposed in this domain remain limited to a pure academic interest on artificial and small instances. Accordingly, these approaches cannot be embedded into real-time environments.

1.2. Contribution

With this work, we close the research gaps outlined above by providing a profound algorithmic framework that covers a wide range of problem variants in the context of stochastic EV charging station search, which can readily be deployed into real-time environments for online optimization purposes. Specifically, our contribution is fourfold. First, we formalize the problem of stochastic EV charging station search as a novel stochastic search problem. We model this problem as a discrete finite horizon MDP and consider two objectives: minimizing i) the time until charging, and ii) the time until completion in case of heterogeneous charging times. Moreover, we present different problem variants in which a driver may or may not wait at occupied stations and charging stations might be homogeneous or heterogeneous. Second, we prove the complexity of this problem class and show that it is NP-hard. Third, we develop an algorithmic tool chain that consists of three algorithms: an exact labeling algorithm for which we present a cost decomposition of the Markovian policy to derive effective dominance rules; a heuristic variant of this labeling algorithm which is amenable for real-time application; and a rollout algorithm. Additionally, we present multiple speed-up mechanisms, e.g., reduced action spaces and sharpened dominance relations. Fourth, we provide extensive numerical studies that base on real-world data for the cities of Berlin and San Francisco. Our results show that compared to myopic approaches, our search algorithms decrease the average time spent finding an available station by up to 44%. We further benchmark the performance of our heuristic algorithms against the exact algorithm and show that a combined, case-dependent usage of both allows for effective real-time application.

1.3. Organization

The remainder of this paper is as follows. In Section 2, we introduce the stochastic charge pole search (SCPS) problem. Section 3 formalizes our problem as an MDP and consecutively develops the corresponding algorithmic framework. In Section 4, we describe our case study and the experimental design. Section 5 discusses our numerical results. Section 6 concludes this paper and provides an outlook on future research. To keep this paper concise, we shift proofs to Appendix A.

2. Problem definition and representation

In this section, we introduce the SCPS problem for which we specify four problem variants, each corresponding to a distinct real-world scenario.

2.1. Problem setting

We focus on a routing problem with stochastic charging station availability, where a driver starts at a given location and seeks to find an unoccupied charging station to recharge her vehicle. Her objective is to minimize her total expected cost, which consists of the driving time during the search and additional factors, e.g., the time until the charging is complete, or the time spent walking from the charge point to the actual destination. We consider this search to be spatially and temporally bounded to account for the driver's limited time budget and penalize unsuccessful searches with a high termination cost to model the resulting discomfort.

Formally, we define this problem on a directed and complete graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ that consists of a set of vertices $\mathcal{V}$ and a set of arcs $(v, v') \in \mathcal{A}$. Each vertex $v \in \mathcal{V}$, except the vertex where the search starts, represents a charging station. The driver starts her search at a designated start vertex $v_0 \in \mathcal{V}$ at time 0, with a maximum search time of $\overline{T}$. We assume that the driver is willing to charge at any vertex in $\mathcal{V}$ and consider a limited search radius by restricting $\mathcal{V}$ to the stations within a maximum distance $\overline{S}$ around the start vertex. Driving from v to v' takes $t_{v,v'} \geq 0$ units of time. We model the availability of a station $v \in \mathcal{V}$ at time $t \in [0, \overline{T}]$ as a visit-dependent binary random variable $a_v \in \{0, 1\}$, which the driver observes by visiting v. Without prior knowledge, i.e., when visiting a station for the first time, the probability that the station is free ($a_v = 1$) is a constant p_v. Table 2 summarizes this notation.

We introduce two time-based penalties: γ_v is the time-equivalent usage cost for using pole v if it is available upon arrival; β_v denotes the cost for unsuccessfully terminating the search at v in case v is occupied, which may happen if all vertices in $\mathcal{V}$ have been unsuccessfully visited, or if it is impossible to visit another candidate within the remaining time budget $\overline{T}$.

We define a solution to our problem as an ordered sequence of charging station visits $C = (v_0, \ldots, v_n)$. In practice, a driver that uses this solution starts at v_0 and visits the

Table 2: Notation used to define the SCPS

$\mathcal{G} = (\mathcal{V}, \mathcal{A})$	Network graph
$\overline{S}$	Maximal distance allowed between any vertex and the start vertex v_0
$\overline{T}$	Maximal overall driving time
W_v	Expected waiting time at an occupied charging station v
Y_v	Expected charging time at charging station v
a_v	Binary random variable modelling the availability of v
p_v	Initial probability that charging station v is available before any visit
$t_{v,v'}$	Driving time on arc (v, v')
β_v	Termination penalty cost at an occupied station v
γ_v	Termination cost at an available station v

charging stations in the given sequence up to the first available charging station. Accordingly, our search terminates either successfully at any $v \in C$, with a total cost of driving time until that vertex plus extra cost γ_v, or unsuccessfully at v_n, incurring driving time along C plus cost β_{v_n}. We denote by α the expected cost of solution C.

2.1.1. Problem setting variants:

In practice, different system characteristics and user preferences may require to solve different problem variants, e.g., some drivers may want to find a charging station as fast as possible without considering station heterogeneity, while others may want to finish charging as quickly as possible. Such differences can be incorporated in our problem through the generic penalties β_v and γ_v, and an additional parameter ω defining whether it is permitted to finish the search by waiting at a station ($\omega = 1$) or not ($\omega = 0$). We identify four different problem variants that reflect the most common use cases in practical applications. Table 3 summarizes these variants and states the respective realizations as a triple $(\beta_v, \gamma_v, \omega)$. In variant $\neg W/\neg C$, the driver is neither allowed to wait at stations nor do stations have (heterogeneous) usage costs. To this end, we consider a constant penalty cost $\overline{\beta}$ for an unsuccessful terminated search. While stations remain homogeneous, the driver is allowed to wait in variant $W/\neg C$. We assume that an expected waiting duration W_v is known for every station v, and to be a constant independent of the arrival time and earlier observations. In variants $\neg W/C$ and W/C, stations have heterogeneous charging durations and the driver seeks to minimize her total time to finish charging up to a preferred charge level. While the driver is not allowed to wait in $\neg W/C$ (i.e., the constant penalty cost $\overline{\beta}$ is induced in case of failure), this constraint is relaxed in W/C.

2.2. Discussion

While our model captures the essential characteristics of the underlying real-world problem, some comments are in order. First, we differentiate charging stations based on the charging duration in problem variants $\neg W/C$ and W/C. The model is however not restrictive and allows charging stations to be heterogeneous with respect to any other criteria, including walking distance to a destination location or the price for charging. Yet one needs to convert non time-based usage costs (e.g., charging prices) to time-based costs to ensure search-cost compatibility. Second, we assume an occupied station to remain occupied during the search for most problem variants. This seems to be a plausible assumption based on the large

Table 3: Summary of problem setting variants and parameters $(\beta_v, \gamma_v, \omega)$

	No waiting allowed ($\neg W$)	Waiting allowed (W)
Charging time insensitive ($\neg Y$)	$\neg W/\neg C = (\overline{\beta}, 0, 0)$	$W/\neg C = (W_v, 0, 1)$
Charging time sensitive (Y)	$\neg W/C = (\overline{\beta}, Y_v, 0)$	$W/C = (W_v + Y_v, Y_v, 1)$

difference between typical search times (minutes) and charging durations (hours) in an urban setting. However, we also present a problem variant with recovering probabilities in Section 3.4 and discuss its impact in Section 5. Third, our basic model is agnostic of the energy spent during the search. Here, a similar argument holds: given that a search typically covers only a small distance radius compared to a vehicle's range, we assume this effect to be negligible. Nevertheless, we present a problem variant which considers the energy spent during the search. Our results for this problem variant show no significant impact of the respective energy consumption and thus confirm our assumption.

3. Methodology

We now introduce an MDP representation for the SCPS problem (Section 3.1) and a corresponding algorithmic framework. As the SCPS problem is NP-hard (see Appendix B) we develop one exact and two heuristic algorithms. In the remainder, we first focus on an exact and a heuristic labeling algorithm in Section 3.2, before we present a rollout algorithm in Section 3.3. We then show how these algorithms must be modified to account for time-dependent recovery probability functions (Section 3.4) and search-related energy consumption (Section 3.5). For the sake of conciseness, we detail these algorithms only for the practically most relevant problem variant, $\neg W/C$, i.e., the charging-time sensitive variant that proscribes waiting. We describe minor required methodological changes for the remaining problem variants $\neg W/\neg C$, $W/\neg C$, and W/C in Appendix C.2.

3.1. Markov decision process

To model the SCPS problem as a finite-horizon MDP with multiple decisions over a time-budgeted process, we use additional notation as summarized in Table 4.

Table 4: Notation used to define the MDP

$\mathcal{S}$	State space
$\mathcal{U}$	Action space
C	Ordered sequence of station visits
x	State
$u(x)$	Action choosen in state $x \in \mathcal{S}$
w	Binary variable indicating whether to wait at the current station in state $x \in \mathcal{S}$
$d(u, u(x))$	Immediate cost induced by taking action u in state $x \in \mathcal{S}$
π	Policy
V^π	Value function, based on policy π
V^*	Optimal value function
a	Binary variable modeling the availability of the last visited station
δ	Binary variable indicating whether the selected station v in $x = (C, 0)$ is already contained in C
$\mathcal{W}$	Permissible values for waiting decisions
$t(\pi)$	Driving time follwing policy π
$\rho(\pi, k)$	Probability that at least one of the k first stations of policy π is available
$\alpha(\pi)$	Solution cost for following policy π

3.1.1. Model variables and transition functions:

We define a state x as a tuple (C, a), where C is an ordered sequence of stations $C = (v_0, ..., v_k)$ that have already been visited, with the driver being located at v_k. Let a be the binary realization of the availability of v_k, indicating whether v_k is free (a=1) or occupied (a=0). Then, the state space results to

$$\mathcal{S} = \{(C, a) : \ C = (v_0, \ldots, v_k), v_j \in \mathcal{V} \ \forall j, a \in \{0, 1\}\}.$$

We denote by $u(x) = (v, w)$ a possible action taken in state $x \in \mathcal{S}$ to transition to state x', with w being a binary that indicates whether to wait at the current station ($w = 1$) or not ($w = 0$), and v stating the next station to visit. We denote by $\mathcal{W}$ the set of values taken by w, such that $\mathcal{W} = \{0\}$ ($\mathcal{W}^0$) for problem variants without waiting and $\mathcal{W} = \{0, 1\}$ ($\mathcal{W}^1$) for problem variants that allow to wait at a charging station. Since we assume a station's availability status to be persistent during the search, we further assume that for problem variants without waiting, each station should be visited at most once, because a station cannot be used during the search's time horizon once it is occupied. For problem variants with waiting, the driver can use an occupied station by paying the price β_v, i.e., waiting. In these settings, revisiting a station when knowing that it is occupied is only reasonable if the driver decides to wait. As the search terminates in this case as soon as the drivers decides to wait, only the last station may be visited twice. Actions are only taken at occupied stations: if $a = 1$, x is a termination state, and the search finishes as the driver charges at the found unoccupied station. Let $\widetilde{\mathcal{V}}(C)$ be the restricted set of charging station vertices, reachable in less than $\overline{T} - \sum_{i=0}^{k-1} t_{v_i, v_{i+1}}$. Then, the action space for a state $x = (C, 0)$ results to

$$\mathcal{U}(x) = \{(v, w) : w \in \mathcal{W}, v \in \widetilde{\mathcal{V}}, (\mathcal{W} = \mathcal{W}^0 \wedge \delta = 0) \vee (\mathcal{W} = \mathcal{W}^1 \wedge (\delta = 0 \vee (\delta = 1 \wedge w = 1)))\}, \tag{3.1}$$

with δ being the binary variable that indicates whether v is already included in C ($\delta = 1$) or not ($\delta = 0$). If $\widetilde{\mathcal{V}}(C)$ is empty, no more station can be reached under the given time constraints and we refer to this state as a forced termination state.

We define the function $p_t(x' \mid x, u)$ to describe the probability that following action $u \in \mathcal{U}(x)$ from state $x \in \mathcal{S}$ would result in state $x' \in \mathcal{S}$, such that $\sum_{x' \in \mathcal{S}} p_t(x'|x, u) = 1$ and consider two cases:

1. If the selected action is to wait, then the driver stays at the current station. The transition is deterministic and $p_t((C, 1)|(C, 0), (v_k, 1)) = 1$ for a path C ending in v_k.

2. If the selected action is to continue searching, the selected next station v is either available or occupied with respect to $\tilde{p}_v(C)$. Hence $p_t((C', 1)|(C, 0), (v, 0)) = \tilde{p}_v(C)$ and $p_t((C', 0)|(C, 0), (v, 0)) = 1 - \tilde{p}_v(C)$, where C' is the sequence C extended by v.

We denote by $d(x, u)$ the immediate induced cost for taking action $u = (v, w)$ in state $x = (C, a)$, which depends on the realized availability a at the last station v_k and the

respective action

$$d(x,u) = (1-a)w\beta_{v_k} + a\gamma_{v_k} + (1-w)(1-a)t_{v_k,v}. \tag{3.2}$$

We note that if $a = 1$, the driver charges at station v_k, and $d(x,u)$ corresponds to the station usage cost γ_{v_k}. If $a = 0$ and the driver waits at the station, the search terminates and cost β_{v_k} results. If $a = 0$ and the search continues at the next station v such that $v_{k+1} := v$, the cost results to the travel time $t_{v_k,v_{k+1}}$.

Finally, we define a policy π as a function mapping a state $x \in \mathcal{S}$ to an action $\pi(x) \in \mathcal{U}(x)$. Accordingly, π implicitly describes a search path $C(\pi) = (v_0, ..., v_n)$ with $\pi(v_i, 0) = (v_{i+1}, 0)$ $\forall i = 0, \ldots, n-1$ and state $x_n = (C(\pi), a)$ that terminates the search at vertex v_n, either because the driver runs out of time or because she decides to wait with $\pi(x_n) = (v_n, 1)$. We refer to $C(\pi)$ as C and to $\tilde{p}_v(C)$ as $\tilde{p}_v$ to keep the notation concise.

3.1.2. Cost function:

We now analyze the cost function $V^\pi(C,a)$ that describes the expected cost for following a policy π, from a start state (C,a). Then $V^\pi(x_0)$, with $x_0 = (v_0, 0)$ represents the expected cost for the driver when following policy π from starting at vertex v_0 and the objective is to find a policy π that minimizes $V^\pi(x_0)$. Then, the cost function $V^\pi(C,a)$ can be expressed as follows

$$V^\pi(C,a) = a\gamma_{v_k} + (1-a)\left[w\beta_{v_k} + (1-w)(t_{v_k,v_{k+1}} + \tilde{p}_{v_{k+1}}V^\pi(C',1) + (1-\tilde{p}_{v_{k+1}})V^\pi(C',0))\right], \tag{3.3}$$

with $C = (v_0, \ldots, v_k)$, $u(C,a) = (v_{k+1}, w)$, and $C' = C \cup \{v_{k+1}\}$. For both realizations of a, Equation 3.3 can be simplified, as follows

$$V^\pi(C,1) = \gamma_{v_k}\ , \tag{3.4}$$

$$V^\pi(C,0) = w\beta_{v_k} + (1-w)\left[t_{v_k,v_{k+1}} + (1-\tilde{p}_{v_{k+1}})V^\pi(C',0) + \tilde{p}_{v_{k+1}}\gamma_{v_{k+1}}\right]\ . \tag{3.5}$$

3.1.3. Cost function expansion:

We now expand V^π to derive an explicit evaluation of the expected cost. To simplify notation, we use $C_{[i:j]}$ to denote a sub-sequence $(v_i, \ldots, v_j)$ of a sequence $C = (v_0, ..., v_i, ..., v_j, ..., v_n)$.

Proposition 1. *Let $C = (v_0, \ldots, v_n)$. Then, the cost for being in state $x_k = (C_{[0:k]}, 0)$, with $k < n$, following policy π until the termination state $x_n = (C,a)$ expands as follows*

$$\begin{aligned} V^\pi(x_k) = &\prod_{i=k}^{n}(1-\tilde{p}_{v_i}(C_{[k:i-1]}))\beta_{v_n} + \sum_{i=k}^{n-1}\left[t_{v_i,v_{i+1}}\prod_{j=k}^{i}(1-\tilde{p}_{v_j}(C_{[k:j-1]}))\right] \\ &+ \sum_{i=k}^{n}\left[\gamma_{v_i}\tilde{p}_{v_i}(C_{[k:i-1]})\prod_{j=k}^{i-1}(1-\tilde{p}_{v_j}(C_{[k:i-1]}))\right]\ . \end{aligned} \tag{3.6}$$

Given that policies encode solutions, we can express the solution cost α for a policy, and

set $\alpha(\pi) = V^\pi(x_0)$. Following Equation 3.6 and Proposition 1, this yields

$$\begin{aligned}\alpha(\pi) = &\prod_{i=0}^{n}(1-\tilde{p}_{v_i}(C_{[0:i-1]}))\beta_{v_n} + \sum_{i=0}^{n-1}\left[t_{v_i,v_{i+1}}\prod_{j=0}^{i}(1-\tilde{p}_{v_j}(C_{[0:j-1]}))\right] \\ &+ \sum_{i=0}^{n}\left[\gamma_{v_i}\tilde{p_{v_i}}(C_{[0:i-1]})\prod_{j=0}^{i-1}(1-\tilde{p_{v_j}}(C_{[0:j-1]}))\right] .\end{aligned} \tag{3.7}$$

For a more concise notation, let $\bar{\rho}(\pi, k)$ be the probability that a driver fails in finding at least one free station in $C_{[0:k]}$

$$\bar{\rho}(\pi, k) = \prod_{i=0}^{k}(1-\tilde{p}_{v_i}(C_{[0:i-1]})) , \tag{3.8}$$

with $\tilde{p}_{v_i}(C_{[0:i-1]})$ denoting the likelihood that v_i is available after having visited all previous stations from v_0 to v_{i-1}, while $\rho(\pi, k) = 1 - \bar{\rho}(\pi, k)$ is the probability that she succeeds in finding at least one free station in $C_{[0:k]}$. Furthermore, let

$$A(\pi) = \sum_{i=0}^{n-1}[t_{v_i,v_{i+1}}\bar{\rho}(\pi, i)] + \sum_{i=0}^{n}\gamma_{v_i}\tilde{p}_{v_i}(C_{[0:i-1]})\bar{\rho}(\pi, i-1) . \tag{3.9}$$

Then, we rewrite Equation 3.7 as

$$\alpha(\pi) = \bar{\rho}(\pi, n)\beta_{v_n} + A(\pi) . \tag{3.10}$$

We denote by $t(\pi) = \sum_{i=0}^{n-1} t_{v_i,v_{i+1}}$ the accumulated driving time for all stations in C and note that for a feasible solution, $t(\pi) \leq \overline{T}$ holds.

3.1.4. Cost structure variants:

The cost functions $V^\pi(C, 0)$, $V^\pi(C, 1)$, and $V^\pi(x_n)$ for a termination state $x_n = (C, a)$ can be parameterized for each problem variant as introduced in Section 2.1.1 (cf. Table 3). For $\neg W/C$, it follows with C' denoting the extension of C by station v_{k+1} for non-termination states that

$$\begin{aligned}V^\pi(C, 0) &= t_{v_k,v_{k+1}} + (1-\tilde{p}_{v_{k+1}})V^\pi(C', 0) + \tilde{p}_{v_{k+1}}Y_{v_{k+1}} , \\ V^\pi(C, 1) &= Y_{v_k} , \\ V^\pi(x_n) &= (1-\tilde{p}_{v_n})\overline{\beta} + \tilde{p}_{v_n}Y_{v_n} .\end{aligned} \tag{3.11}$$

Appendix C.1 details the cost function parameterizations for the remaining problem variants.

3.2. Dynamic programming based labeling algorithms

To find an optimal solution for the SCPS problem, we develop a dynamic programming based labeling algorithm. Similar to solving multi-criteria constrained shortest path problems, we propagate partial policies in order to find an in-expectation cost-optimal policy. Herein, we use a dominance criterion to withdraw non-promising partial policies early to keep the explored search space as small as possible. Formally, we associate each partial policy π_v with a label L_v whose resources depend on the problem variant. A label $L_v = (t_v, A_v, \rho_v, \alpha_v, S_v)$ consists of the accumulated driving time t_v, the partial cost A_v (cf. Equation 3.10), the

likelihood ρ_v to successfully finish the search up to vertex v (cf. Equation 3.8), the cost α_v (cf. Equation 3.7), and the set of reachable and non-visited poles S_v.

To describe our labeling algorithm, we denote by $\mathcal{L}^a$ the set of active labels and by L_0 the initial label that corresponds to our start location. Let $\mathcal{F}_{v,v'}(L)$ be a set of resource extension functions (REFs) that expand a label L whose partial policy ends at vertex v to a label L' whose partial policy ends at vertex v'. Let $\alpha(L)$ be the cost associated with label L. We define $\delta^+(L)$ as a function that returns a set of tuples (v, v') which denotes all feasible physical successor locations $v \in \mathcal{V}$ for a label L whose partial policy ends at $v \in \mathcal{V}$.

Using this notation, Figure 1 shows a pseudo code of our dynamic programming algorithm. We initialize our list of active labels $\mathcal{L}^a$ and our so far best found solution L^* with L_0 (l.1) and start propagating labels until our search terminates when $\mathcal{L}^a$ is empty (l.2). We then process labels in $\mathcal{L}^a$ in cost increasing order (l.3). Once a label got selected for propagation, we remove it from $\mathcal{L}^a$ (l.4) and propagate it considering all of its feasible successors (l.5) using the REFs (l.6). We check whether an existing label in $\mathcal{L}^a$ dominates a newly created label L' (l.7). If this is not the case, we remove labels which are dominated by L' from $\mathcal{L}^a$ (l.8) and add L' to $\mathcal{L}^a$ respectively (l.9). Whenever the newly created label is a termination label, i.e., its corresponding state indicates that there are no feasible successors left, we check if the found label improves the so far best found label and update L^* accordingly (l.10&l.11). In the remainder of this section, we detail the REFs used to extend a label and the dominance criterion to discard a dominated label.

3.2.1. Resource extension functions:

To extend a label L which corresponds to a partial policy ending at vertex $v \in \mathcal{V}$ to a new label L' which corresponds to a partial policy ending at vertex $v' \in \mathcal{V}$, we write $L' \leftarrow \mathcal{F}_{v,v'}(L)$ and use the following set $\mathcal{F}_{v,v'}$ of REFs :

$$A_{v'} = A_v + (1 - \rho_v)(t_{v,v'} + \tilde{p}_{v'}\gamma_{v'}) \tag{3.12}$$

$$1 - \rho_{v'} = (1 - \rho_v)(1 - \tilde{p}_{v'}) \tag{3.13}$$

$$t_{v'} = t_v + t_{v,v'} \tag{3.14}$$

Figure 1: Dynamic programming based labeling algorithm.

1: $\mathcal{L}^a \leftarrow \{L_0\}$, $L^* \leftarrow L_0$
2: **while** $\mathcal{L}^a \neq \emptyset$ **do**
3: $\quad L \leftarrow$ `costMinimumLabel(`$\mathcal{L}^a$`)`
4: $\quad \mathcal{L}^a \leftarrow \mathcal{L}^a \setminus \{L\}$
5: $\quad$ **for** $(v, v') \in \delta^+(L)$ **do**
6: $\quad\quad L' \leftarrow \mathcal{F}_{vv'}(L)$
7: $\quad\quad$ **if** `isNotDominated(`$L', \mathcal{L}^a$`)` **then**
8: $\quad\quad\quad$ `dominanceCheck(`$\mathcal{L}^a, L'$`)`
9: $\quad\quad\quad \mathcal{L}^a \leftarrow \mathcal{L}^a \cup \{L'\}$
10: $\quad\quad\quad$ **if** $(\delta^+(L') = \emptyset) \wedge \alpha(L') < \alpha(L^*)$ **then**
11: $\quad\quad\quad\quad L^* \leftarrow L'$
12: **return** L^*

$$S_{v'} = S_v \setminus \{v'\} - \bigcup_{v'' \in S_v, t_{v'} + t_{v',v''} > \overline{T}} \{v''\} \tag{3.15}$$

$$\alpha_{v'} = A_{v'} + (1 - \rho_{v'})\beta_{v'} \tag{3.16}$$

Equation 3.12 propagates partial cost A_v along arc (v, v') with respect to its definition (cf. Equation 3.9) considering the arc-dependent driving time $t_{v,v'}$, the availability probability $\tilde{p}_{v'}$, and usage cost $\gamma_{v'}$ of vertex v'. Equation 3.13 propagates the success rate ρ_v (cf. Equation 3.8) based on the availability probability of vertex v'. The accumulated driving time $t_{v'}$ is straightforwardly propagated along arc (v, v') considering the arc-dependent driving time $t_{v,v'}$ (Equation 3.14). To obtain the set $S_{v'}$ from S_v (Equation 3.15), we remove from S_v the last visited vertex v' and all vertices $v'' \in S_v$ that cannot be reached from vertex v' within the remaining search time $\overline{T} - t_{v'}$, i.e., if $t_{v',v''} \geq \overline{T} - t_{v'}$.

3.2.2. Dominance criterion:

Equation 3.10 decomposes the non-monotonous cost $\alpha(\pi)$ into monotonous resources $A(\pi)$ and $\rho(\pi)$ that partly define a label. Given the monotonicity of $A(\pi)$, $\rho(\pi)$, and $t(\pi)$, we can then consider two partial policies π_1 and π_2 that end with the same vertex visit v and their associated labels L_1 and L_2 and say that for problem variants $\neg W/\neg C$ and $\neg W/C$, L_1 dominates L_2 ($L_1 \succ L_2$), if the following conditions are true

$$1 - \rho_v(\pi_1) \leq 1 - \rho_v(\pi_2) \tag{3.17}$$

$$A_v(\pi_1) \leq A_v(\pi_2) \tag{3.18}$$

$$t_v(\pi_1) \leq t_v(\pi_2) \tag{3.19}$$

$$v' \in S_v(\pi_1), \forall v' \in S_v(\pi_2) \tag{3.20}$$

$$\bar{T} - t_v(\pi_1) + p_{v'}(\gamma_{v'} - \bar{\beta}) \leq 0, \forall v' \in S_v(\pi_1) \tag{3.21}$$

Here, Conditions (3.17)–(3.19)&(3.21) ensure that the cost and duration of π_1 are smaller than the cost and duration of π_2. Conditions (3.19)&(3.20) check whether all non-visited vertices reachable by π_2 can be reached from π_1 as well. Condition (3.21) checks whether all reachable stations from v for π_1 contribute to decrease $\alpha_v(\pi_1)$. This check is necessary for settings where π_1 can be extended with k more stations than π_2 to detect corner cases in which these k additional stations may increase $\alpha(\pi_1)$ to a value larger than $\alpha(\pi_2)$.

While Algorithm 1 solves the SCPS problem optimally with the dominance criterion above, one may consider to drop some of the dominance conditions to obtain a heuristic dominance criterion that withdraws more labels at the price of losing optimality. In the remainder of this paper, we study a heuristic labeling algorithm, where we preserve Conditions (3.17)&(3.18) to obtain cost dominance but neglect Conditions (3.19),(3.20)&(3.21). In this context, the label definition simplifies to $L_v = (A_v, \rho_v, \alpha_v)$. We provide evidences to the selection of this heuristic dominance criterion in Section 5.2.

3.3. Rollout algorithm

In this section, we introduce a rollout algorithm, built as a forward dynamic programming procedure. We identify the best action at a given state as the one that yields minimal approximated cost. Here, the core idea of the cost approximation is to greedily expand the current policy from each candidate action up to a defined horizon to obtain an associated cost value via backpropagation. We apply a one-step decision rollout strategy (cf. Goodson et al. 2017) whose complexity equals a post-state decision rule as the approximation reduces to $w = 0$ decisions.

Let k be the index of the k^{th} decision epoch and let x_k be the non terminated epoch's state with $x_k = (C = (v_0, v_1, ..., v_k), 0)$ and v_k being the last station visited in epoch k. Let $x_{k+1} = (C', 0)$ be the state in the $(k+1)^{th}$ epoch that results from action $u = (v, 0)$ at epoch k, with $u \in \mathcal{U}(x_k)$. Using this additional notation, Figure 2 details the pseudo-code of our algorithm. We initialize the sequence of station visits C with the start vertex (l.1) and expand the sequence until the time horizon $\bar{T}$ is reached (l.2). From the current state x_k, we seek to determine the next best action $(v^*, 0)$ and initialize the variables that encode it (l.3). For all possible succeeding states x_{k+1} (l.4&l.5), we use a heuristic policy $\tilde{\pi}_v$ that bases on a greedy procedure to propagate state x_{k+1} up to a forced termination state x_{k+K}, with a look-ahead of K epoch extensions. For all propagated states x_l with $l \in [k+1, K-1]$, the greedy procedure chooses the action $(v_{l+1}, 0)$, with station v_{l+1} being selected based on availability probabilities of vertices reachable from v_l and the driving time to each of these vertices. We then use these anticipated states to evaluate the expected value of the policy-specific cost $V^{\tilde{\pi}_v}(x_{k+1})$ for the candidate state x_{k+1}. We define $\texttt{Greedycost}(x_{k+1})$ as the function that carries out the greedy expansion from $x_{k+1} = (C', 0)$ and returns $V^{\tilde{\pi}_v}(x_{k+1})$ (l.6). Repeating the greedy procedure and cost evaluation for all possible next actions $u = (v, 0) \in \mathcal{U}(x_k)$ allows us to find the action $u = (v, 0)$ that minimizes the cost to transition from state x_k to state x_{k+1} (l.7) that we define as $Q(x_k, v, x_{k+1})$, cf. Equation 3.23. Eventually, the selected action $u = (v^*, 0)$ yields minimal cost $Q(x_k, v^*, x_{k+1})$ (l.8-l.11).

In this setting, we calculate the cost for being in state x_{k+1}, $V^{\tilde{\pi}_v}(x_{k+1})$, based on the greedy

Figure 2: Forward programming based algorithm.

1: $v_k \leftarrow v_0$, $C \leftarrow (v_0)$, $x_k \leftarrow (C, 0)$, $t \leftarrow 0$
2: **while** $t \leq \overline{T}$ **do**
3: $\quad v^* \leftarrow 0$, $x^* \leftarrow 0$, $C^* \leftarrow 0$, $Q \leftarrow \infty$
4: $\quad$ **for** $(v, 0) \in \mathcal{U}(x_k)$ **do**
5: $\quad\quad x_{k+1} \leftarrow (C', 0)$
6: $\quad\quad V \leftarrow \texttt{greedyCost}(x_{k+1})$
7: $\quad\quad Q(x_k, v, x_{k+1}) \leftarrow t_{v_k,v} + (1 - \tilde{p}_v)V + \tilde{p}_v\gamma_v$
8: $\quad\quad$ **if** $Q(x_k, v, x_{k+1}) < Q$ **then**
9: $\quad\quad\quad Q \leftarrow Q(x_k, v, x_{k+1})$
10: $\quad\quad\quad v^* \leftarrow v$, $x^* \leftarrow x$, $C^* \leftarrow C'$
11: $\quad C \leftarrow C^*$, $x_k \leftarrow x^*$, $t \leftarrow t + t_{v_k,v^*}$, $v_k \leftarrow v^*$
12: **return** C

policy $\tilde{\pi}_v$ as

$$V^{\tilde{\pi}_v}(x_{k+1}) = \prod_{l=k+1}^{K}(1-\tilde{p}_{v_l})\beta_{v_l} + \sum_{l=k+1}^{K-1}[d(x_l, \tilde{\pi}_v(x_l))] \prod_{m=k+1}^{l}(1-\tilde{p}_{v_m}), \tag{3.22}$$

which allows to derive $Q(x_k, v, x_{k+1})$ as

$$Q(x_k, v, x_{k+1}) = t_{v_k,v} + (1-\tilde{p}_v)V^{\tilde{\pi}_v}(x_{k+1}) + \tilde{p}_v\gamma_v. \tag{3.23}$$

3.4. Time-dependent probability recovery function

In the basic setting of the SCPS problem, we assume that a station does not change its availability during the search's time horizon and restrict the amount of visits to each individual station. We now relax these assumptions and show which modifications are necessary to consider a recovery function r_v that allows the availability of an occupied station v to recover over time. In this case, we define $\tilde{p_v}(C)$ as follows

$$\tilde{p_v}(C) = \begin{cases} p_v, & \text{if } v \notin C \\ r_v(\Delta_v) & \text{otherwise,} \end{cases} \tag{3.24}$$

where $\Delta_v = \sum_{j=\ell}^{k-1} t_{v_j,v_{j+1}} + t_{v_k,v_{k+1}}$, with l denoting the position of the last visit to v in C and $v_{k+1} := v$.

We still consider a charging station v to be initially available at a probability of p_v. When it is blocked at the driver's arrival time $t_v = \sum_{j=0}^{\ell-1} t_{v_j,v_{j+1}}$, it may become available over time, i.e., the availability probability of v recovers according to $r_v(\Delta_v)$, which denotes a station's availability probability for an arbitrary point in time $t = \Delta_v + t_v$ that remains after the first visit and before the end of the time horizon. We specify $r_v(\Delta)$ for any Δ based on Schmoll and Schubert (2018) as

$$r_v(\Delta) = \frac{\mu_v}{\lambda_v + \mu_v}(1 - e^{-(\mu_v+\lambda_v)(\Delta)}). \tag{3.25}$$

Here, $\frac{1}{\lambda_v}$ and $\frac{1}{\mu_v}$ denote the average time station v remains available, respectively occupied, and remain constant over the search's time horizon. We can then express p_v as a function of λ_v and μ_v, with $p_v = \frac{\mu_v}{\lambda_v+\mu_v}$ (cf. Jossé et al. 2015) and simplify Equation 3.25 to

$$r_v(\Delta_v) = p_v(1 - e^{-(\frac{\mu_v}{p_v})(\Delta_v)}) \ . \tag{3.26}$$

We now assume that stations can be visited as many times as needed. Then, the action space for a state $x = (C, 0)$ slightly changes as follows,

$$\mathcal{U}(x) = \{(v, w) : v \in \widetilde{\mathcal{V}}(C), w \in \mathcal{W}\}.$$

In the remainder of this section, we outline the changes that are necessary to adapt the labeling algorithm to such a setting. Apart from the new availability probability definition $\tilde{p}_v(C)$, no modifications are necessary for the rollout algorithm.

Modifications for the Labeling Algorithm: The MDP definition remains unchanged, because we can determine the arrival time at a station based on the arrival time at the preceding station extended by the driving time that remains deterministic. Thus Δ_v and

$\tilde{p_v}(C)$ can be calculated from C without any further information. However, we need to modify the initial dominance criterion (3.17)–(3.21) as the optimal solution may now contain multiple visits to any charging station, independent of the problem variant.

We introduce an additional resource R_v that denotes the set of reachable but visited poles so that a label is now defined as follows: $L_v = (t_v, A_v, \rho_v, \alpha_v, S_v, R_v)$. We accordingly add the following REF to the set of REFs $\mathcal{F}_{v,v'}$

$$R_{v'} = R_v \cup \{v'\} - \bigcup_{v'' \in R_v, t_{v'} + t_{v',v''} > \overline{T}} \{v''\} \tag{3.27}$$

To obtain $R_{v'}$ (Equation 3.27), we insert the visited vertex v' in R_v and similarly subtract all vertices $v'' \in R_v$ that cannot be reached from vertex v' within the remaining search time.

We consider two partial policies π_1 and π_2 that end with the same visit at vertex v and their associated labels L_1 and L_2. We then say $L_1 \succ L_2$, if (3.17)–(3.20) hold and

$$v' \in R_v(\pi_1) \cup S_v(\pi_1), \forall v' \in R_v(\pi_2) \tag{3.28}$$

$$\frac{1 - \rho_v(\pi_1)}{1 - \rho_v(\pi_1)} \leq \prod_{v' \in R_v(\pi_1)} \frac{(1 - p_{v'}(1 - e^{-(\mu+\lambda)(\Delta_{v'}(\pi_2))}))}{(1 - p_{v'}(1 - e^{-(\mu+\lambda)(\Delta_{v'}(\pi_1))}))} \tag{3.29}$$

hold. Here, Condition 3.28 checks whether all visited vertices reachable by π_2 can be reached from π_1 as well. Equation 3.29 accounts for different probabilities of charging stations that have already been visited in both policies at different points in time by considering the biggest possible difference of probability values for v' between both paths. We note that we leave the heuristic variant of the dominance criterion unchanged in this context.

3.5. Integrating search related energy consumption

In this setting, we assume that the vehicle starts its search with an initial state of charge (SoC) b_0, which reduces over the course of the search depending on the driven distances. Then, longer driving distances result in higher energy consumption and an additional trade-off results between visits to far-distanced stations with a high availability probability and visits to near-distanced stations with medium to low availability probabilities. Here, a higher availability probability may be counterbalanced by traveling longer distances, which increases the energy that must be recharged or limits future visits to potential stations accordingly.

To account for this setting, we introduce additional notation and denote the energy consumed when traversing arc (v, v') as $k_{v,v'}$ and a vehicle's SoC after having visited all stations in $C = (v_0, ..., v_k)$ as b. In this setting, we can only transition from state $(C, b, 0)$ using arc (v_k, v_{k+1}) if $b \geq k_{v_k,v_{k+1}} + \underline{b}$, with $\underline{b}$ denoting the minimum feasible state of charge a vehicle must keep. Analogously to common monotonicity assumptions in related settings (cf. Sweda et al. 2017), we assume that

$$t_v(\pi_1) < t_v(\pi_2) \iff b_v(\pi_1) \geq b_v(\pi_2)$$

and state the necessary MDP modifications which hold as follows

$$\mathcal{S} = \{(C, b, a) : C = (v_0, ..., v_k),\ v_j \in \mathcal{V}\ \forall j,\ b \in [0, q_{max}],\ a \in \{0, 1\}\}\ , \tag{3.30}$$

$$\begin{aligned}\mathcal{U}(x) = \{(v, w) : w \in \mathcal{W}, v \in \bar{\mathcal{V}},\\ (\mathcal{W} = \mathcal{W}^0 \wedge \delta = 0) \vee (\mathcal{W} = \mathcal{W}^1 \wedge (\delta = 0 \vee (\delta = 1 \wedge w = 1)))\}\ ,\end{aligned} \tag{3.31}$$

$$\begin{aligned}V^\pi((C, b, a)) = (1 - a)w\beta_{v_k} + a\gamma_{v_k} + (1 - w)(1 - a)[t_{v_k, v_{k+1}}\\ + (1 - \tilde{p}_{v_{k+1}})V^\pi(C', b - k_{v_k, v_{k+1}}, 0) + \tilde{p}_{v_{k+1}}V^\pi(C', b - k_{v_k, k+1}, 1)]\ .\end{aligned} \tag{3.32}$$

First, we include a vehicle's SoC into the state space (Equation 3.30). Second, we modify the action space such that it depends on $\bar{\mathcal{V}}(C)$, which denotes a restricted set of charging station vertices that are reachable from state (C, a) in less than $\overline{T} - t(\pi)$ time with $t(\pi) = \sum_{k=0}^{n} t_{v_k, v_{k+1}}$ and such that $k_{v_k, v_{k+1}} + \underline{b} \leq b$ (Equation 3.31). The policy-specific cost function results straightforwardly from the modified action and state spaces (Equation 3.32). To account for this new setting in our algorithms, the following modifications are necessary. The rollout algorithm requires no modification. For the labeling algorithms, sets S_v and R_v now denote vertices that are reachable within the remaining time budget and within the remaining energy budget b_v. Accordingly, we add to the existing dominance conditions (3.17)–(3.21) that

$$b_v(\pi_1) \geq b_v(\pi_2), \tag{3.33}$$

must hold. The heuristic dominance relation remains unchanged, since the modified setting does not affect Conditions (3.17)&(3.18). Additionally, the charging duration Y_v at station v now depends on the amount of time needed to recharge from b_0 up to the maximum SoC $\underline{b}$. We introduce δY_v to denote the additional charging time due to the battery depletion δb_v during the search, where $\delta b_v = b_0 - b_v = \sum_{l=0}^{l=i} k_{v_l, v_{l+1}}$. We then account in each algorithm for the charging duration as

$$Y_v' = Y_v + \delta b_v \frac{Y_v \overline{b}}{(1 - b_0)}. \tag{3.34}$$

3.6. Computational complexity improvements

In this section, we proof additional characteristics that allow to improve the computational complexity of certain problem variants. We first introduce three action space reductions, before we focus on a sharpened dominance relation for both the exact and the heuristic labeling algorithm.

3.6.1. Action space reductions

In the following, we discuss some action space reductions, which we summarize in Table 5. We refer to the initially defined action space as *complete*. In addition, we create the following reduced action spaces.

Table 5: Modified action spaces

complete	$\mathcal{U}((C,0) = \{(v,w) : w \in \mathcal{W}, v \in \widetilde{\mathcal{V}}, (\mathcal{W} = \mathcal{W}^0 \wedge \delta = 0) \vee (\mathcal{W} = \mathcal{W}^1 \wedge (\delta = 0 \vee (\delta = 1 \wedge w = 1)))\}$
direct	$\mathcal{U}((C,0) = \{(v,w) : (v_k,v) \in \widetilde{\mathcal{A}}(C), w \in \mathcal{W}\}$
direct/restricted	$\mathcal{U}((C,0) = \{(v,w) : (v_k,v) \in \widetilde{\mathcal{A}}(C), \delta = 0, w \in \mathcal{W}\}$
T^r*-restricted*	$\mathcal{U}((C,0) = \{(v,w) : w \in \mathcal{W}, v \in \widetilde{\mathcal{V}}, (\mathcal{W} = \mathcal{W}^0 \wedge \delta = 0) \vee (\mathcal{W} = \mathcal{W}^1 \wedge (\delta = 0 \vee (\delta = 1 \wedge w = 1))), t_{v_k,v} \leq T^r\}$

direct: To further reduce the search space, we restrict the visits from the last visited station v_k to any feasible neighbor station v_{k+1}, such that there does not exist any station v' on the shortest path from v_k to v_{k+1} and denote by $\widetilde{\mathcal{A}}(C)$ the set of all feasible arcs (v_k, v_{k+1}). We however allow v_{k+1} to be visited multiple times and show with Proposition 2 that accordingly *direct* doesn't lead to a loss of optimality for problem variants $\neg W/\neg C$ and $W/\neg C$.

direct/restricted: For very large instances of the problem, we combine the *direct* action space with the visits restrictions from *complete*. The setting significantly reduces the search space but at the expense of losing optimality for problem variants $\neg W/\neg C$ and $W/\neg C$.

T^r***-restricted***: Finally, we restrict visits from station v_k to stations v_{k+1} reachable in less than T^r time units, i.e., $t_{v_k,v_{k+1}} \leq T^r$, while preserving the visit restriction of *complete*.

Proposition 2. *Action space* $\mathcal{U}$ *can be modified such that* $\mathcal{U}(C,0) = \{(v) : (v_k,v) \in \widetilde{\mathcal{A}}\}$ *without loss of optimality when* $\gamma_v = 0,\ \forall v \in \mathcal{V}$.

3.6.2. Sharpened dominance relation for the dynamic programming algorithms

In the following, we aim to sharpen the dominance relation, i.e., we aim to discard labels faster without dropping optimality. In our initial problem setting, we assume that both travel times and charging station availability probabilities are unbounded. We now account for a bound on each of these values that still reflects a real-world application. Specifically, we assume that *i*) the EV driver must travel at least a certain amount of time between two charging stations

$$0 < \underline{t} \leq t_{v,v'}, \forall (v,v') \in \mathcal{A},$$

and *ii*) that one can never be entirely sure that a charging station is available

$$p_v \leq \overline{p} < 1, \forall v \in \mathcal{V}.$$

These bounds can be computed during a preprocessing step and allow for a sharpened dominance relation for the dynamic programming based labeling algorithms without invalidating our generic model. We note that charging and waiting times at a station are bounded as well.

Let W_{min} and Y_{min} (respectively W_{max} and Y_{max}) be the minimal (respectively maximal) waiting and charging times.

We then consider two partial policies π_1 and π_2 that end with the same vertex visit and their associated labels L_1 and L_2 and we refine the initial dominance criterion (3.17)–(3.21) into stronger dominance checks. Let $\Delta A_v = A_v(\pi_1) - A_v(\pi_2)$, $\Delta\bar{\rho}_v = \bar{\rho}_v(\pi_1) - \bar{\rho}_v(\pi_2)$ and $\Delta\alpha_v = \alpha_v(\pi_1) - \alpha_v(\pi_2)$

In the initial setting, if $\Delta\bar{\rho}_v \leq 0$ and $\Delta A_v > 0$, i.e., Equation (3.17) holds but Equation (3.18) not, we cannot conclude that L_1 dominates L_2. In the new setting, we can ensure that if quantity $\Delta\alpha_v$ is small enough while quantity $\bar{\rho}_v$ is large enough, then $L_1 \succ L_2$, as the lower bound on $t_{v,v'}$ and upper bound on p_v bound the propagated values of α_v and ρ_v. We then say $L_1 \succ L_2$, if (3.17)&(3.19)–(3.21) still hold and

$$\overline{p}\Delta\alpha_v \leq (-\Delta\bar{\rho}_v)(\underline{t} + \overline{p}(\gamma_{min} - \beta_{max})) \ . \tag{3.35}$$

Table 6 summarizes all variant-specific parameters β_{max}, γ_{min}.

4. Design of experiments

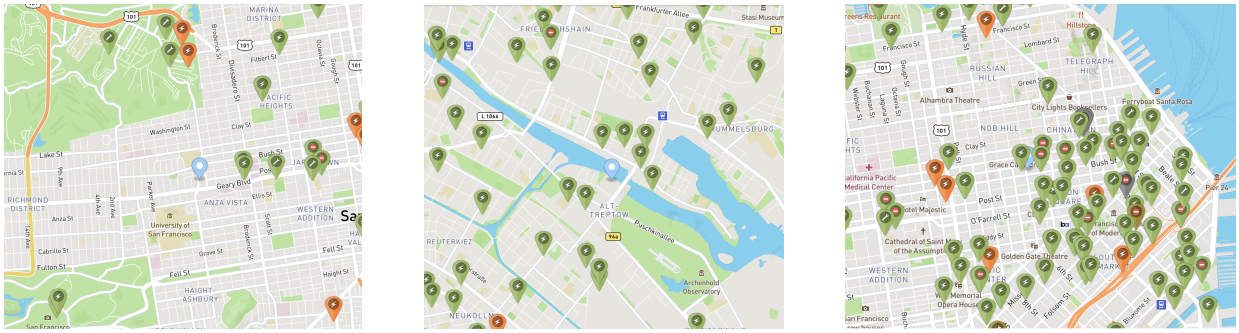
To benchmark our algorithms, we develop real-world instances that allow for extensive simulation experiments. We consider three different spatial patterns (see Figure 3), based on the west side of San Francisco, USA (*SF-1*), the city center of Berlin, Germany (*BER-1*), and the financial district of San Francisco, USA (*SF-2*). Here, we account for free-flow speeds to calculate travel times $t_{v,v'}$ that denote the time-shortest path between two stations v and v'. As our search algorithms appear to be rather insensitive to the search's starting point, we randomly choose one starting point for each pattern and use this starting point in every instance that builds on the respective pattern.

Besides the significant sensitivity to the instance size given by $\overline{S}$ and $\overline{T}$, we found during preliminary studies that our search algorithms are sensitive to two general instance characteristics: the search area's charging station density and the charging station's availability probability. Accordingly, we use the patterns described above to create a set of instances that covers a broad parameter range for these characteristics. Figure 4 shows the amount of charging stations for each pattern, depending on the search radius $\overline{S}$ around each pattern's starting point.

Our TomTom internal availability study in the city of Berlin (cf. Table 1) shows on average high charging stations' availability in areas with a large number of on-street parking spots. In areas with few available parking spots, drivers often use available stations as free parking spots,

Table 6: Parameter setting for each problem variant

Parameter	$\neg W/\neg C$	$W/\neg C$	$\neg W/C$	W/C
β_{max}	β	W_{max}	β	$W_{max} + Y_{max}$
γ_{min}	0	0	Y_{min}	Y_{min}



(a) *SF-1* (b) *BER-1* (c) *SF-2*

Figure 3: City maps

Each subplot shows the geographic area used to build the respective instance graph. *SF-1* represents the west side of San Francisco, USA, *BER-1* the city center of Berlin, Germany, and *SF-2* the financial district of San Francisco, USA.This figure bases on Open Charge Map contributors (2019) data, which is data licensed under CC BY-SA 4.01.

thus blocking access for EV drivers. In the study, the sole parking availability factor largely impacts the station availability. To reflect these amplitudes, we introduce three availability settings, drawing probabilities p_v for each charging station v from a $\beta-$distribution, which is centered on an expected availability of $[0.15, 0.60, 0.90]$ to consider a low- (*low-15%*), medium- (*avg-60%*) and high-availability (*high-90%*) scenario. The *avg-60%* and *high-90%* settings represent areas with average to high parking availability. The *low-15%* setting depicts a fictitious extreme case scenario prospectively corresponding to stricter parking policies and allows us to evaluate our algorithm behavior in such an environment.

For problem variants $W/\neg C$ and W/C, we consider a waiting duration W_v for each station v, which is uniformly distributed in $[3, 15, 60, 120]$ minutes. For problem variants $\neg W/C$ and W/C, we consider a charging duration Y_v, uniformly distributed in $[30, 60, 120]$ minutes, to account for heterogeneous stations.

With this setup, we create a total of 9 scenarios by combining each area (*BER-1*, *SF-1*, *SF-2*) with each availability probability distribution (*low-15%*, *avg-60%*, *high-90%*). We then consider a small and a large-size instance for each combination, defined by the search's time budget and search radius ($\overline{T}$[min]/$\overline{S}$[meters]). For denser areas, *BER-1* and *SF-2*, this corresponds to (5/800), resp. (10/2000), while for the sparse area *SF-1*, this corresponds to (10/2000), resp. (15/2600), resulting in a set of 18 instances per variant, 72 in total. We denote for each area, the smaller-size instance by *area*/1 (e.g., *SF-1*/1) and the larger-size instance by *area*/2.

For our studies, we set the penalty cost to $\overline{\beta} = 120$ minutes for $\neg W/\neg C$ and to $\overline{\beta} = 200$ minutes for $\neg W/C$ and refer to Section 5.2 for a discussion on its selection and sensitivity. To evaluate our algorithms, we conduct $N = 1000$ simulation runs, each with a different station availability realization, drawn from the respective probability distribution. For all average values reported in our results discussion, we applied two-tailed Wilcoxon signed-rank tests to verify its significance.

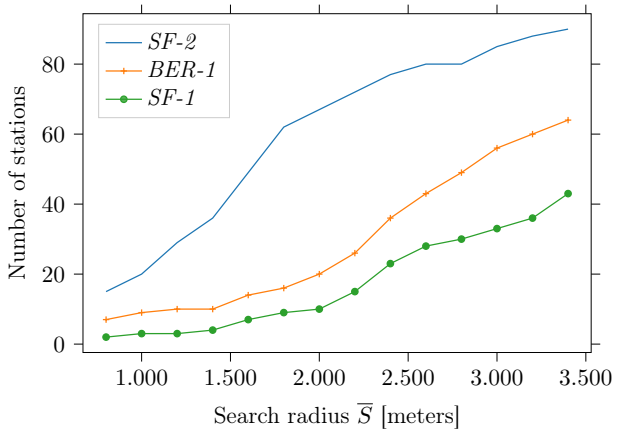


Figure 4: Number of stations depending on the search radius for each city district

5. Results

In the following, we discuss our results. We first detail the performance of our algorithms with respect to several quality metrics as well as their computational effort (Section 5.1), before we discuss results of additional sensitivity analysis and modeling assumptions (Section 5.2). Similar to the methodology section, we mainly focus the discussion on problem variant $\neg W/C$ and provide detailed results for all additional variants in Appendix E. We implemented the proposed algorithms single-threaded with Python 3.6.9, using PyPy 7.3.0 with GCC 7.3.1 and performed all experiments on a Virtual Machine on a Hypervisor, with 19 cores and 60 GB of RAM, running Ubuntu 18.04.3 LTS. For the following discussion we note that all algorithms can be implemented more efficiently, e.g., in a C++ environment. However, as this work stems from an industry project, we are not allowed to disclose the C++ related computational times for confidentiality reasons and use the Python implementation for a fair discussion of the algorithms' complexity.

5.1. Performance analysis

We discuss the algorithms' performance from two perspectives. First, we analyze the practical benefits of our algorithms based on user-centered metrics in a simulation environment. We then relate these results to a technically focused discussion on their computational complexity and solution quality.

During algorithm testing and development, we noticed that both the labeling heuristic and the rollout algorithm are sensitive to the used action space (cf. Section 3.6) in between different problem variants. For the following discussion, we report results for the respective best-performing reduced action space and refer to Appendix D for a detailed discussion of this impact.

5.1.1. Applicability

To evaluate the performance of our algorithms in practice, we use metrics that relate to the drivers' charging experience rather than to the theoretical objective value α. In the following, we refer to $\widehat{\alpha}$ as the realized search cost, which corresponds to the realized driving time needed to find a station and complete charging in case of a successful search. We first analyze for each algorithm the average search cost deviation, which we calculate as

$$\Delta\widehat{\alpha} = \sum_{i=1}^{1.000} \frac{(\widehat{\alpha}_i - \widehat{\alpha}_i^*)/\widehat{\alpha}_i^*}{1.000}$$

with $\widehat{\alpha}_i$ being the realized search cost when following the solution of the respective algorithm for simulation run i, and $\widehat{\alpha}_i^*$ being the best realized search cost for simulation run i out of all algorithms. Second, we compare the realized success rate, denoted with $\widehat{\rho}$, that results straightforwardly from the number of simulation runs, for which the respective search strategy finally led to finding an unoccupied charging station. Note that for problem variants with waiting, the search always ends successfully; in the worst case, with a long waiting time at the last visited charging station. We accordingly compare the maximum realized search cost for these variants in Appendix E.1.

As the exact labeling algorithm (LE) cannot solve large instances of our problem to optimality, we limit the comparison, in the simulation environment, to our heuristic algorithms (LH and RO) and add a naive and a greedy solution as myopic benchmarks. The greedy search (G) creates a sequence of charging station visits based on a greedy cost combining the travel time weighted by a station's availability probability and a penalty cost weighted by its occupancy probability. The naive search (N) mimics a driver without assistance by selecting the closest non-visited station with highest availability probability.

Table 7 details the average realized search cost deviations between all algorithms while Table 8 details the realized success rate of each algorithm based on 1.000 simulation runs for each scenario to avoid a statistical bias. In Table 7, an average deviation of zero implies that an algorithm always found the best search strategy, while increasing deviations indicate that the best search strategy was not found by the respective algorithm. In some low availability scenarios, LH yields higher average search costs than both RO and G, while N is always strongly outperformed with respect to $\Delta\widehat{\alpha}$.

As can be seen in Table 8, all algorithms show comparable performances in average and high availability scenarios but the LH algorithm shows much higher success rates compared to the RO algorithm and the greedy search in low availability scenarios, which compensates the larger average search cost. Remarkably, for some scenarios, the naive approach yields the highest success rates. This highlights that there exists a trade off between search costs and success rates, which we illustrate in Figure 5, by showing aggregated results over all density scenarios. Analyzing Tables 7&8 and Figure 5 jointly, we conclude that for some scenarios the naive algorithm outperforms the LH algorithm in terms of success rates at the price of significantly higher search cost. Vice versa, the RO algorithm outperforms the LH

Table 7: Average search cost deviations for $\neg W/C$

	low-15%				*avg-60%*				*high-90%*			
	N	G	RO	LH	N	G	RO	LH	N	G	RO	LH
SF-1/1	0.12	**0.02**	0.05	0.03	1.10	0.01	**0.00**	**0.00**	0.98	0.07	**0.00**	**0.00**
SF-1/2	1.41	0.18	**0.11**	0.20	2.85	0.02	0.01	**0.00**	0.06	0.05	**0.00**	**0.00**
BER-1/1	0.20	0.19	0.07	**0.04**	0.13	**0.01**	**0.01**	**0.01**	2.76	**0.00**	**0.00**	**0.00**
BER-1/2	0.42	0.35	**0.05**	0.21	0.94	0.03	**0.01**	**0.01**	2.92	0.06	**0.00**	**0.00**
SF-2/1	1.24	**0.01**	0.12	0.09	0.21	0.03	**0.00**	**0.00**	1.02	0.02	**0.00**	**0.00**
SF-2/2	0.19	0.06	0.03	**0.02**	0.99	0.03	**0.00**	**0.00**	1.03	0.03	**0.00**	0.03

The table compares the average search cost deviation $\Delta\widehat{\alpha}$ for LH, RO, G, and N for each instance of the $\neg W/C$ problem variant.

Table 8: Success rate for $\neg W/C$

	low-15%				*avg-60%*				*high-90%*			
	N	G	RO	LH	N	G	RO	LH	N	G	RO	LH
SF-1/1	0.80	0.55	0.79	0.83	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
SF-1/2	0.94	0.74	0.69	0.96	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
BER-1/1	0.78	0.73	0.78	0.78	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00
BER-1/2	0.93	0.92	0.68	0.93	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
SF-2/1	0.74	0.51	0.42	0.77	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
SF-2/2	0.89	0.89	0.91	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

The table compares the success rate $\widehat{\rho}$ for LH, RO, G, and N for each instance of the $\neg W/C$ problem variant.

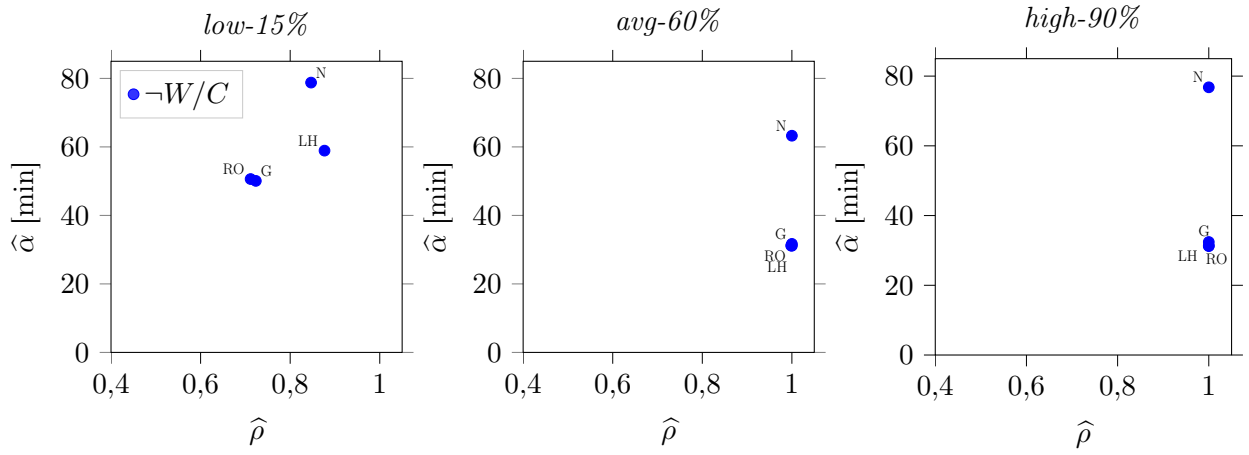


Figure 5: Averaged search cost vs. success rate for problem variant $\neg W/C$

algorithm for some scenarios in terms of average search costs at the price of significantly reduced success rates.

Additional analysis in Appendix E.1 show similar trends for the other non-waiting problem variant $\neg W/\neg C$. For the problem variants with waiting (W), the improvement between G and both heuristics LH and RO with respect to the average search cost is significantly larger than for $\neg W$ variants in low availability scenarios. In addition, results show an analogous trade-off between the average search cost and the maximum search cost.

Summarizing over all problem variants results (cf Appendix E.1), the developed search algorithms can significantly improve the search quality across all scenarios. Compared to G, the advanced algorithms decrease the search cost by 21 % in average and up to 44% for areas with a scarce number of charging stations and low charging station availability. For $\neg W$ variants, the failure rate decreases by 30 % with low charging station availability. Moreover, advanced algorithms allow to reduce search times by five ($W/\neg C$) to 31 (W/C) minutes compared to myopic approaches for W problem variants with low station availability. This reduction potential decreases for scenarios with average to high station availability. However, advanced algorithms appear to be more robust in these cases and lower the worst search costs by 20 % ($W/\neg C$) and 12 % (W/C). Comparing the LH and RO algorithm among each other, LH prolongs the search compared to RO but obtains a significantly higher success rate for $\neg W$ variants. For scenarios *avg-60%* and *high-90%* RO and LH show a similar performance.

5.1.2. Computational tractability

To compare the performance of the LH and RO algorithm against the exact labeling algorithm, we derive a set of 504 test instances by varying $\overline{T} \in [5, 10, 15, 20]$ minutes, $\overline{S} \in [800, 1000, ..., 3400]$ meters, and the availability distribution in {*low-15%*, *avg-60%*, *high-90%*} for each search area in {*SF-1*, *BER-1*, *SF-2*}. Here, we use a large time limit of 15.000 seconds to obtain a sufficient set of solutions that are eligible for our comparison, i.e., solutions for instances that could be solved with all three algorithms. We compute for each algorithm the true objective value α as well as the simulated estimate of the true objective value, denoted with $\bar{\alpha}$, based on 1.000 simulation runs for each test instance.

Table 9 shows for both heuristic algorithms (RO, LH) and for the exact labeling algorithm (LE) the rate of instances solved within this time limit ($\hat{n}$) and the average computational time of the successful runs ($\hat{t}$). For the heuristics RO and LH only, it shows the the optimality gap ($\Delta(\alpha) = \frac{\alpha^{\text{heur}} - \alpha^{\text{opt}}}{\alpha^{\text{opt}}}$) and the deviation of the objective value estimate to the LE's value ($\Delta(\bar{\alpha}) = \frac{\bar{\alpha}^{\text{heur}} - \bar{\alpha}^{\text{opt}}}{\bar{\alpha}^{\text{opt}}}$) both averaged over all instances. Figure 6 shows the distribution of optimality gaps $\Delta(\alpha)$ split per availability and density scenario.

As can be jointly observed in Table 9 and Figure 6, our LH algorithm provides optimal or close to optimal solutions. The RO algorithm shows a similar solution quality for scenarios with higher availability but shows a significantly worse solution quality for scenarios with a low charging station availability. These high deviations result from penalty costs for

Table 9: Aggregated computational results over all tested instances per scenario for $\neg W/C$

		L-H				RO				L-E	
		$\hat{\Delta}(\alpha)$	$\hat{\Delta}(\bar{\alpha})$	$\hat{t}$	$\hat{n}$	$\hat{\Delta}(\alpha)$	$\hat{\Delta}(\bar{\alpha})$	$\hat{t}$	$\hat{n}$	$\hat{t}$	$\hat{n}$
low-15%	*SF-1*	0.00	0.00	104	100	0.10	0.10	0.04	100	87.7	64
	BER-1	0.00	-0.02	412	100	0.43	0.41	0.18	100	1338	46
	SF-2	0.06	0.08	1088	70	0.48	0.49	0.86	100	5313	2
avg-60%	*SF-1*	**0.00**	0.00	9.34	100	0.05	0.04	0.05	100	7.12	64
	BER-1	0.00	0.00	0.54	100	0.03	0.03	0.15	100	1057	64
	SF-2	**0.00**	0.00	355	57	0.00	0.00	0.83	100	5560	25
high-90%	*SF-1*	0.00	0.00	75.1	93	0.01	0.01	0.05	100	8.48	64
	BER-1	**0.00**	0.00	2.82	100	0.00	0.00	0.15	100	266	59
	SF-2	**0.00**	0.00	749	34	**0.00**	0.00	1.06	100	2303	4

Abbreviations hold as follows: $\hat{\Delta}(\alpha)$ - averaged optimality gap [%], $\hat{\Delta}(\bar{\alpha})$ - averaged simulated estimate deviation [%], $\hat{t}$ - averaged computational time [s], $\hat{n}$ - rate of instances that can be computed in less than 15.000 seconds. We note that an average $\hat{\Delta}(\alpha)$ of 0.00 indicates that an algorithm (almost) always finds the optimal solution. If it always finds the optimal solution we highlight the respective $\hat{\Delta}(\alpha)$ in bold font, whereas we leave it in normal font if some solutions remain heuristic but are not reflected in the value of $\hat{\Delta}(\alpha)$ due to rounding.

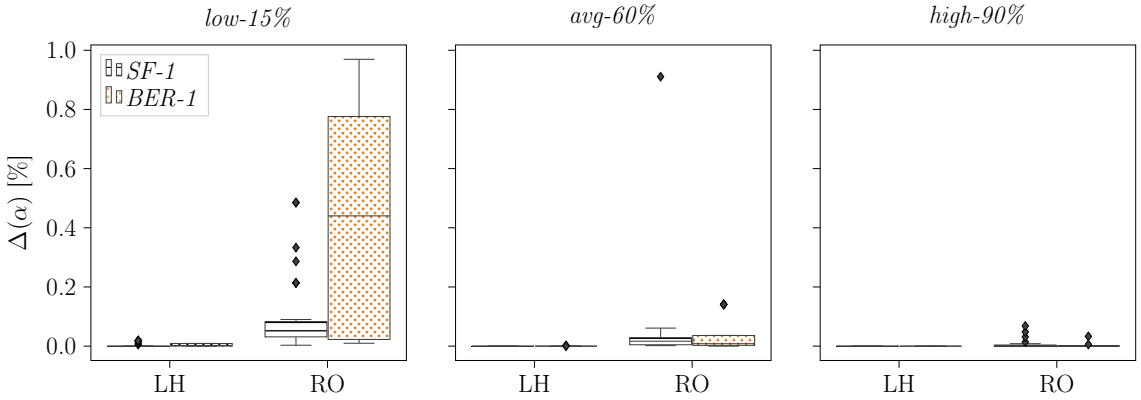


Figure 6: Optimality gap distribution, per availability and density scenarios for $\neg W/C$

Results are only shown for *SF-1* and *BER-1*, due to the very small number of instances solved to optimality for *SF-2*.

unsuccessful searches, which are more likely to occur at low charging station availability. While the RO algorithm succeeds in solving all instances with computational times of a few seconds, the LH algorithm improves upon its exact counterpart with respect to the number of instances solved and computational times but can neither solve all instances nor preserve computational times at the order of seconds. Results show very similar $\hat{\Delta}(\alpha)$ and $\hat{\Delta}(\bar{\alpha})$ values, especially in higher availability scenarios. In some low-availability scenarios (*BER-1*), LH yields close to optimal solutions but slightly outperforms LE with respect to the estimates of the true objective value ($\Delta(\bar{\alpha}) \leq 0$).

Figure 7 shows the computational time of LH, RO, and LE for a representative subset

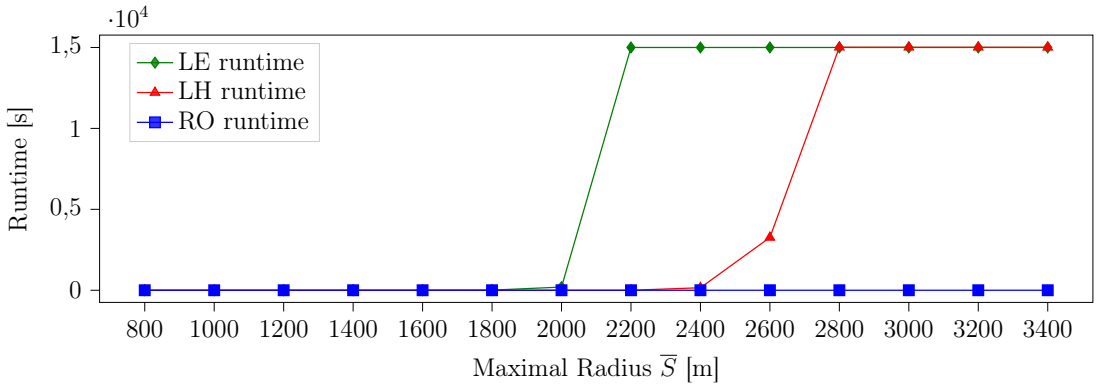


Figure 7: Computational times for LH, LE and RO for the *BER-1* and *high-90%* scenario with fixed $\overline{T}$=20 min for $\neg W/C$

of instances (*SF-1*, $\overline{T} = 20$, *high-90%*) for different search radii. As can be seen, RO and LH remain equally efficient for search radii up to 2400m but the computational time of LH increases exponentially for bigger search radii. Synthesizing Table 9 and Figure 7, we observe a trade-off between LH and RO: while RO yields robust computational times at the price of varying solution quality, LH yields a robust solution quality at the price of exponentially increasing computational time.

A comparable analysis for the remaining variants can be found in Appendix E.2. Overall the analysis shows similar computation and solution-quality trends (cf. Table 16) but in scenarios with low charging station availability, we observe for RO a lower optimality gap for the W variants than for the $\neg W$ variants.

From a practitioner's perspective, computational times of a few milliseconds are imperative to deploy a search algorithm in practice, e.g., embedded into a navigation application. Here, one could resolve the trade-off between RO and LH in two different ways. On the one hand, one could apply both algorithms selectively, using LH for tractable problem sizes and RO for larger problem sizes. On the other hand, one could always apply LH and terminate its search after a given time limit. To analyze which strategy appears to be more promising in our case, we compare the performance of RO and LH against each other, limiting the computation time of LH to one second, which equals a sufficiently small computational time when using an efficient implementation.

Figure 8 shows this comparison for problem variant $\neg W/C$, while we provide figures for all other problem variants in Appendix E. In general, the time-limited LH outperforms RO (blue areas) for small search radii, in particular for instances with low charging station availability, whereas RO outperforms the time-limited LH (red areas) in some cases for large search time budgets and bigger search radii. Accordingly, using both algorithms selectively appears to be a reasonable deployment strategy in practice.

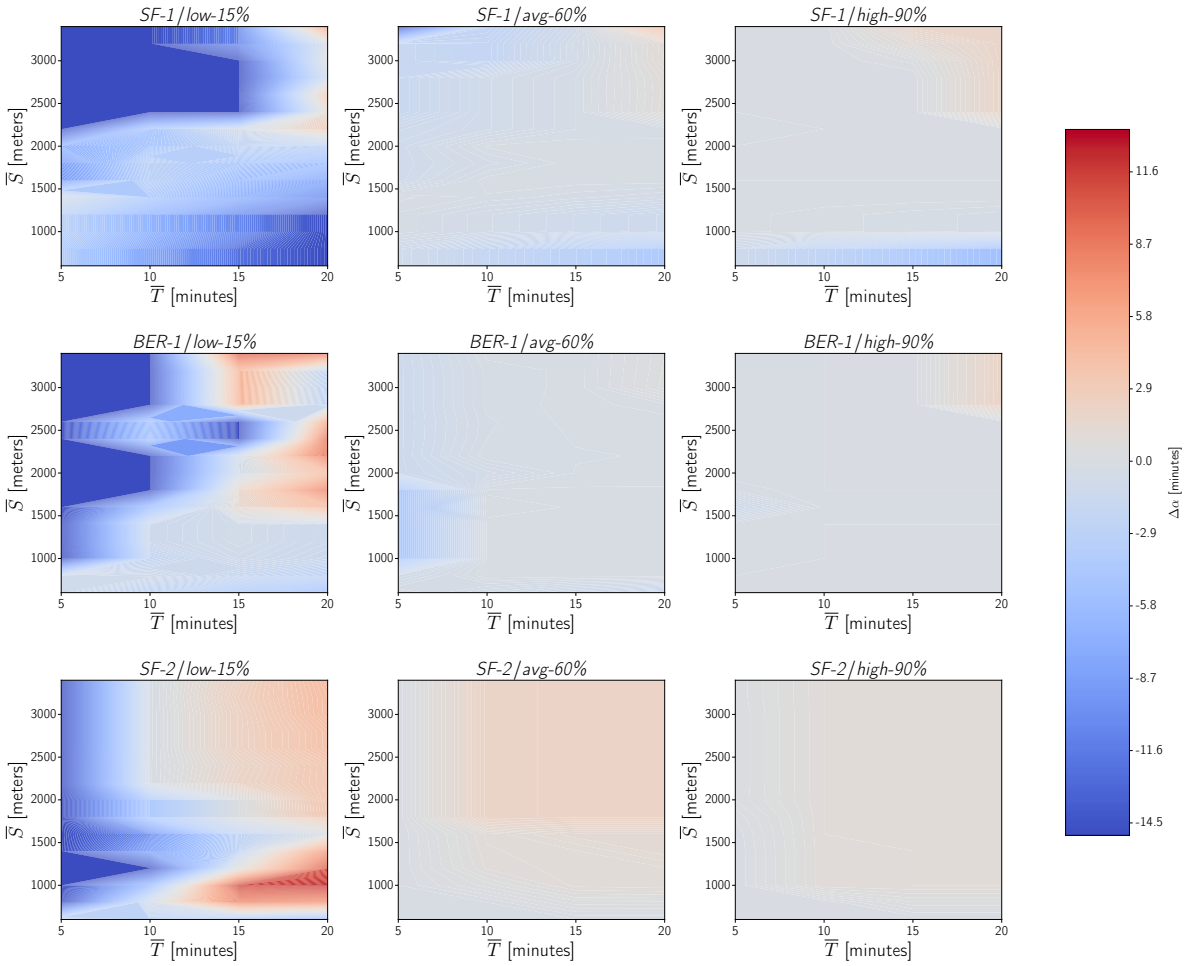


Figure 8: Extensive comparison of the LH and RO heuristics for problem variant $\neg W/C$

Each subplot shows $\Delta\alpha = \alpha^{LH} - \alpha^{RO}$ (with α^{LH}, resp. α^{RO} being the solution cost for LH, resp. RO) as a function of $\overline{T}$ and $\overline{S}$, where we limit LH computational times to 1 second. Each subplot corresponds to one of the 9 scenarios resulting from the combination of each area (*SF-1*, *BER-1*, *SF-2*) with each availability distribution (*low-15%*, *avg-60%*, *high-90%*). Over all subplots, availability increases from left to right and station density increases from top to bottom.

5.2. Extended analysis

In the following, we analyze the sensitivity of our algorithms towards parameter and design decisions. We first analyze the algorithms' sensitivity towards the penalty cost $\bar{\beta}$ (for $\neg W/C$), before we study the impact of a time dependent recovery function, substantiate the heuristic dominance decisions, and study the impact of additional charging times due to battery depletion during the search.

5.2.1. Termination penalty

We limit the discussion of termination penalty sensitivities to low charging station availability instances as these appear to be the most sensitive. Apparently, analyzing the objective value α does not allow for a meaningful interpretation of $\bar{\beta}$ sensitivities as α naturally increases with increasing $\bar{\beta}$. To circumvent this issue we decompose α into user-relevant metrics (see

Appendix F), and analyze the computational time, the expected search and charging time t^s and the probability $\bar{\rho}$ that the search unsuccessfully terminates, averaged over six *low-15%*-instances for problem variant $\neg W/C$. We discuss results for problem variant $\neg W/\neg C$ in Appendix E.

As can be seen in Figure 9, we observe a goal conflict between the expected search and charging time and the search's success rate: with increasing β we obtain better success rates at the price of higher expected search times. We note that one must choose $\beta \geq 30$ minutes, as otherwise the cost for not visiting any station is lower than the lowest cost for visiting at least one station (i.e., the lowest charging time). We observe that the β values which are necessary to obtain the best possible success rate are significantly higher (200 minutes) than this lower bound.

While the computational times of LE significantly decrease with increasing β, the computational times of LH and RO remain insensitive to changes in β for low charging station availability scenarios. However, additional analysis show that LH computational times increase with increasing β in high charging station availability scenarios, such that varying β for risk-adverse searches in a real-world implementation (i.e., using a higher β) should be chosen with respect to the computational overhead for LH.

5.2.2. Time-dependent recovery function

To analyze the impact of considering time-dependent recovering probabilities, we compare the objective values obtained in our main model (without recovering) with the model introduced in Section 3.4 (with recovering). In the former, we compute the search path with a persistent charging station occupancy but calculate the true objective value α considering time-dependent recovery for a fair comparison of both models. Based on preliminary studies, we set the average occupancy time to two hours such that $\frac{1}{\mu} = 120$ minutes. In addition, we observe that the estimated objective value $\bar{\alpha}$ is (nearly) identical to the true objective value α, in-line

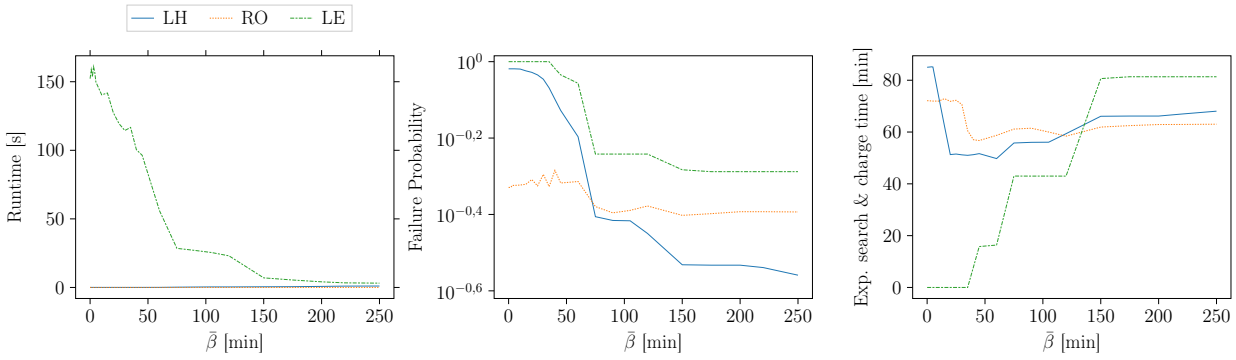


Figure 9: Impact of $\bar{\beta}$ on averaged computational time, expected search & charging time and failure rate for the *low-15%* instances for $\neg W/C$

For LE, data are averaged over 6 instances corresponding to two instances ($\bar{S} = 800/\bar{T} = 5$ and $\bar{S} = 1.000,\bar{T} = 10$) per spatial scenario (*SF-1*, *BER-1* and *SF-2*) and for LH and RO over 9 instances corresponding to three instances ($\bar{S} = 1.000/\bar{T} = 5$, $\bar{S} = 1.500,\bar{T} = 10$ and $\bar{S} = 2.000,\bar{T} = 15$) per spatial scenario.

with Table 9. Accordingly, we only report the value of α, for the sake of conciseness.

Table 10 compares the value of α for both models. As can bee seen no significant difference exists between the initial model and the updated model at the exception of smaller areas with large time budget, particularly in case of low station availability. This is the only particular case where visiting stations multiple times within the time budget might be worth an extra detour, as all stations can be visited at least once. No trade-off between unknown far-distanced stations and known short-distanced stations appears as it seems always better to visit unknown stations first. An additional analysis in Appendix E.3 for problem variant $\neg W/\neg C$ yields similar insights while there is no significant impact of the time-dependent recovery function on W problem variants.

5.2.3. Relaxed dominance criteria

To design our labeling heuristic, we relaxed the dominance check of LE as described in Section 3. In the following, we substantiate the design decision from Section 3. Here, we identify each possible variant of the dominance check with a boolean quintuple that signifies whether an equation (3.17, 3.18, 3.19, 3.20, 3.21 is active (= 1) or not (=0) in the respective dominance check, e.g., quintuple (1,0,1,0,0) identifies the dominance check variant in which only Equation 3.17 and Equation 3.19 are active.

Figure 10 shows the trade-off between the optimality gap and the computational times for all dominance criteria. As can be seen, the (heuristic) dominance criterion as chosen in

Table 10: Potential solution improvement for the time-dependent probability recovery function for problem variant $\neg W/C$

		low-15%				*avg-60%*			
		LH		RO		LH		RO	
$\overline{T}$	$\overline{S}$	α^{ref}	α^{new}	α^{ref}	α^{new}	α^{ref}	α^{new}	α^{ref}	α^{new}
5	800	88.7	87.9	89.7	89.0	32.1	32.1	32.9	32.9
5	2000	80.9	80.9	157	157	31.9	31.9	33.1	33.1
5	3400	80.9	80.9	115	115	31.9	31.9	33.1	33.1
10	800	**86.3**	**81.7**	86.6	87.2	31.9	31.9	31.9	31.9
10	2000	67.1	67.1	98.9	98.9	31.6	31.6	32.3	32.3
10	3400	47.8	48.5	68.4	68.4	31.5	31.5	31.7	31.7
15	800	**86.3**	**78.4**	**86.6**	**82.6**	31.9	31.8	31.9	31.8
15	2000	61.6	61.6	66.7	66.0	31.5	31.5	31.7	31.7
15	3400	44.5	44.5	45.9	45.9	31.5	31.5	31.6	31.6
20	800	**86.3**	**75.9**	**86.6**	**82.4**	31.9	31.8	31.8	31.8
20	2000	58.9	60.8	59.7	59.4	31.5	31.5	31.6	31.6
20	3400	41.4	56.4	48.6	48.4	31.4	31.4	31.6	31.6

The table compares for *BER-1* combined with *low-15%* and *avg-60%* the objective value obtained in the updated setting (α^{new}) and the initial setting (α^{ref}). The table excludes *high-90%* results as these do not show any deviations. Significant differences are shown in bold characters. LH's runtime is limited to 3600 seconds, which explains the low performance of the updated setup in $\overline{T} = 20$ / $\overline{S} = 3400$.

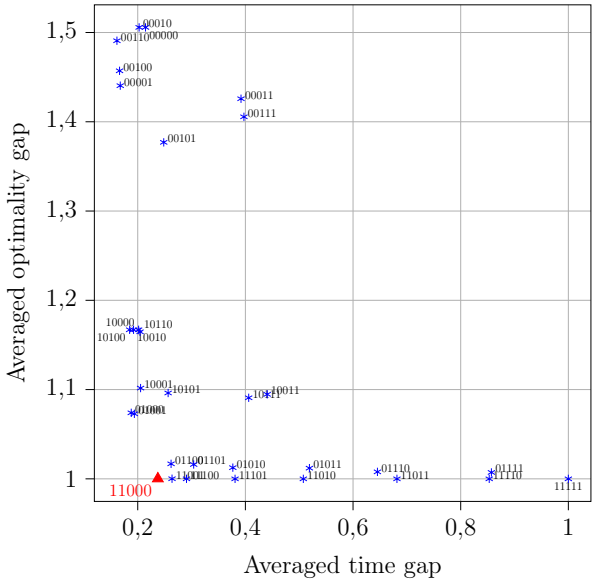


Figure 10: Comparison of heuristic dominance criteria for problem variant $\neg W/C$

The Figure shows the averaged optimality gap $g_\alpha = \sum_i {}^{\alpha_i}/_{\alpha_i^{\text{opt}}}$ as a function of the averaged computational time gap $g_t = \sum_i {}^{t_i}/_{t_i^{\text{opt}}}$ for each possible heuristic criterion for $\neg W/C$. Both values are averaged over 16 instances corresponding to *BER-1* and *SF-1* combined resp. with *low-15%*, *avg-60%* and *high-90%* for $\overline{S} \in [1200, 1400, 1600, 1800]$ and fixed $\overline{T} = 10$. The red triangle shows our selected dominance criteria.

Section 3 – (1,1,0,0,0) – yields the lowest computational times possible to achieve the best possible solution quality among all heuristic dominance criteria.

5.2.4. Battery depletion

For the following experiments, we consider a flat topography and base additional charging time calculations due to battery depletion on the technical characteristics of a Renault Zoe (battery: Z.E.50 [52 kWh], engine: R110/R135, (Renault 2020)). Furthermore, we assume that the battery level remains at minimum at 20% of its maximum capacity to account for safety considerations or driver anxiety. We approximate the extra charging time based on a linear charge curve since the additional depletion adds only a limited amount to a (partial) recharge and remains in the linear part of the overall charge curve. To calculate energy consumption, we assume a constant speed of $v = 50$ km/h, which remains a worst case estimate in an urban context. We define Y_v as the expected time to charge the battery from its initial state to a full state at station v. Then, the depleted amount of energy after t (additional) minutes of driving results to $\delta b_v(v) = \frac{t}{T_a(v)} \times 52$.

In this setting, our results show that the impact of additional charging times due to an extended search and the resulting battery depletion is very limited in urban areas. For LH, we observe an average objective value change of 0.02%, which amounts to a maximum deviation of 1.5% for single instances. The RO algorithm appears to be even less sensitive to the changed objective and shows no significant differences.

6. Conclusion and outlook

In this paper, we studied charging station search algorithms for stochastic environments, motivated by real-world applications in today's navigation system applications. We introduced the underlying problem as a finite horizon MDP that covers several real-world problem variants. In this setting, we aim to find cost minimal search paths for an individual driver. We developed three solution algorithms: an exact labeling algorithm as well as a heuristic labeling algorithm and a rollout heuristic. We benchmarked these algorithms using an extensive real-world case study with instances for the cities of San Francisco, USA and Berlin, Germany. Our results show that the heuristic algorithms allow for a significant speed-up compared to the exact algorithm at a price of a reasonable performance loss. Moreover, we show that our algorithms significantly improve a driver's success to find a free charging station compared to myopic and greedy search approaches. This is in particular the case if the number of free charging stations in the search area is scarce. In this case, our algorithms reduce a driver's search time by up to 44%.

In future work, we aim to leverage this work to study the impact of coordination and information sharing between multiple drivers. By so doing, we can study the impact of additional coordination that reduces the amount of uncertainty in the system. Moreover, studying the charging station search problem from a system perspective with a perfect information setting may yield an interesting upper bound that allows for an improved assessment of the solution quality of our algorithms in a stochastic setting.

Acknowledgments

This work was partially funded by the German Federal Ministry for Economic affairs and Energy within project iMove (01ME16003B). The authors would like to thank Marko Rosenmuëller and Andreas Linscheid for their support in the Berlin availability study.

References

Amazon. 2019. Amazon co-founds the climate pledge, setting goal to meet the paris agreement 10 years early. https://www.businesswire.com/news/home/20190919005609/en/ (last accessed: 2019-12-14). Last accessed: 11.27.2019.

Arndt, T., D. Hafner, T. Kellermeier, S. Krogmann, A. Razmjou, M. S. Krejca, R. Rothenberger, T. Friedrich. 2016. Probabilistic routing for on-street parking search. *24th Annual European Symposium on Algorithms (ESA 2016)*, vol. 57. 6:1–6:13.

Bensasson, B. 2019. Pivot power acquisition press release. https://www.edf.fr/sites/default/files/20191104pr_edf_group_acquires_pivot_power_certified.pdf_ang_0.pdf. Last accessed: 11.27.2019.

Bonges, H. A., A. C. Lusk. 2016. Addressing electric vehicle (ev) sales and range anxiety through parking layout, policy and regulation. *Transportation Research Part A: Policy and Practice* **83** 63–73.

DHL. 2019. Dhl electro mobility press release. https://www.dpdhl.com/en/media-relations/specials/electro-mobility.html. Last accessed: 11.27.2019.

Fröhlich, K. 2019. Bmw electro mobility press release. https://www.press.bmwgroup.com/global/article/detail/T0302391EN. Last accessed: 11.27.2019.

Goodson, J. C., B. W. T Thomas, J. W. Ohlmann. 2017. A rollout algorithm framework for heuristic solutions to finite-horizon stochastic dynamic programs. *European Journal of Operational Research* **258**(1) 216–229.

Guo, Q., O. Wolfson. 2018. Probabilistic spatio-temporal resource search. *GeoInformatica* **22**(1) 75–103.

Jafari, E., S. D. Boyles. 2017. Multicriteria stochastic shortest path problem for electric vehicles. *Networks and Spatial Economics* **17**(3) 1043–1070.

Jossé, G, K. A. Schmid, M. Schubert. 2015. Probabilistic resource route queries with reappearance. *Proceedings of the 18th International Conference on Extending Database Technology*. 445–456.

Kullman, N., J. C. Goodson, J. E. Mendoza. 2021. Electric vehicle routing with public charging stations. *Transportation Science* **55**(3) 637–659.

Open Charge Map contributors. 2019. Charging stations retrieved from Open Charge Map. https://api.openchargemap.io/v3/referencedata/.

Renault. 2020. Autonomie, batterie et recharge, nouvelle renault zoe. https://www.renault.fr/vehicules-electriques/zoe/batterie-recharge.html.

Schmoll, S., M. Schubert. 2018. Dynamic resource routing using real-time dynamic programming. *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. 4822–4828.

Sweda, T. M., I. S. Dolinskaya, D. Klabjan. 2017. Adaptive routing and recharging policies for electric vehicles. *Transportation Science* **51**(4) 1326–1348.

A. Proofs

Proof of Proposition 1. Let π be the policy associated to sequence C, with $C = (v_0, v_1, ..., v_n)$. We consider state $x_k = (v_k, 0)$ with charging station v_k not being available and $v_k \neq v_n$. Thus, for $i < n$, $\prod_{j=i}^{i}(1 - p_{v_j}) = 1$. For $i = n$, $\prod_{j=i}^{i}(1 - p_{v_j}) = 1 - p_{v_n}$.

We then introduce F as follows

$$\begin{aligned} F^{\pi}(x_k) = &\prod_{i=k}^{n}(1 - \tilde{p}_{v_i}(C_{[k:i-1]}))\beta_{v_n} + \sum_{i=k}^{n-1}\left[t_{v_i,v_{i+1}}\prod_{j=k}^{i}(1 - \tilde{p}_{v_j}(C_{[k:j-1]}))\right] \\ &+ \sum_{i=k}^{n}\left[\gamma_{v_i}\tilde{p}_{v_i}(C_{[k:i-1]})\prod_{j=k}^{i-1}(1 - \tilde{p}_{v_j}(C_{[k:i-1]}))\right] . \end{aligned} \tag{A.1}$$

We notice that $F^{\pi}(x_n) = (1 - p_{v_n})\beta_{v_n} + p_{v_n}\gamma_{v_n} = V^{\pi}(x_n)$.
We now show that F fulfills the recursive definition of the policy specific cost function and by recursion that $F = V$. From state $x_k = (C_{[0,k]}, 0)$, we let the cost for being in the next state $x_{k+1} = (C_{[0,k+1]}, 0)$ and seek to express $F^{\pi}(x_k)$ as a function of $F^{\pi}(x_{k+1})$ to fulfill the recursive Definition 3.5.

$$\begin{aligned} F^{\pi}(x_{k+1}) = &\prod_{j=k+1}^{n}(1 - p_{v_j})(\beta_{v_n}) + \sum_{i=k+1}^{n-1}[t_{v_i,v_{i+1}}\prod_{j=k+1}^{i}(1 - p_{v_j})] \\ &+ \sum_{i=k+1}^{n}[\gamma_{v_i}p_{v_i}\prod_{j=k+1}^{i-1}(1 - p_{v_j})] \end{aligned} \tag{A.2}$$

$$
\begin{aligned}
F^{\pi}(x_k) &= (1-p_{v_k}) \prod_{j=k+1}^{n} (1-p_{v_j})(\beta_{v_n}) + t_{v_k,v_{k+1}} + (1-p_k) \sum_{i=k+1}^{n-1} [t_{v_i,v_{i+1}} \prod_{j=k+1}^{i} (1-p_{v_j})] \\
&\quad + p_{v_k}\gamma_{v_k} + (1-p_{v_k}) \sum_{i=k}^{n} [\gamma_{v_i} p_{v_i} \prod_{j=k}^{i-1} (1-p_{v_j})] \\
F^{\pi}(x_k) &= t_{v_k,v_{k+1}} + (1-p_{v_k}) F^{\pi}(x_{k+1}) + p_{v_k}\gamma_{v_k}
\end{aligned}
\tag{A.3}
$$

Accordingly, F fulfills the recursive Definition 3.5 for $w = 0$, which concludes the proof. □

Proof of Proposition 2. We consider two simple search sequences (v) and (v, v') extended with the same visit sequence $C = (v_0, ..., v_n)$. Let $C' = (v) \circ C$, resp. $C'' = (v, v') \circ C$, associated to policies π', resp. π''. Let $t_{v,v_0} = t_{v,v'} + t_{v',v_0}$ and let v' be a direct neighbor of v (i.e., there is no station v'' such that $t_{v,v'} = t_{v,v''} + t_{v'',v'}$) and let v_0 not be a direct neighbor.

We now show that from the considered state $x_0 = ((v), 0)$, visiting v' before v_0 is always better than straightforwardly visiting v_0. We get

$$V^{\pi'}((v), 0) = t_{v,v_0} + (1-\tilde{p}_{v_0}) V^{\pi'}((v, v_0), 0),$$

$$V^{\pi''}((v), 0) = t_{v,v'} + (1-\tilde{p}_{v'}) t_{v',v_0} + (1-\tilde{p}_{v'})(1-\tilde{p}_{v_0}) V^{\pi''}((v, v', v_0), 0)$$

and distinguish two cases:

Case 1 ($v' \notin C$): In this case, the valuation of any unexplored station after v_0 does not depend on preceding visits in the respective sequence, i.e., $V^{\pi''}((v, v_0), 0) = V^{\pi''}((v, v', v_0), 0)$. Given that $(1-\tilde{p}_v) \leq 1$, we straightforwardly obtain $V^{\pi''}((v), 0) \leq V^{\pi'}((v), 0)$.

Case 2 ($v' \in C$): In this case, the valuation of any unexplored station after v_0 depends on preceding visits in the respective sequence and we obtain the following cost expansions, visiting v' at position k:

For path C', v' is visited for the first time such that $\tilde{p}_{v_k} = p_{v'}$ and we get

$$V^{\pi'}(C', 0) = \prod_{j=0}^{n} (1-\tilde{p}_{v_j})(\overline{\beta}) + \sum_{i=0}^{k-1} t_{v_i,v_{i+1}} \prod_{j=0}^{i} (1-\tilde{p}_{v_j}) + \sum_{i=k}^{n-1} [t_{v_i,v_{i+1}} (1-\tilde{p}_{v'}) \prod_{j=0, j\neq k}^{i} (1-\tilde{p}_{v_j})
\tag{A.4}$$

For path C'', v' is visited for the second time such that $\tilde{p}_{v_k} = 0$ and we get

$$(1-\tilde{p}_{v'})V^{\approx''}(C'',0) = (1-\tilde{p}_{v'})\prod_{j=0}^{n}(1-p_{v_j})(\overline{\beta}) + (1-\tilde{p}_{v'})\sum_{i=0}^{k-1}[t_{v_i,v_{i+1}}\prod_{j=0}^{i}(1-p_{v_j}) + \sum_{i=k}^{n-1}[t_{v_i,v_{i+1}}(1-\tilde{p}_{v'})\prod_{j=0,j\neq k}^{i}(1-p_{v_j}) \quad \text{(A.5)}$$

Since $(1-\tilde{p}_{v'}) \leq 1$, we have

$$(1-\tilde{p}_{v'})V^{\approx''}((v,v',v_0),0) \leq V^{\pi'}((v,v_0),0)$$

and consequently

$$V^{\pi''}(x_0) \leq V^{\pi'}(x_0).$$

In both cases, π'' is preferred over π' (thus C'' over C'), such that candidate stations can be restricted to neighbor stations only, which concludes the proof.

□

B. Problem Complexity

Proposition 3. *The SCPS problem is NP-hard, even with metric travel times.*

Proof of Proposition 3. We show hardness through reduction from the traveling salesman problem (TSP) with metric and integer travel times. The decision problem variant of the TSP can be defined as follows: we consider a set $\mathcal{V}$ of n sites and travel times $t_{v,v'} \in \mathbb{N}$ between these. Travel times are bounded $1 \leq t_{v,v'} \leq \Delta$ for all $v, v' \in \mathcal{V}$. We are asked for a tour (i.e., a Hamiltonian path) $v_0, \ldots, v_n = v_0$ whose length satisfies $\sum_{i=0}^{n-1} t_{v_i,v_{i+1}} \leq \theta$ for a given θ. Given that travel times are integer, we can assume w.l.o.g. that $\theta \in \mathbb{N}$. Further, we assume that triangle inequality holds, i.e., $t_{v,v'} + t_{v',v''} \geq t_{v,v''}$ for all $v, v', v'' \in \mathcal{V}$. We note that hardness for this restricted metric case implies hardness for the generic case as well.

Step 1: We construct an instance for the station search from the TSP instance as follows: We select an arbitrary vertex $s \in \mathcal{V}$ and designate it as start vertex $v_0 := s$ for the search. We then create a duplicate s' in the same location ($t_{s,s'} = 0$), which serves as the termination vertex. Let q be an arbitrary value satisfying

$$\left(1 - \frac{1}{\Delta(n+1)}\right)^{\frac{1}{n-1}} \leq q < 1\,. \quad \text{(B.1)}$$

We parameterize the search as follows: All vertices c have an availability of $p_i := 1 - q$ without recovery. There is no penalty for successful charging ($\gamma_i := 0\ \forall i$). For unsuccessfully

terminating at s', the penalty is

$$\beta_{s'} := \frac{2\Delta}{q(1-q)} + 1 \ . \tag{B.2}$$

For all other vertices $c \neq s'$ the penalty is

$$\beta_v := \frac{1}{q^{n+1}}(\beta_{s'} + n\Delta) + 1 \ . \tag{B.3}$$

Now for any search path $C = (v_0, \ldots, v_k)$ that does not visit any vertex multiple times, it holds that

$$\begin{aligned} \alpha(C) &= \left(\prod_{i=0}^{k}(1-p_{v_i})\right)\beta_{v_k} + \sum_{i=0}^{k-1} t_{v_i,v_{i+1}} \prod_{j=0}^{i}(1-p_{v_j}) + \sum_{i=0}^{k} \gamma_{v_i} p_{v_i} \prod_{j=0}^{i-1}(1-p_{v_j}) \\ &= q^{k+1}\beta_{v_k} + \sum_{i=0}^{k-1} q^{i+1} t_{v_i,v_{i+1}} \ . \end{aligned} \tag{B.4}$$

Step 2: We now claim that the TSP instance possesses a solution with cost at most $\theta \in \mathbb{N}$ if, and only if, the station search admits a search path C with $\alpha(C) \leq q\theta + q^{n+1}\beta_{s'}$. This is done by transforming solutions between the two problems and carefully mapping their objective values.

For the first direction, we assume that a TSP tour is given. We convert it to a search path $C = (v_0, \ldots, v_n)$ by cutting at s such that $v_0 = s$ and $v_n = s'$. Then

$$\alpha(C) = q^{n+1}\beta_{s'} + \sum_{i=0}^{n-1} q^{i+1} t_{v_i,v_{i+1}} \leq q^{n+1}\beta_{s'} + q\sum_{i=0}^{n-1} t_{v_i,v_{i+1}} \leq q^{n+1}\beta_{s'} + q\theta \ .$$

Vice versa, we assume that we are given a search path P with $\alpha(P) \leq q\theta + q^{n+1}\beta_{s'}$. Then for an optimal search path $C = (v_0, \ldots, v_k)$ it holds that $\alpha(P) \leq \alpha(C) \leq q\theta + q^{n+1}\beta_{s'}$. Given metric travel times and no recovery, we can assume that C does not visit any vertex more than once. We construct a tour through the following observations:

1. C visits s': Assume it does not. Let C' be C extended by ending at s'. Then

$$\begin{aligned} \alpha(C') &= \alpha(C) - q^{k+1}\beta_{v_k} + q^{k+2}\beta_{s'} + q^{k+1}t_{v_k,s'} \\ &= \alpha(C) - q^{k+1}(\beta_{v_k} - q\beta_{s'} - t_{v_k,s'}) \overset{(*)}{<} \alpha(C) \end{aligned} \tag{B.5}$$

   using in (*) that $\beta_{v_k} > q\beta_{s'} + \Delta$ by (B.3). This contradicts the optimality of C.

2. C visits s' last: Assume it does not. We obtain C' from C by moving s' to the end.

Then it holds that

$$\begin{aligned} &\alpha(C) \geq q^{k+1}\beta_{v_k} \geq q^{n+1}\beta_{v_k} \text{ and} \\ &\alpha(C') \leq q^{k+1}\beta_{s'} + \sum_{i=0}^{k-1} q^{i+1}\Delta \leq \beta_{s'} + n\Delta \ . \end{aligned} \tag{B.6}$$

Given that $q^{n+1}\beta_{v_k} > \beta_{s'} + n\Delta$ (B.3), it holds that $\alpha(C) > \alpha(C')$, contradicting optimality.

3. C visits every vertex: Assume $C = (v_0, \ldots, v_{k-1}, s')$ omits some vertex v'. Let $C' = (v_0, \ldots, v_{k-1}, v', s')$. Then

$$\begin{aligned} \alpha(C) - \alpha(C') &= \left(q^k t_{v_{k-1},s} + q^{k+1}\beta_{s'}\right) - \left(q^k t_{v_{k-1},v'} + q^{k+1} t_{v',s'} + q^{k+2}\beta_{s'}\right) \\ &= q^k (t_{v_{k-1},s} - t_{v_{k-1},v'} - q t_{v',s'} + (q - q^2)\beta_{s'}) \\ &\geq q^k(-2\Delta + q(1-q)\beta_{s'}) \overset{\text{(B.2)}}{>} 0 \ , \end{aligned} \tag{B.7}$$

again contradicting optimality.

4. It is clear now that C corresponds to a TSP tour, by identifying s with s'. It holds that

$$\alpha(C) = \sum_{i=0}^{n-1} q^{i+1} t_{v_i,v_{i+1}} + q^{n+1}\beta_{s'} \leq q\theta + q^{n+1}\beta_{s'} \ . \tag{B.8}$$

Assume the tour would violate the threshold, i.e., $\sum_{i=0}^{n-1} t_{v_i,v_{i+1}} > \theta$. Given integrality, the length then is at least $\theta + 1$. It follows that

$$\begin{aligned} \sum_{i=0}^{n-1} q^{i+1} t_{v_i,v_{i+1}} &\geq \sum_{i=0}^{n-1} q^n t_{v_i,v_{i+1}} = q \sum_{i=0}^{n-1} q^{n-1} t_{v_i,v_{i+1}} \\ &\overset{\text{(B.1)}}{\geq} q \sum_{i=0}^{n-1} (1 - \frac{1}{\Delta(n+1)}) t_{v_i,v_{i+1}} = q \left[\sum_{i=0}^{n-1} t_{v_i,v_{i+1}} - \sum_{i=0}^{n-1} \frac{t_{v_i,v_{i+1}}}{\Delta(n+1)} \right] \\ &\geq q \left[\theta + 1 - \sum_{i=0}^{n-1} \frac{1}{n+1} \right] > q\theta \ . \end{aligned} \tag{B.9}$$

This contradicts (B.8), thereby proving that $\sum_{i=0}^{n-1} t_{v_i,v_{i+1}} \leq \theta$.

□

C. Variant-specific methods

We now present the required methodological changes for the problem variants $\neg W/\neg C$, $W/\neg C$ and W/C. Besides describing the cost structure variants (Section C.1), we detail the

necessary modifications for the labeling and the rollout algorithms (Section C.2).

C.1. Cost structure variants

For $\neg W/\neg C$, $W/\neg C$ and W/C problem variant as introduced in Section 2.1.1, the cost functions $V^\pi(C,0)$, $V^\pi(C,1)$ and $V^\pi(x_n)$ for a termination state $x_n = (C,a)$, can be expressed as follows, with C' denoting the extension of C by station v_{k+1} for non-termination states.

No waiting, without charging cost ($\neg W/\neg C$)

$$\begin{aligned} V^\pi(C,0) &= t_{v_k,v_{k+1}} + (1-\tilde{p}_{v_{k+1}})V^\pi(C',0)\ , \\ V^\pi(C,1) &= 0\ , \\ V^\pi(x_n) &= (1-\tilde{p}_{v_n})\overline{\beta}\ . \end{aligned} \tag{C.1}$$

Waiting permitted, without charging cost ($W/\neg C$)

$$\begin{aligned} V^\pi(C,0) &= wW_{v_k} + (1-w)[t_{v_k,v_{k+1}} + (1-\tilde{p}_{v_{k+1}})V^\pi(C',0)]\ , \\ V^\pi(C,1) &= 0\ , \\ V^\pi(x_n) &= (1-\tilde{p}_{v_n})W_{v_n}\ . \end{aligned} \tag{C.2}$$

Waiting permitted, with heterogeneous charging costs (W/C)

$$\begin{aligned} V^\pi(C,0) &= w(Y_{v_k} + W_{v_k}) + (1-w)[t_{v_k,v_{k+1}} + (1-\tilde{p}_{v_{k+1}})V^\pi(C',0) + \tilde{p}_{v_{k+1}}L_{v_{k+1}}]\ , \\ V^\pi(C,1) &= Y_{v_k}\ , \\ V^\pi(x_n) &= (1-\tilde{p}_{v_n})W_{v_n} + Y_{v_n}\ . \end{aligned} \tag{C.3}$$

C.2. Algorithm variants

The main methodology section discusses the algorithm implementation for problem variant $\neg W/C$, which can be applied to the problem variant $\neg W/\neg C$ without any further changes, because both variants ignore the waiting decision. In the following, we specify the required changes to account for waiting times and waiting decisions that can be applied to both problem variants $W/\neg C$ and W/C.

C.2.1. Labeling algorithm

Similar to the changes required to account for the time-dependent recovery functions detailed in Section 3.4, we now introduce the resource R_v in the label definition (cf. Equation 3.27), and define a label as follows: $L_v = (t_v, A_v, \rho_v, \alpha_v, S_v, R_v)$.

For problem variants $W/\neg C$ and W/C, we recall that a search can terminate before the whole time budget is spent and that a station terminating the search might have been visited

earlier. Accordingly we drop Condition 3.21 and use Condition 3.28 (defined in Section 3.4) in the dominance criterion that checks whether all visited vertices reachable by π_2 can be reached from π_1 as well. With L_1 and L_2 being the associated labels to π_1 and π_2, we then say that $L_1 \succ L_2$ if Conditions (3.17)–(3.20) are true and Condition (3.28) holds.

Here, each newly created label can also serve as a termination label, as the driver can decide to wait to terminate the search. Figure 11 shows the adapted pseudo-code (l.10), where we relax the termination condition stating that no feasible successors should be left.

C.2.2. Rollout algorithm

For problem variants $W/\neg C$ and W/C, we use an additional procedure `refinePolicy(C)` that bases on the resulting visits sequence C to introduce the waiting decision at the best stage $1 \leq k \leq n$, with n being the total amount of station visits in C. Figure 12 shows the adapted pseudo-code, where we account for the waiting decision.

We first calculate the no-waiting case and compute policy π with the associated ordered sequence of charging stations $C = (v_0, ..., v_n)$. We denote π as an intermediate policy and introduce π_S representing the final search policy. Then, `refinePolicy(C)` calculates π_S using the intermediate policy π while permitting wait-actions (l.12). For each intermediary charging station v_k at the k^{th} decision epoch, $v_k \in (v_0, ..., v_n), k \neq n$, π provides a sub sequence of charging stations to visit until the end of the search $(v_{k+1}, ..., v_n)$ and thus the policy-specific

Figure 11: Dynamic programming based labeling algorithm.

1: $\mathcal{L}^a \leftarrow \{L_0\}$, $L^* \leftarrow L_0$
2: **while** $\mathcal{L}^a \neq \emptyset$ **do**
3: $\quad L \leftarrow$ `costMinimumLabel(`$\mathcal{L}^a$`)`
4: $\quad \mathcal{L}^a \leftarrow \mathcal{L}^a \setminus \{L\}$
5: $\quad$ **for** $(v, v') \in \delta^+(L)$ **do**
6: $\quad\quad L' \leftarrow \mathcal{F}_{ij}(L)$
7: $\quad\quad$ **if** `isNotDominated(`$L', \mathcal{L}^a$`)` **then**
8: $\quad\quad\quad$ `dominanceCheck(`$\mathcal{L}^a, L'$`)`
9: $\quad\quad\quad \mathcal{L}^a \leftarrow \mathcal{L}^a \cup \{L'\}$
10: $\quad\quad\quad$ **if** $\alpha(L') < \alpha(L^*)$ **then**
11: $\quad\quad\quad\quad L^* \leftarrow L'$
12: **return** L^*

Figure 12: Forward programming based algorithm.

1: $v_k \leftarrow v_0$, $C \leftarrow (v_0)$, $x_k \leftarrow (C, 0)$, $t \leftarrow 0$
2: **while** $t \leq \overline{T}$ **do**
3: $\quad v^* \leftarrow 0$, $x^* \leftarrow 0$, $C^* \leftarrow 0$, $Q \leftarrow \infty$
4: $\quad$ **for** $(v, 0) \in \mathcal{U}(x_k)$ **do**
5: $\quad\quad x_{k+1} \leftarrow (C', 0)$
6: $\quad\quad V \leftarrow$ `greedyCost`(x_{k+1})
7: $\quad\quad Q(x_k, v, x_{k+1}) \leftarrow t_{v_k,v} + (1 - \tilde{p}_v)V + \tilde{p}_v\gamma_v$
8: $\quad\quad$ **if** $Q(x_k, v, x_{k+1}) < Q$ **then**
9: $\quad\quad\quad Q \leftarrow Q(x_k, v, x_{k+1})$
10: $\quad\quad\quad v^* \leftarrow v$, $x^* \leftarrow x$, $C^* \leftarrow C'$
11: $\quad C \leftarrow C^*$, $x_k \leftarrow x^*$, $t \leftarrow t + t_{v_k,v^*}$, $v_k \leftarrow v^*$
12: $C \leftarrow$ `refinePolicy(`C`)`
13: **return** C

cost value $V^{\pi}(x_k)$, associated to policy π and state $x_k = ((v_0, ..., v_k), 0)$. We aim to quantify for each station v_k whether the termination cost β_{v_k} is actually lower than the expected cost for continuing the search $V^{\pi}(x_k)$. If this is the case, the optimal decision is to wait and we refine π into π_S with $\pi_S(x_k) = (v_k, w = 1)$ and $V^{\pi_S}(x_l) \leq V^{\pi}(x_l)\ \forall l \in [0, n]$.

We define π_S as

$$\pi_S(x_k) = \underset{\pi(x_k),(v_k,w=1)}{\arg\min} w\beta_{v_k} + (1-w)[t_{v_k,v_{k+1}} + (1-\tilde{p}_{v_{k+1}})\min(V^{\pi}(x_{k+1}), V^{\pi_S}(x_{k+1})) + \tilde{p}_{v_{k+1}}\gamma_{v_{k+1}}], \tag{C.4}$$

where $\pi(x_k) = (v_{k+1}, w = 0)$. If there exists an index k such that $0 \leq k < n$, $x_k = ((v_0, ..., v_k), 0)$ and $\pi_S(x_k) = (v_k, 1)$, then state x_k terminates the search, as the driver will wait at v_k if v_k is not immediately available. In this case π_S encodes the solution $(v_0, ..., v_k)$.

C.2.3. Integrating search related energy consumption

Charge times become path-sensitive for problem variant W/C, similar to problem variant $\neg W/C$, whereas $\neg W/\neg C$ and $W/\neg C$ do not account for path-dependent charging times based on their definitions.

D. Reduced action spaces results

Table 11 compares the computational times, and the percentage share of instances that can be computed in less than 15.000 seconds by each heuristic. Further, for the instances that can be solved to optimality within 15.000 seconds, the table compares for both heuristics the averaged optimality gap and computational time gap.

Preliminary results show that using the *complete* action space for problem variants $\neg W/\neg C$ and $W/\neg C$ with LH is computationally too heavy to be of any practical use, such that we restrict results to the other restricted action spaces. As can be seen, while *direct* only slightly helps saving computational times on average for $\neg W/\neg C$ and LH, it allows to solve 6% more instances within the allocated time for $W/\neg C$. For RO, results show a 96% decrease of the computational time for $W/\neg C$. For $\neg W/C$, restricting the next station visits from the current location to the ones accessible in less than five minutes allows to save 36% of the computational times compared to *complete*. Accordingly, Table 12 shows the most appropriate action space for each heuristic and problem variant.

E. Problem variant-specific results

In the following, we discuss the results for problem variants $\neg W/\neg C$, $W/\neg C$ and W/C of the applicability study (Section E.1), the computational tractability study (Section E.2) and the extended analysis (Section E.3).

Table 11: Aggregated computational results over all tested instances for each problem variant and used action space

Graph setup		LH				RO			
Problem variant	Action space	$\hat{\Delta}(t)$	$\hat{\Delta}(\alpha)$	$\hat{t}$	$\hat{n}$	$\hat{\Delta}(t)$	$\hat{\Delta}(\alpha)$	$\hat{t}$	$\hat{n}$
$\neg W/\neg C$	*direct*	-0.79	0.01	197	0.73	-0.65	0.27	0.08	1.00
	direct/restricted	-0.84	0.04	235	0.82	–	–	–	–
	T^r*-restricted*	-0.83	0.00	221	0.74	–	–	–	–
	complete	–	–	–	–	-0.77	0.13	0.58	1.00
$W/\neg C$	*direct*	-0.80	0.00	72.7	0.83	-0.80	0.03	0.03	1.00
	direct/restricted	-0.84	0.02	16.7	1.00	–	–	–	–
	T^r*-restricted*	-0.84	0.01	250	0.78	–	–	–	–
	complete	–	–	–	–	-0.64	0.04	0.76	1.00
$\neg W/C$	T^r*-restricted*	-0.83	0.01	311	0.84	-0.82	0.12	0.31	1.00
	complete	-0.84	0.01	486	0.84	-0.81	0.12	0.38	1.00
W/C	T^r*-restricted*	-0.74	0.00	120	0.83	-0.69	0.00	0.88	1.00
	complete	-0.80	0.00	100	0.83	-0.66	0.00	1.05	1.00

Abbreviations hold as follows: $\hat{\Delta}(t)$ - averaged computational time gap with $\Delta(t) = \frac{t^{\text{heur}} - t^{\text{opt}}}{t^{\text{opt}}}$ [%], $\hat{\Delta}(\alpha)$ - averaged optimality gap over all tested instances with $\Delta(\alpha) = \frac{\alpha^{\text{heur}} - \alpha^{\text{opt}}}{\alpha^{\text{opt}}}$ [%], $\hat{t}$ - averaged computational time [s], $\hat{n}$ - rate of instances that can be computed in less than 15.000 seconds.

Table 12: Best action space /heuristic combination per problem variant

Problem variant	LH	RO
$\neg W/\neg C$	*direct*	*complete*
$W/\neg C$	*direct*	*direct*
$\neg W/C$	T^r*-restricted*	*complete*
W/C	*complete*	*complete*

For all problem variants, the table shows for each heuristic (LH, RO) the graph setting to that provides the best trade-off between computational times and solution quality.

E.1. Applicability

We use the same metrics to evaluate the performances of our algorithms for problem variant $\neg W/\neg C$. Since the search always ends successfully in waiting variants $W/\neg C$ and W/C, we analyze the algorithms based on their maximum realized search cost. Similarly to the simulated search cost evaluation, we use the maximum realized search cost deviation, that results to $\Delta\widehat{\alpha}^{\max} = {}^{\widehat{\alpha}^{\max}}/_{\widehat{\alpha}^{\max^*}}$ with $\widehat{\alpha}^{\max} = \max \widehat{\alpha}_i$ being the maximum cost out of all simulation runs for a single algorithm and with $\widehat{\alpha}^{\max^*}$ being the maximum cost over all simulation runs and all algorithms. Note that the realized search cost corresponds to the realized driving time needed to find a station for variants $\neg W/\neg C$ and $W/\neg C$, and to complete charging for W/C. For W variants, $\hat{\alpha}$ includes the waiting time whenever the last visited station is occupied.

Table 13 details the average realized search cost deviations between all algorithms for problem variants $\neg W/\neg C$, $W/\neg C$ and W/C while Table 14 details the realized success rate

Table 13: Average search cost deviations for $\neg W/\neg C$, $W/\neg C$ and W/C

		low-15%				*avg-60%*				*high-90%*			
		N	G	RO	LH	N	G	RO	LH	N	G	RO	LH
$\neg W/\neg C$	*SF-1*/1	1.38	0.78	**0.00**	0.77	2.63	0.41	**0.00**	**0.00**	1.33	0.35	0.19	**0.00**
	SF-1/2	1.13	0.59	**0.09**	0.14	2.66	2.55	**0.00**	**0.00**	2.52	1.08	**0.00**	**0.00**
	BER-1/1	0.08	**0.02**	**0.02**	0.08	0.39	0.39	**0.25**	0.27	0.28	0.98	**0.00**	**0.00**
	BER-1/2	0.18	0.08	**0.01**	**0.01**	3.38	0.16	**0.03**	**0.03**	4.14	0.24	**0.00**	**0.00**
	SF-2/1	4.74	4.72	0.57	**0.28**	5.98	5.98	**0.01**	**0.01**	6.51	3.18	**0.01**	**0.01**
	SF-2/2	8.31	6.52	0.25	**0.16**	10.7	6.16	0.17	**0.10**	6.43	6.43	**0.01**	**0.01**
$W/\neg C$	*SF-1*/1	5.67	6.61	6.68	**0.60**	2.71	1.72	**0.01**	**0.01**	1.08	0.12	**0.00**	**0.00**
	SF-1/2	2.30	6.19	0.67	**0.41**	2.71	1.69	**0.01**	**0.01**	2.53	0.11	**0.00**	**0.00**
	BER-1/1	0.41	1.32	**0.20**	**0.20**	0.15	0.36	**0.04**	**0.04**	0.25	**0.00**	**0.00**	**0.00**
	BER-1/2	4.39	1.35	**0.20**	**0.20**	4.36	0.62	**0.24**	**0.24**	4.12	**0.00**	**0.00**	**0.00**
	SF-2/1	58.1	4.55	**0.42**	**0.42**	6.41	0.55	**0.01**	**0.01**	6.52	0.04	**0.01**	**0.01**
	SF-2/2	33.1	4.72	**0.19**	**0.19**	10.8	0.73	0.17	**0.09**	6.40	0.05	**0.01**	0.05
W/C	*SF-1*/1	2.37	1.17	**0.00**	**0.00**	1.14	0.04	**0.00**	**0.00**	0.98	**0.00**	**0.00**	**0.00**
	SF-1/2	2.11	4.24	**0.01**	**0.01**	2.83	0.04	**0.00**	**0.00**	0.06	**0.00**	**0.00**	**0.00**
	BER-1/1	0.74	0.01	**0.00**	**0.00**	0.12	0.02	**0.00**	**0.00**	2.79	0.02	**0.00**	**0.00**
	BER-1/2	1.37	**0.01**	**0.01**	**0.01**	0.94	0.02	**0.00**	**0.00**	2.92	**0.00**	**0.00**	**0.00**
	SF-2/1	3.48	0.03	**0.01**	**0.01**	0.17	0.01	**0.00**	**0.00**	1.02	**0.00**	**0.00**	**0.00**
	SF-2/2	0.89	0.04	**0.02**	**0.02**	0.98	0.01	**0.00**	**0.00**	1.02	**0.00**	**0.00**	**0.00**

The table compares the average search cost deviation $\Delta\widehat{\alpha}$ for LH, RO, G, and N for each instance of problem variants $\neg W/\neg C$, $W/\neg C$, and W/C.

of each algorithm for $\neg W/\neg C$. Figure 13 shows the trade-off between the average realized search costs and (i) the realized success rate, aggregated over all scenarios for problem variant $\neg W/\neg C$ (Figure 13a) or (ii) the maximum realized search cost, aggregated over all scenarios for W problem variants (Figure 13b).

As can be seen in Table 13, larger search cost deviations occur for the W variants, especially for $W/\neg C$, due to the heterogeneous penalty costs that result from waiting at an occupied station. Particularly, on the *SF-1* instances at low-availability, RO performs significantly worse compared to LH with respect to the average search time. In this case, a single varying station visit between two search strategies can cause such differences due to the limited amount of candidate stations and the large amplitude between penalty costs. For $\neg W/\neg C$, the LH algorithm shows higher success rates compared to the RO algorithm (cf. Table 14 and Figure 13a), which highlights the superiority of the LH algorithm compared to the RO algorithm as both algorithms show comparable performances with respect to the average search cost.

Table 15 shows the deviation between the maximum search cost of all algorithms for W problem variants, which are guaranteed to be feasible at the price of high waiting cost. Here, we observe a trade-off between the average search cost and the maximum search cost (cf. Figure 13b). As one can see, the LH and the RO algorithm show a similar performance for scenarios with medium and high charging station availability, but the LH algorithm performs

Table 14: Success rate for $\neg W/\neg C$

		low-15%				*avg-60%*				*high-90%*			
		N	G	RO	LH	N	G	RO	LH	N	G	RO	LH
$\neg W/\neg C$	*SF-1*/1	0.81	0.85	0.36	0.86	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	SF-1/2	0.95	0.85	0.81	0.85	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	BER-1/1	0.78	0.77	0.77	0.80	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	BER-1/2	0.95	0.80	0.78	0.80	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	SF-2/1	0.70	0.80	0.78	0.86	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	SF-2/2	0.90	0.86	0.87	0.88	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

The table compares the success rate $\widehat{\rho}$ for LH, RO, G, and N for each instance of the $\neg W/\neg C$ problem variant.

Table 15: Maximum realized search cost deviation for $W/\neg C$ and W/C

		low-15%				*avg-60%*				*high-90%*			
		N	G	RO	LH	N	G	RO	LH	N	G	RO	LH
$W/\neg C$	*SF-1*/1	22.6	2.29	10.8	**0.00**	1.42	**0.00**	0.10	0.24	2.11	**0.00**	0.13	0.13
	SF-1/2	4.45	2.29	**0.00**	**0.00**	2.01	0.11	**0.00**	**0.00**	0.69	**0.00**	0.13	0.13
	BER-1/1	1.12	**0.00**	0.32	0.32	0.28	2.08	**0.00**	**0.00**	0.27	0.34	**0.00**	**0.00**
	BER-1/2	33.7	**0.00**	0.32	0.32	0.60	2.08	**0.00**	**0.00**	0.71	0.34	**0.00**	**0.00**
	SF-2/1	29.0	0.27	**0.00**	**0.00**	0.89	1.05	**0.00**	**0.00**	0.74	**0.00**	0.70	0.70
	SF-2/2	30.2	0.27	**0.00**	**0.00**	3.60	2.04	**0.00**	1.07	0.02	0.55	**0.00**	0.55
W/C	*SF-1*/1	6.07	1.22	**0.00**	**0.00**	2.60	**0.00**	0.01	0.01	0.85	**0.00**	**0.00**	**0.00**
	SF-1/2	2.78	6.24	**0.00**	**0.00**	2.61	0.01	**0.00**	**0.00**	**0.00**	0.03	0.03	0.03
	BER-1/1	2.69	**0.00**	**0.00**	**0.00**	0.73	0.36	**0.00**	**0.00**	1.63	1.70	**0.00**	**0.00**
	BER-1/2	6.41	**0.00**	**0.00**	**0.00**	0.96	1.38	**0.00**	**0.00**	2.39	0.01	**0.00**	**0.00**
	SF-2/1	6.34	0.06	**0.00**	**0.00**	2.57	**0.00**	**0.00**	**0.00**	0.99	**0.00**	0.01	0.01
	SF-2/2	6.49	0.20	**0.00**	**0.00**	0.85	**0.00**	0.02	0.01	0.95	**0.00**	0.01	**0.00**

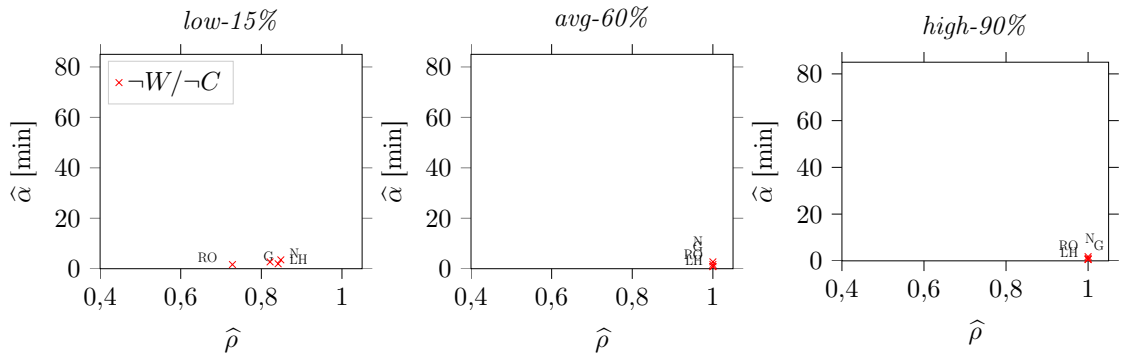
The table compares the average maximal search cost deviation $\Delta\widehat{\alpha}^{\max}$ for LH, RO, G, and N for each instance of problem variants $W/\neg C$ and W/C.

better in the SF-1/1 scenario with low charging station availability. Across all scenarios the greedy algorithm sometimes yields the least maximum cost. However, this improvement stems from significantly increased search costs (cf. Table 13).

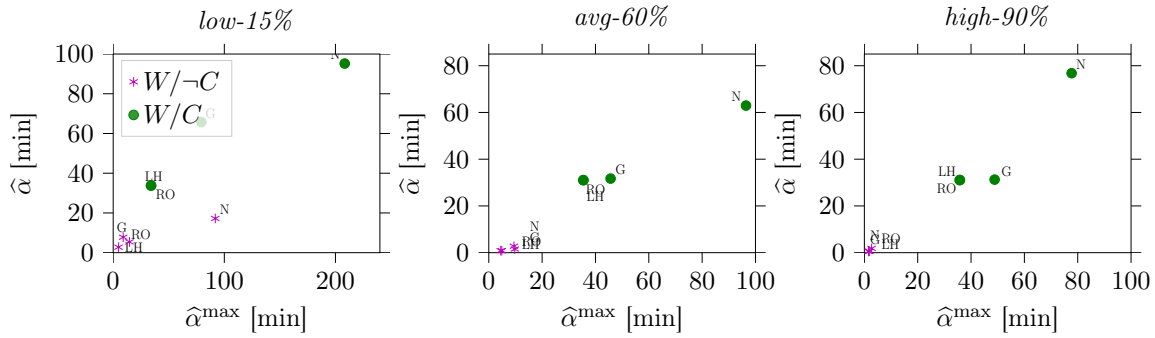
For W problem variants, we observe that the advanced algorithms outperform the myopic algorithms significantly, independent of the charging station availability. This performance difference is higher for problem variant $W/\neg C$. For problem variant W/C, the greedy algorithm performs close to the advanced algorithms for scenarios with medium or high charging station availability.

E.2. Computational tractability

Table 16 compares the performance of the LH and RO algorithm against the exact labeling algorithm for the remaining $\neg W/\neg C$, $W/\neg C$ and W/C problem variants. As can be seen, the observed differences are sensitive to the problem variant. Significant differences occur



(a) Averaged search cost vs. success rate for problem variant $\neg W/\neg C$



(b) Averaged search cost vs. worst search cost for problem variants $W/\neg C$ and W/C

Figure 13: Trade off between the average search cost and the success rate, resp. the worst search cost

for the $\neg W/\neg C$ problem variant, similar to $\neg W/C$, with low charging station availability and varying search radii, while the algorithms perform similarly on the waiting problem variants. These high deviations result from penalty costs for unsuccessful searches, which can only result for $\neg W$ problem variants and are more likely to occur at low charging station availability.

Figures 14-16 show the extensive heuristic comparisons for the remaining problem variants $W/\neg C$, $\neg W/C$, W/C. Similarly to $\neg W/C$, LH significantly outperforms RO for both non charging variants in low availability scenarios, especially in low density areas, whereas there exists no significant difference for problem variant W/C.

E.3. Extended analysis

In the following, we discuss the algorithm's sensitivity toward $\bar{\beta}$ for problem variant $\neg W/\neg C$, the heuristic dominance criterion for problem variants $\neg W/\neg C$, $W/\neg C$ and W/C and the impact of time-dependent recovery functions.

Termination penalty

For problem variant $\neg W/\neg C$, t^s corresponds to the expected search time, due to the usage cost homogeneity ($\gamma_v = 0,\ \forall v \in \mathcal{V}$). As can be seen in Figure 17, a similar goal conflict as for problem variant $\neg W/C$, between expected search time and success rate, exists. Note that

Table 16: Aggregated computational results over all tested instances per scenario for $\neg W/\neg C$, $W/\neg C$ and W/C

			L-H				RO				L-E	
			$\hat{\Delta}(\alpha)$	$\hat{\Delta}(\bar{\alpha})$	$\hat{t}$	$\hat{n}$	$\hat{\Delta}(\alpha)$	$\hat{\Delta}(\bar{\alpha})$	$\hat{t}$	$\hat{n}$	$\hat{t}$	$\hat{n}$
		SF-1	0.01	0.01	1.78	100	0.32	0.38	0.09	100	164	64
	low-15%	*BER-1*	0.04	0.04	0.49	100	0.43	0.46	0.28	100	1386	46
		SF-2	0.03	0.01	444	66	0.22	0.18	1.38	100	8646	4
		SF-1	**0.00**	0.00	266	96	0.04	0.04	0.11	100	7.02	64
$\neg W/\neg C$	*avg-60%*	*BER-1*	0.00	-0.02	84.5	100	0.01	-0.02	0.23	100	473	59
		SF-2	**0.00**	0.00	803	34	0.11	0.10	1.76	100	4816	5
		SF-1	**0.00**	0.00	107	66	0.03	0.03	0.10	100	16.4	64
	high-90%	*BER-1*	**0.00**	0.00	66.7	66	**0.00**	0.00	0.35	100	517	57
		SF-2	**0.00**	0.00	1.70	25	**0.00**	0.00	0.95	100	4156	4
		SF-1	0.04	0.05	0.05	100	0.09	0.10	0.01	100	482	68
	low-15%	*BER-1*	**0.00**	0.00	1.60	100	**0.00**	0.00	0.04	100	283	64
		SF-2	**0.00**	0.00	160	64	0.03	0.04	0.10	100	2111	25
		SF-1	0.00	0.00	0.05	100	0.03	0.03	0.01	100	2.33	64
$W/\neg C$	*avg-60%*	*BER-1*	**0.00**	0.01	0.72	100	**0.00**	0.00	0.01	100	1046	64
		SF-2	**0.00**	0.00	491	54	0.09	0.09	0.03	100	4219	25
		SF-1	**0.00**	0.00	0.06	100	0.00	0.00	0.01	100	1.68	64
	high-90%	*BER-1*	**0.00**	0.00	0.38	100	**0.00**	0.00	0.01	100	475	64
		SF-2	**0.00**	0.00	0.43	25	**0.00**	0.00	0.05	100	2024	25
		SF-1	**0.00**	0.00	1.90	100	**0.00**	0.00	0.12	100	493	68
	low-15%	*BER-1*	**0.00**	0.00	5.98	100	**0.00**	0.00	0.77	100	803	66
		SF-2	**0.00**	0.00	358	79	**0.00**	0.00	2.95	100	2565	25
		SF-1	**0.00**	0.00	0.25	100	**0.00**	0.00	0.12	100	472	68
W/C	*avg-60%*	*BER-1*	**0.00**	0.00	0.83	100	0.00	0.00	0.57	100	495	64
		SF-2	**0.00**	0.00	527	64	**0.00**	0.00	2.59	100	2031	25
		SF-1	**0.00**	0.00	1.99	89	**0.00**	0.00	0.08	100	352	68
	high-90%	*BER-1*	**0.00**	0.00	0.05	89	**0.00**	0.00	0.34	100	629	66
		SF-2	**0.00**	0.00	4.27	29	**0.00**	0.00	1.88	100	2661	25

Abbreviations hold as follows: $\hat{\Delta}(\alpha)$ - averaged optimality gap [%], $\hat{\Delta}(\bar{\alpha})$ - averaged simulated estimate deviation [%], $\hat{t}$ - averaged computational time [s], $\hat{n}$ - rate of instances that can be computed in less than 15.000 seconds. We note that an average $\hat{\Delta}(\alpha)$ of 0.00 indicates that an algorithm (almost) always finds the optimal solution. If it always finds the optimal solution we highlight the respective $\hat{\Delta}(\alpha)$ in bold font, whereas we leave it in normal font if some solutions remain heuristic but are not reflected in the value of $\hat{\Delta}(\alpha)$ due to rounding.

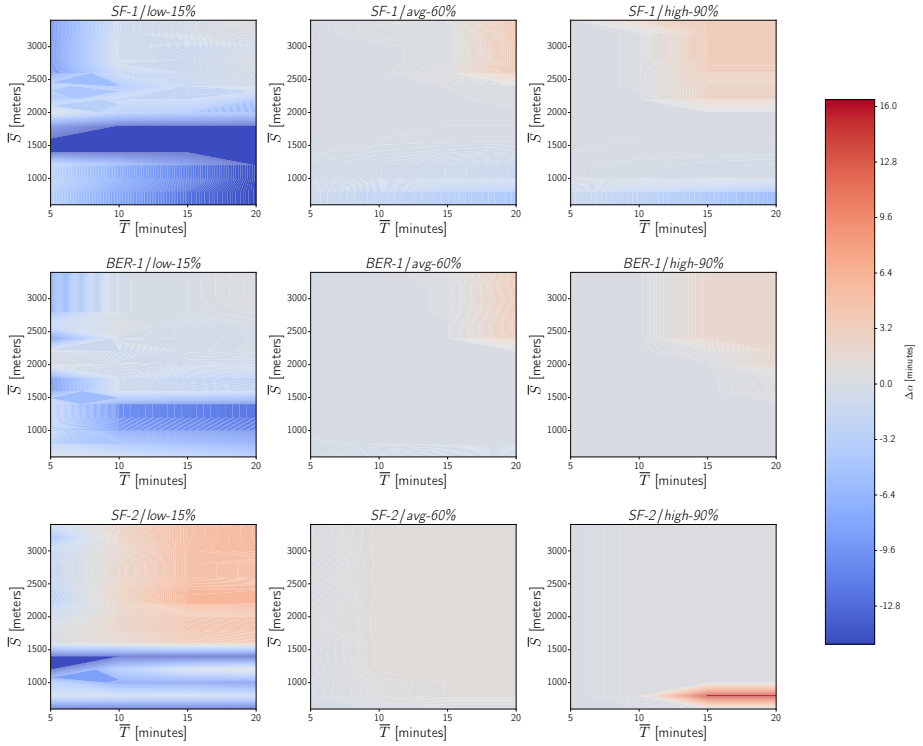


Figure 14: Extensive comparison of the LH and RO heuristics for problem variant $\neg W/\neg C$

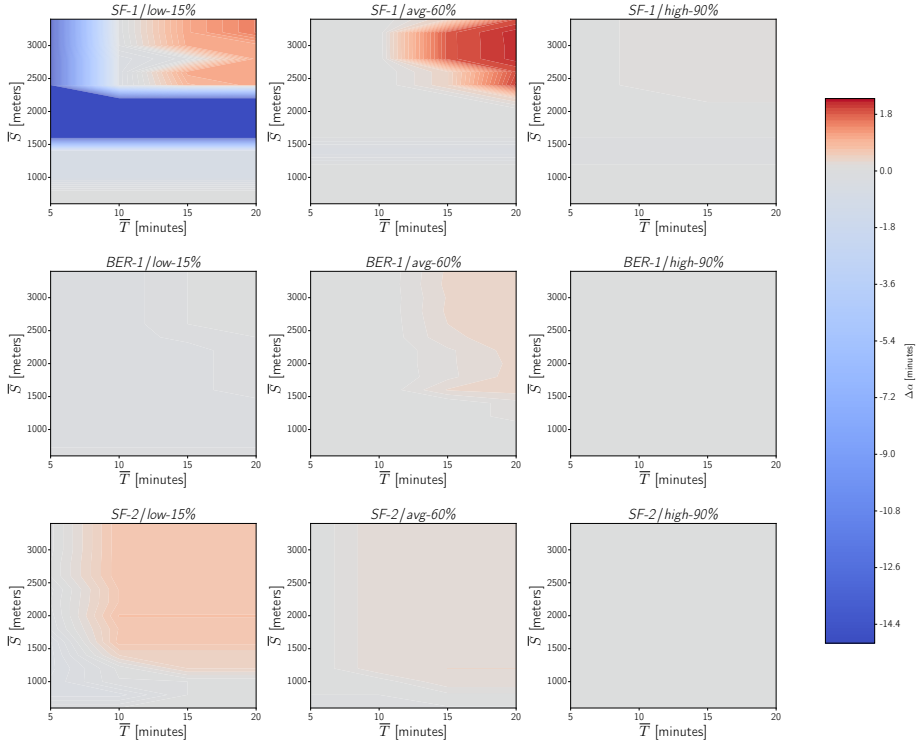


Figure 15: Extensive comparison of the LH and RO heuristics for problem variant $W/\neg C$

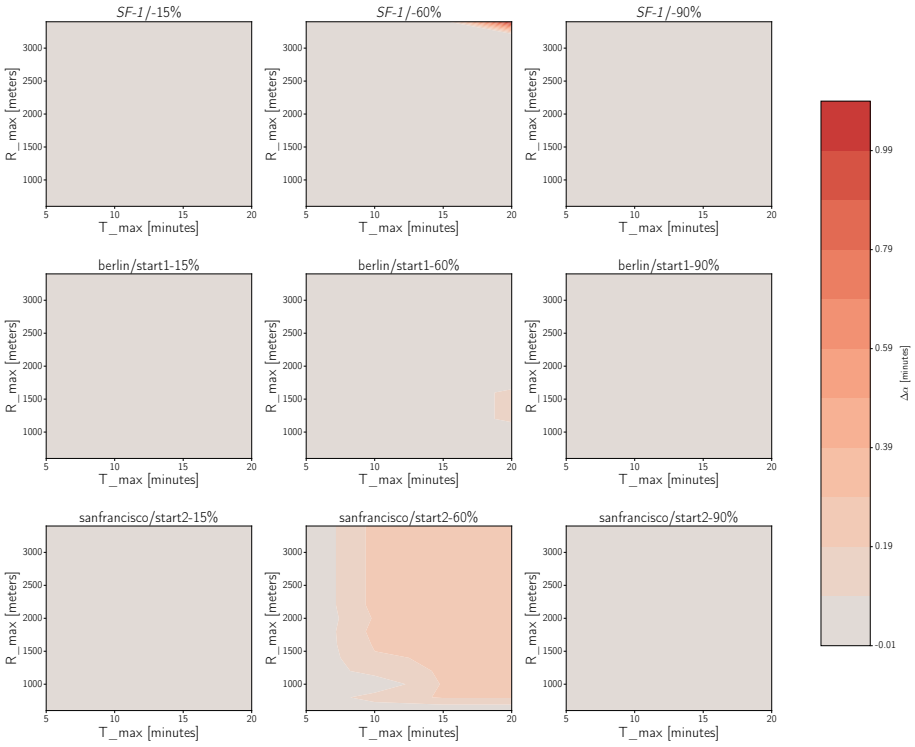


Figure 16: Extensive comparison of the LH and RO heuristics for problem variant W/C

a lower $\bar{\beta}$ value is needed to obtain best success rate solutions (120 minutes).

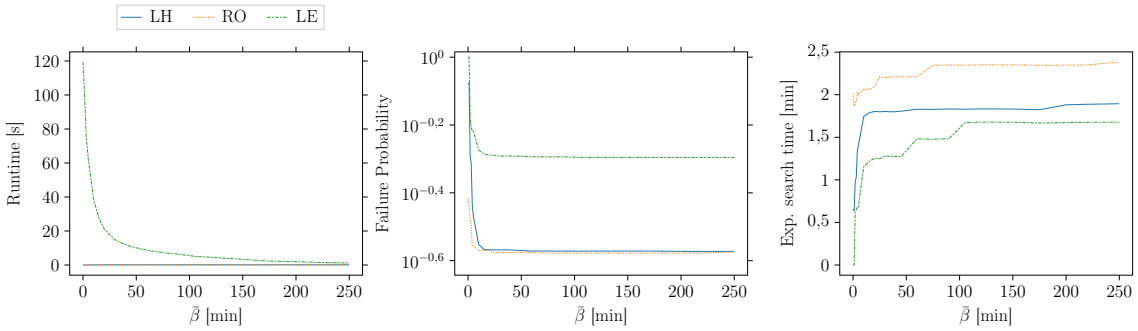


Figure 17: Impact of $\bar{\beta}$ on averaged computational time, expected search time and failure rate for the *low-15%* instances for $\neg W/\neg C$

For LE, data are averaged over 6 instances corresponding to two instances ($\bar{S} = 800/\bar{T} = 5$ and $\bar{S} = 1.000$,$\bar{T} = 10$) per spatial scenario (*SF-1*, *BER-1* and *SF-2*) and for LH and RO over 9 instances corresponding to three instances ($\bar{S} = 1.000/\bar{T} = 5$, $\bar{S} = 1.500$,$\bar{T} = 10$ and $\bar{S} = 2.000$,$\bar{T} = 15$) per spatial scenario.

Time-dependent recovery function

Results show no significant impact of the time-dependent recovery functions for the problem variants $W/\neg C$ and W/C, such that we only report values for problem variant $\neg W/\neg C$ in Table 17. As can be seen, results yield similar insights as for problem variant $\neg W/C$: time-dependent recovery functions mainly impact results in scenarios with smaller search areas, large time budget, and low charging station availability.

Table 17: Potential solution improvement for the time-dependent probability recovery function for problem variant $\neg W/\neg C$

		low-15%				*avg-60%*			
		LH		RO		LH		RO	
$\overline{T}$	$\overline{S}$	α^{ref}	α^{new}	α^{ref}	α^{new}	α^{ref}	α^{new}	α^{ref}	α^{new}
5	800	25.8	25.7	28.6	33.1	1.41	1.53	1.55	1.81
5	2000	13.8	13.85	13.9	13.8	1.32	1.32	1.34	1.34
5	3400	9.63	9.62	16.3	16.27	1.32	1.32	1.34	1.34
10	800	**23.9**	**21.1**	28.6	25.9	1.35	1.33	1.4	1.6
10	2000	6.84	6.84	7.32	11.8	1.23	1.23	1.23	1.23
10	3400	3.26	3.26	3.44	3.44	1.23	1.23	1.23	1.23
15	800	**23.9**	**18.1**	**28.6**	**24.4**	1.35	1.31	1.4	1.59
15	2000	4.25	4.28	5.05	6.14	1.23	1.23	1.23	1.23
15	3400	2.45	2.58	2.53	2.53	1.23	3.34	1.23	1.23
20	800	**23.9**	**16.3**	**28.6**	**23.4**	1.35	1.3	1.4	1.58
20	2000	3.51	3.51	4.46	4.15	1.23	1.33	1.23	1.23
20	3400	2.41	3.13	2.50	2.50	2.02	3.37	1.23	1.23

The table compares for *BER-1* combined with *low-15%* and *avg-60%* the objective value obtained in the updated setting (α^{new}) and the initial setting (α^{ref}). The table excludes *high-90%* results as these do not show any deviations. Significant differences are shown in bold characters.

Relaxed dominance criteria

Figure 18 shows the trade-off between the optimality gap and the computational times for all dominance criteria for problem variants $\neg W/\neg C$, $W/\neg C$ and W/C. Similarly to $\neg W/C$, the (heuristic) dominance criterion as chosen in Section 3 – (1,1,0,0,0) – yields the lowest computational times possible to achieve the best possible solution quality among all heuristic dominance criteria for problem variants $\neg W/\neg C$ and $W/\neg C$. For problem variant W/C, (1,1,0,0,1) yields the best trade-off by slightly decreasing computational times obtained with (1,1,0,0,0), but selecting (1,1,0,0,0) allows the best possible generic implementation for LH.

F. Objective value decomposition

In this section, we show by proving Proposition 4 that the objective value $\alpha(\pi)$ as derived in Section 2 can be expressed based on $t(\pi)$, $\rho(\pi)$ and an additional quantity $t^s(\pi)$ representing the expected time to find and use a station assuming at least one station is available among C, the visits sequence associated with π. We derive the quantity $t^s(\pi)$ based on the work of Arndt et al. (2016) that describes $t^s(\pi)$ as a sum over all stations of the accumulated driving time to a station and its utilization cost weighted by the probability the driver charges exactly at this station.

Proposition 4. *Cost α can be decomposed to exhibit $t^s(\pi)$ as follows,*

$$\alpha(\pi) = t^s(\pi) \cdot \rho(\pi) + \bar{\rho}(\pi) \cdot (t(\pi) + \beta_{v_n}) \tag{F.1}$$

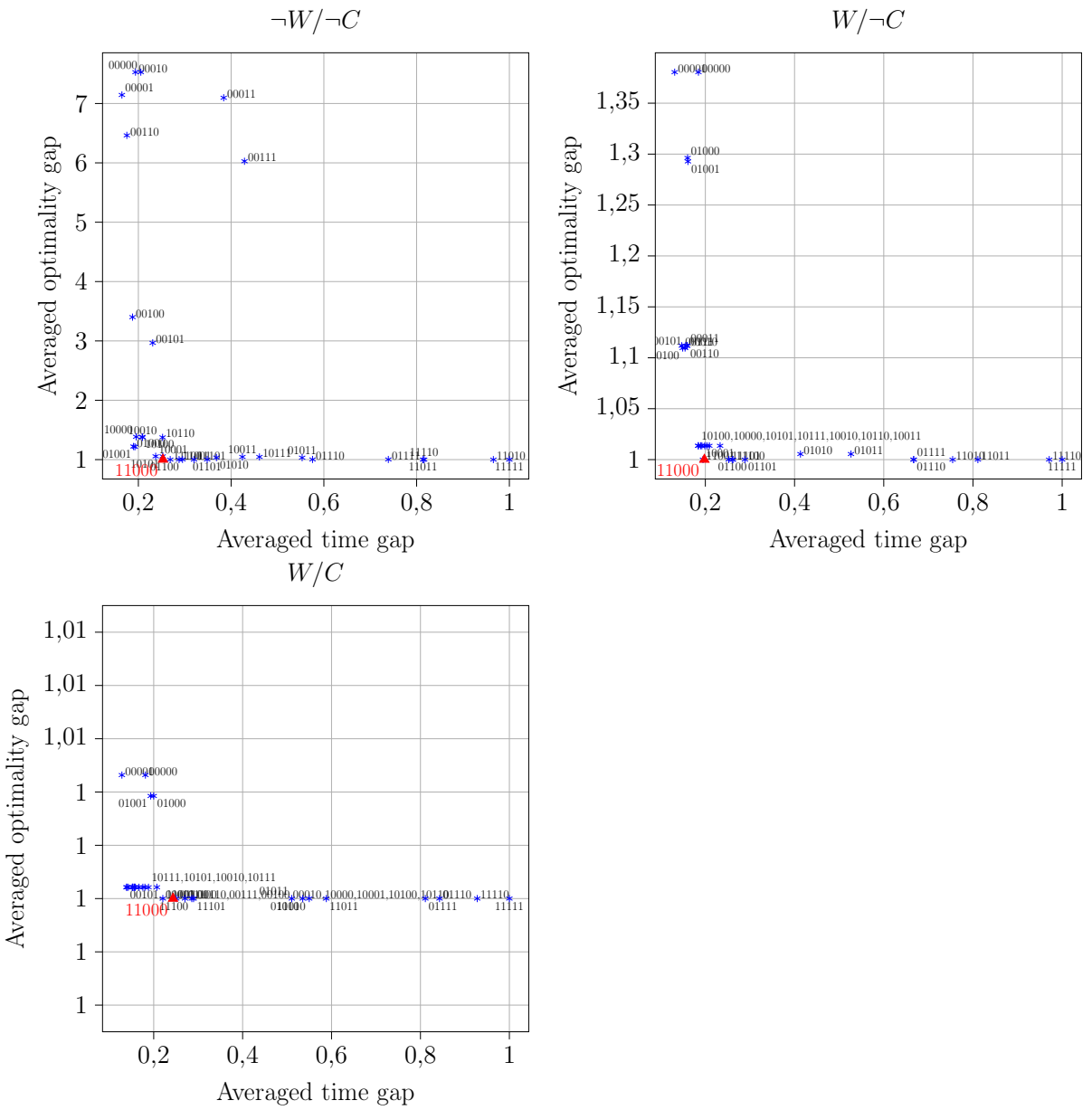


Figure 18: Comparison of heuristic dominance criteria for problem variants $\neg W/\neg C$, $W/\neg C$ and W/C

The Figure shows the averaged optimality gap $g_\alpha = \sum_i {}^{\alpha_i}/_{\alpha_i^{\text{opt}}}$ as a function of the averaged computational time gap $g_t = \sum_i {}^{t_i}/_{t_i^{\text{opt}}}$ for each possible heuristic criterion for $\neg W/C$. For each variant, both values are averaged over 16 instances corresponding to *BER-1* and *SF-1* combined resp. with *low-15%*, *avg-60%* and *high-90%* for $\overline{S} \in [1200, 1400, 1600, 1800]$ and fixed $\overline{T} = 10$. Red triangles show our selected dominance criteria.

·

Proof of Proposition 4. We recall that $\alpha(\pi) = A(\pi) + \bar{\rho}(\pi) \cdot \beta_{v_n}$ (cf. Equation 3.9). For the sake of conciseness, we simplify the notation for the remainder of this proof as follows: $C = (0, 1, ..., n)$ such that $t_{k,k+1} = t_{v_k,v_{k+1}}$, $\bar{\rho}_k = \prod_{i=0}^{k} \bar{p}_k$ We let $\rho_n = \rho(\pi)$, $A_n = A(\pi)$, $t_n = t(\pi)$, $t_n^s = t^s(\pi)$.

We then define t^s based on Arndt et al. (2016) as

$$t_n^s = \frac{\sum_{k=0}^{n-1} \bar{\rho}_{k-1} p_k (t_k + \gamma_k)}{\rho_n}.$$

We now note that $\bar{\rho}_k p_k$ represents the probability of station k being available when visited, given that no station in $i \in (0, ..., k-1)$ was available.

We then introduce the quantity $B_n = \sum_{k=0}^{n-1} \bar{\rho}_{k-1} p_k (t_k + \gamma_k)$ such that $t_n^s \cdot \rho_n = B_n$ and note that to prove F.1, it is sufficient to show

$$A_n = B_n + \bar{\rho}_n \cdot t_n, \tag{F.2}$$

which follows by recursion:
Step 1: For $n = 0$, F.1 holds : $A_0 = t_{0,1}$ and $B_0 = p_1 t_{0,1} + \bar{p}_1 t_{0,1} = A_0$.
Step 2: We assume that F.1 holds and show that $A_{n+1} = B_{n+1} + \bar{\rho}_{n+1} \cdot t_{n+1}$ holds, too:

$$A_{n+1} = A_n + \bar{\rho}_n \cdot (t_{n,n+1} + p_{n+1}\gamma_{n+1})$$

.

$$B_{n+1} = B_n + \bar{\rho}_n p_{n+1} \cdot (t_{n+1} + \gamma_{n+1}) \tag{F.3}$$

Given F.1,

$$\begin{aligned} & A_{n+1} = B_n + \bar{\rho}_n \cdot t_n + \bar{\rho}_n(t_{n,n+1} + p_{n+1}\gamma_{n+1}) \\ \Leftrightarrow\ & A_{n+1} = B_n + \bar{\rho}_n \cdot t_n + \bar{\rho}_n(t_{n,n+1} + p_{n+1}\gamma_{n+1}) + \bar{\rho}_n p_{n+1} \cdot (t_{n+1} + \gamma_{n+1}) - \bar{\rho}_n p_{n+1} \cdot (t_{n+1} + \gamma_{n+1}) \end{aligned} \tag{F.4}$$

From F.3 and F.4, we get :

$$\begin{aligned} & A_{n+1} = B_{n+1} + \bar{\rho}_n \cdot (t_n) + \bar{\rho}_n(t_{n,n+1} + p_{n+1}\gamma_{n+1}) - \bar{\rho}_n p_{n+1} \cdot (t_{n+1} + \gamma_{n+1}) \\ \Leftrightarrow\ & A_{n+1} = B_{n+1} + \bar{\rho}_n \cdot t_{n+1} - \bar{\rho}_n(p_{n+1}) t_{n+1} \\ \Leftrightarrow\ & A_{n+1} = B_{n+1} + \bar{\rho}_{n+1} \cdot t_{n+1} \end{aligned} \tag{F.5}$$

This concludes the proof. □

4 Coordinated Charging Station Search in Stochastic Environments: A Multi-Agent Approach

This chapter is based on an working article published as:

Guillet M., Schiffer M. (2022). Coordinated Charging Station Search in Stochastic Environments: A Multi-Agent Approach. *https://arxiv.org/pdf/2204.14219.pdf*

Abstract

Range and charge anxiety remain essential barriers to a faster electric vehicle market diffusion. To this end, quickly and reliably finding suitable charging stations may foster an electric vehicle uptake by mitigating drivers' anxieties. Here, existing commercial services help drivers to find available stations based on real-time availability data but struggle with data inaccuracy, e.g., due to conventional vehicles blocking the access to public charging stations. In this context, recent works have studied stochastic search methods to account for availability uncertainty in order to minimize a driver's detour until reaching an available charging station. So far, both practical and theoretical approaches ignore driver coordination enabled by charging requests centralization or sharing of data, e.g., sharing observations of charging stations' availability or visit intentions between drivers. Against this background, we study coordinated stochastic search algorithms, which help to reduce station visit conflicts and improve the drivers' charging experience. We model a multi-agent stochastic charging station search problem as a finite-horizon Markov decision process and introduce an online solution framework applicable to static and dynamic policies. In contrast to static policies, dynamic policies account for information updates during policy planning and execution. We present a hierarchical implementation of a single-agent heuristic for decentralized decision making and a rollout algorithm for centralized decision making. Extensive numerical studies show that compared to an uncoordinated setting, a decentralized setting with visit-intentions sharing decreases the system cost by 26%, which is nearly as good as the 28% cost decrease achieved in a centralized setting, and saves up to 23% of a driver's search time while increasing her search reliability.

1. Introduction

Electric vehicles (EV) can play a crucial role in decarbonizing the transportation sector, given a rapid energy transition and continuous battery technology improvement (Schuller and Stuart 2018). To sustain the global electric vehicle market's growth of the past two years (Deloitte 2020), it is necessary to mitigate the remaining barriers to private EV adoption. In addition to the well-known range anxiety, a new phenomenon referred to as charge anxiety, caused by unreliable and insufficient public charging infrastructure, remains an essential barrier, particularly in cities (Myersdorf 2020). Besides increased price transparency, a seamless charging experience may reduce these anxieties if drivers can easily find and use an available charging station (McKinsey 2020).

In this context, increasing public infrastructure coverage and improving interoperability between charging service providers is necessary to facilitate easy and reliable access to public charging stations but requires a long planning horizon (Volkswagen 2019). Accordingly, complementary short-term solutions are necessary to alleviate deficiencies in the currently undersized charging infrastructure. In theory, cooperative charging strategies based on vehicle-to-vehicle communication can allow for fairer charging capacity allocation (You et al. 2016) but are not implemented yet due to immature technology and uncertain economic-value (Lauinger et al. 2017). In practice, existing map-based services help EV drivers to locate available charging stations based on real-time charging station availability data, but fail to provide a reliable charging station search experience: stations reported as available can be unusable due to inaccurate reporting or ICEing, i.e., conventional vehicles blocking the access to a charging station (Guillet et al. 2022). In such cases, drivers must take detours to reach another station, which may lead to increasing anxiety.

Moreover, simultaneous uncoordinated searches of multiple drivers may conflict if drivers head to the same charging station. Navigation services platforms that offer services to find available charging stations through local navigation devices or an online API, may centralize driver requests or leverage active and passive driver community input, e.g., GPS trace data, to coordinate search recommendations. In the literature, stochastic charging station search methods offer a reliable alternative to existing search services, accounting for charging stations' availability uncertainty. Such methods consider charging stations as stochastic resources and aim to find a sequence of charging station visits – a search path – that minimizes the expected search cost to reach an available station. These approaches are amenable for real-time applications, may significantly save drivers' time, and increase the search's reliability. However, such approaches so far focus always on a single-agent setting. Accordingly, coordinating drivers in a multi-agent setting has not been studied so far, but may avoid possible visit conflicts and further improve the driver's charging experience by increasing the search's reliability and decreasing the search time.

With this paper, we close this research gap by extending stochastic single-agent search algorithms to a stochastic multi-agent setting, accounting for information-sharing and possi-

ble requests centralization. We consider scenarios in which drivers may share their planned visits, their observations of a charging station's occupancy, or both. Our goal is to identify the coordination strategy that yields the best improvement on the drivers' search times and the search's reliability.

1.1. Related literature

In the following, we first review literature that relates to stochastic charging station search problems in a single-agent setting, including EV routing with uncertain charging station availabilities and more generic stochastic resource search problems. We then focus on multi-agent resource search problems with stochastic availability or locations.

Only a few papers that deal with EV routing problems address charging station availability uncertainty but do not focus on open-search problems. Kullman et al. (2021) solve the electric vehicle routing problem with a public-private recharging strategy, while Sweda et al. (2017) and Jafari and Boyles (2017) focus on shortest paths with multiple charging stops for EVs. Sweda et al. (2017) propose adaptive charging and routing strategies when utilizing the public charging infrastructure, whereas Jafari and Boyles (2017) additionally model both stochastic travel time and charging consumption. Alternatively, a few papers deal with stochastic resource search problems in general settings (Guo and Wolfson 2018, Schmoll and Schubert 2018) or more specific settings, e.g., on-street parking spots (Arndt et al. 2016) or stochastic taxi customer demand (Tang et al. 2013). Guillet et al. (2022) are the first to cover multiple variants of the stochastic charging station search problem for EVs, considering charging or waiting times at stations. However, all aforementioned papers, including Guillet et al. (2022), are limited to single-agent settings and ignore possible agents coordination.

Focusing on multi-agent settings, most works on resource search problems under uncertainty focus on cooperative searches, i.e., settings in which agents share a unique common goal. In Bourgault et al. (2003), all agents aim at locating a resource with no a-priori information on its location in a decentralized decision-making setting. Chung and Burdick (2008) solve a similar setting with centralized decision-making, while Wong et al. (2005), and Dai and Sartoretti (2020), extend the single resource target search problem to a multi-target search problem. Qin et al. (2020) address the multi-agent travel officer problem in a centralized decision-making setting, in which agents must cooperatively collect resources with stochastic availability in given locations. In all of these papers, the cooperative search terminates after the (all) resource(s) have been found. This is not the case in our multi-agent charging station search setting: here, each agent terminates her search when she found at least one non-shareable available resource. Existing work on multi-agent settings for EVs mostly focuses on autonomous EV fleet management, such as ride-sharing planning (Al-Kanj et al. 2020) or online requests matching for ride-hailing (Kullman et al. 2022) and do not cover stochastic resource search problems.

In summary, most papers that deal with heterogeneous resource search problems with

stochastic availability focus on a single-agent setting. Multi-agent search problems mostly focus on cooperative (multi) resource search problems with unknown resource locations. In contrast, the search problem studied in this work is rather collaborative than cooperative because there exists a trade-off between an individual agent's search experience and the system performance. Furthermore, while agents usually synchronously search for one or multiple resources, our setting requires the agents' search to start at different times and in different locations which are sequentially revealed.

1.2. Aims and scope

With this work, we close the research gap outlined above by providing coordinated stochastic search algorithms, that consider both resources and agents heterogeneity, tailored to various information-sharing and decision-making scenarios of high practical relevance. Specifically, our contribution is three-fold. First, we formalize the underlying centralized multi-agent decision-making problem. To this end, we define the planning problem as a single-decision-maker Markov Decision Process (MDP) and show that with an additional policy constraint, this MDP can be alternatively represented as a set of single-agent MDPs, which enables decentralized and static planning. Second, we present several online algorithms that allow to solve a variety of real-world scenarios. Here, we consider both settings in which a navigation device addresses a charging request on-the-fly, by providing either a full search path (static planning) or only the next best charging station to visit (dynamic planning) to the driver. Moreover, we consider centralized and decentralized planning settings with different levels of information-sharing in which drivers share either their planned visits, or their charging station occupancy observations, or both. Third, we conduct extensive numerical studies based on real-world instances to analyze which coordination strategy yields the highest improvement potential from a system and a driver perspective.

Focusing on a short planning horizon, our results show that, from a system perspective, a centralized coordination strategy can decrease the system cost by 28%, and that a static decentralized coordination strategy already achieves a 26% cost decrease if visit intentions are shared. In a decentralized setting with intention-sharing, observation-sharing does not increase the system's performance further, but enforcing agents' collaboration is required when drivers depart within a short time span. While a decentralized setting with only observation-sharing performs worse than intention-sharing settings, it provides a computationally efficient implementation in practice. When implemented in a dynamic setting, it yields a 10% cost decrease when drivers depart within a short time span, but achieves a 26% cost decrease with larger departure time span. From a driver perspective, coordination may save up to 23% (intention-sharing setting) of a driver's search time, while increasing her search reliability. Our results further show that a coordinated search outperforms uncoordinated searches, with respect to both best and worst solutions that an individual driver may obtain. Finally, we show in additional analyses that coordination also positively impacts a

driver's search for a longer planning horizons, yielding up to 46% decrease in system cost.

1.3. Organization

The remainder of this paper is as follows. Section 2 introduces our problem setting. In Section 3, we formalize our multi-agent decision making problem as an MDP, before we characterize MDP policies that allow for decentralized policy execution and planning. Section 4 introduces our solution framework for online charging requests. In Section 5, we describe our case study and the corresponding experimental design. We discuss numerical results in Section 6. Section 7 concludes this paper and provides an outlook on future research.

2. Problem setting

We focus on a non-adversarial multi-agent search problem, with stochastic charging station availability, where multiple drivers seek to find an unoccupied charging station in their vicinity at the earliest possible time. Drivers enter the system sequentially and reveal their current position at the same time. Accordingly, drivers start their search asynchronously; each driver departs from a given location, i.e., its current position, and may visit multiple occupied charging stations before reaching an available charging station. A driver visits the stations recommended by her navigation device, which synchronizes with a central navigation service platform. In this setting, an EV driver's objective is to minimize her expected time to reach an available station and any related charging costs. The navigation service provider's objective is to satisfy as many drivers as possible by minimizing the sum of all drivers' individual search cost as well as the likelihood that a driver does not reach any available station within a given time budget.

We assume each individual search to be spatially and temporally bounded to account for a driver's limited time budget. Every unsuccessful search induces an individual penalty. Moreover, we assume that a driver cannot wait at an occupied station nor visit a station twice and that she stops at the first available station she visits. Stations are heterogeneous and using a station yields a cost. Drivers are heterogeneous with respect to the charging and penalty costs, their time budget, and the radius delimiting their (circular) search area.

Practically, the navigation device transmits a search solution to the driver: the driver may either receive a full sequence of stations to visit until an available station is reached, i.e., a search path, or she may receive the next station to visit dynamically. We refer to the former as static planning and to the latter as dynamic planning. Moreover, the solution planning can be (i) centralized within the navigation service platform or (ii) decentralized, i.e., at agent-level. In the latter case, solution planning can happen directly within the local navigation device and devices only use the platform to share information with each other. In both cases, agents may share their station occupancy observations intermittently or in

real-time with the central platform. In the decentralized case, they may additionally share the planned charging station visits. To capture these varying characteristics which are of practical relevance, we introduce the following problem settings as summarized in Table 1.

Static planning: In such settings, a search path is planned upon request of the agent and cannot be dynamically updated once the agent started her search. Accordingly, an agent's actions only depend on the agent's own observations and the initial information available prior to her search. We assume that planned search paths and (if shared) collected observations, can be transmitted to the central platform to be available to subsequent agents. We introduce four decentralized decision-making settings according to the type of shared information. In the DEC setting, no information is shared between agents, while each agent is aware of prior availability observations of other agents when computing her search path in the DEC-O setting. Both settings DEC & DEC-O are purely informative, such that agents are unaware of other agents' planned search paths. The DEC-I and DEC-IO extend the DEC & DEC-O settings: each agent additionally knows previous agents' search paths, i.e., their intended station visits and visit times. Agents may use the information selfishly, e.g., by visiting other agents' target stations first, or they may use the information collaboratively.

Dynamic planning: In such settings, an agent does not receive the whole search path at once. Instead, it receives the next station to visit at each occupied visited station until it reaches an available station. Here, we consider a centralized-decision making setting (CEN) in which a central planner, i.e., the navigation platform, is fully aware of all agents' observations and actions, and assigns station visits on the fly to minimize the total expected search cost of all agents, while maximizing the likelihood that all agents find an available station. Here, each action assigned to an agent depends on the other agents' actions and all decisions are system-optimized as the platform does not prefer any agent. Further, we consider a dynamic variant (DEC-O-dyn) of the decentralized observation-sharing setting DEC-O. In this setting, the initial solution is dynamically re-planned upon each occupied station visit to account for the latest shared observations.

Note that we discard CEN with pure observation-sharing, DEC-I-dyn, and DEC-IO-dyn

Table 1: Problem settings overview

	visit intentions	availability observations	path planning	decision-making	type
DEC			static	decentralized	selfish
DEC-I	✓		static	decentralized	collaborative
DEC-O		✓	static	decentralized	informative
DEC-IO	✓	✓	static	decentralized	collaborative
CEN	✓	✓	dynamic	centralized	collaborative
DEC-O-dyn		✓	dynamic	decentralized	informative

from our analyses for the following reasons: a CEN setting with pure observation-sharing does not exist, as the planner who centrally assigns stations to drivers is aware of all drivers' visit intentions. In DEC-I-dyn, intention-sharing implies observation-sharing, as an intention update occurs when a driver visits an occupied station, which reveals the driver's station occupancy observation. In DEC-IO-dyn, each driver has nearly as much information as a centralized planner, at the exception of other drivers' search radii or time budget preferences. Since both settings are very similar, we focus on CEN, which requires less information sharing in practice.

A few comments on our problem setting are in order. First, we model our coordinated charging station search as a closed system. While this assumptions seems to be a limitation from a theoretical perspective, it is appropriate from a practitioners perspective as it reflects real-world planning scenarios, where charging requests usually accumulate during certain periods of the days (see Figure 1). Even for scenarios with charging requests homogeneously distributed during a day, we observe a sufficiently long period without charging requests during the night which imposes a natural system boundary. Second, we mostly focus on short planning horizons and assume in this context constant availability probabilities and observations persistency over the planning horizon. This assumption is in line with practice for the following reason: in a single-agent scenario, modeling time-dependent recovering availability probabilities has no impact on the computed search path as an agent either succeeds or gives up the search after a limited time budget (cf. Guillet et al. 2022). By restricting our experimental studies to short planning horizon scenarios, we can analogously ignore time-dependent recovering probability modeling. However, we discuss how to account for relaxed constant probability assumptions as well as for relaxed observations persistency (see Section 4.3), and detail the impact of coordination in longer planning horizon settings (see Section 6.2.3). Third, we assume that all the shared information is available to the other drivers or the central decision-maker, but limited to a certain vicinity. This assumption allows to reduce computational effort and is reasonable in practice as far-distanced drivers do not interfere with each other.

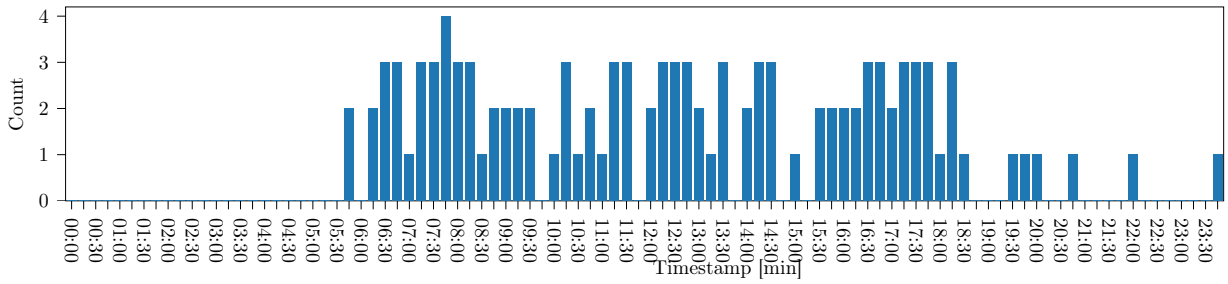


Figure 1: Number of ending trips on Tuesday 12.01.22 in the east-area of Berlin, Germany

The figure shows 1/3 of all collected anonymized OD pairs of all vehicles using TomTom navigation services, that end their trips in the east-area of Berlin.

3. MDP representation

We refer to the problem setting discussed in Section 2 as the multi-agent stochastic charging pole search (MSCPS) problem and model it as a sequential multi-agent decision making problem with a finite time-horizon. In the following, we first consider a centralized representation of the system states, and represent an offline multi-agent search with omniscient single decision-maker, i.e., assuming that all future requests are known ahead, as a finite-horizon MDP in Section 3.1. We then reduce the solution space such that the solution policy can be decentrally executed in Section 3.2, and show that in this case the MDP can be represented by a set of individual MDPs in Section 3.3.

We formalize the MSCPS problem on a complete directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ consisting of a set of vertices $\mathcal{V}$ and a set of arcs $(v, \acute{v}) \in \mathcal{A}$, where a vertex $v \in \mathcal{V}$ consists of a charging station or a revealed driver's departure location. For each agent $i \in \mathcal{D}$ let t_0^i be her departure time and v_0^i be her start vertex. Each agent i has a defined time budget $\bar{T}^i$ and we define the total planning horizon $T = [0, \max_{i \in \mathcal{D}}(t_0^i + \bar{T}^i)]$ during which all searches occur. Each agent can charge at any charging station located at a vertex $v \in \mathcal{V}$ within a limited search radius $\bar{S}^i$. We denote with $t_{v,\acute{v}} \geq 0$ the time to drive from v to $\acute{v}$. An unsuccessful search yields a penalty cost $\bar{\beta}^i$ for the agent. We let γ_v^i be the (time-equivalent) cost for using pole v for agent i, if v is available upon the arrival of i. We let p_v be the probability that station v is initially free. Finally, we assume in our basic setting that an occupied station remains occupied during the whole planning horizon T.

3.1. MDP notation

An agent triggers a new decision epoch either by requesting to charge her vehicle or by observing a new station. We refer to the requesting or observing agent as the *deciding agent* denoted with λ although a central planner may take the actual decision. If the observed station is occupied and there is at least one reachable station within her remaining time budget, λ selects her next station visit; otherwise λ has terminated her search. Note that each station visited by λ can no longer be used by any succeeding agent, since the station is either (i) already occupied, or (ii) available and thus occupied by λ. Accordingly, the set of visited stations, i.e., stations that have been observed, corresponds to the set of occupied stations.

State space: We represent a system state $x \in \mathcal{X}$ out of state space $\mathcal{X}$ as

$$x = (\mathbf{x_d}, \mathcal{J}, \mathcal{T}, \mathcal{O}) \,, \tag{3.1}$$

with $\mathcal{J} \subseteq \mathcal{D}$ being the set of active agents, $\mathcal{T} \subseteq \mathcal{D}$ being the set of successfully terminated agents, $\mathcal{O} \subseteq \mathcal{V}$ being the set of all visited stations, and $\mathbf{x_d} = (x^i)_{\forall i \in \mathcal{J} \bigcup \mathcal{T}}$ being the vector that describes the state of each agent. Here, we define an agent's state x^i as

$$x^i = (v^i, t^i, s^i) \,, \tag{3.2}$$

with $v^i \in \mathcal{V}$ being the station assigned to agent i in state x; t^i being the arrival time at v^i

and $s^i \in \{\mathrm{d}, \mathrm{f}, \mathrm{t}, \mathrm{r}\}$ being the status of the agent: an agent can either (i) be en-route to the station ($s^i = \mathrm{r}$), unaware of v^i's realized availability, (ii) observe v^i to be available, which successfully terminates her search ($s^i = \mathrm{f}$), (iii) observe v^i to be occupied, having sufficient time to reach a new station ($s^i = \mathrm{d}$) or (iv) not ($s^i = \mathrm{t}$), which unsuccessfully terminates her search. The observation of v^i in (ii) and (iii) triggers a new decision epoch.

Action space and immediate cost: We denote with u, the action taken in state x for agent λ, i.e., the next station to visit, and by $\mathcal{U}(x)$ the set of feasible actions, such that $u \in \mathcal{U}(x)$. We let $d(x, u)$ be the cost immediately induced by taking decision u in state x, which does not depend on any future uncertainty realization. For clarity, we refer to state x as x^{s} if the station observed by λ is available, or as x^{f} if λ observes an occupied station or begins her search.

Available station (x^{s}) : Here, the status of λ is $s^\lambda = \mathrm{f}$, such that no further decision can be made; accordingly $u = \emptyset$ such that λ belongs to the set of terminated agents ($\mathcal{T}$). The immediate cost is the cost for λ to use v^λ, such that $d(x, u) = \gamma^\lambda_{v^\lambda}$.

Occupied station (x^{f}): Here, the status of λ is $s^\lambda = \mathrm{d}$ if λ can select an unvisited station v to visit next. Accordingly we get $u \in \{v : v \in \mathcal{V}, v \notin \mathcal{O}, t^\lambda - t^\lambda_0 + t_{v^\lambda, v} \leq \bar{T}^\lambda\}$. The immediate cost results to the driving time for λ from her current station to the chosen station, such that $d(x, u) = t_{v^\lambda, u}$. If no station can be reached within λ's remaining time budget, λ has failed her search, which induces an immediate driver penalty $\bar{\beta}^\lambda$. In this case, $s^\lambda = \mathrm{t}$ and we set $d(x, u) = \bar{\beta}^\lambda$. At the next stage, its new assigned station is $\acute{v}^\lambda = u$, while its new arrival time is $\acute{t}^\lambda = t^\lambda + t_{v^\lambda, u}$.

We define a policy $\pi \in \Pi$ as the state-action mapping function, such that $\pi(x) \in \mathcal{U}(x)$.

State transition: Upon a single-agent's action, the system transitions from state x to the next state $\acute{x}$, with $\acute{\lambda}$ being the new deciding agent. The new state $\acute{x}$ can either be a successful state $\acute{x}^{\mathrm{s}}$ for $\acute{\lambda}$ with probability $p_{\acute{v}}$ or an unsuccessful state $\acute{x}^{\mathrm{f}}$ with probability $1 - p_{\acute{v}}$.

We introduce the value function $V^\pi(x)$, that corresponds to the expected sum of future costs obtained when executing π from state x, and that can be recursively expressed as

$$V^\pi(x) = d(x, \pi(x)) + p_{\acute{v}} V^\pi(\acute{x}^{\mathrm{s}}) + (1 - p_{\acute{v}}) V^\pi(\acute{x}^{\mathrm{f}}), \tag{3.3}$$

with $x \in \{x^{\mathrm{s}}, x^{\mathrm{f}}\}$. Then, our objective is to find a policy π^* that minimizes the accumulated costs when executing π^* from the initial state x_0, i.e, $V^{\pi^*}(x_0) \leq V^\pi(x_0) \ \ \forall \pi$.

To increase the robustness of our solution, we introduce a global penalty cost β^G that is induced in a termination state in case at least one agent unsuccessfully terminated her search. We note that a higher β^G may decrease the quality disparities of the single-agent solutions by favoring a conservative system with equally bad solutions. On the contrary, a lower β^G may favor single-agent high quality solutions at the detriment of other agents. We define x_n as a termination state as soon as the set of active agents $\mathcal{J}$ becomes empty. In state x_n, each agent $i \in \mathcal{T}$ has either successfully ($s^i = \mathrm{f}$) or unsuccessfully ($s^i = \mathrm{t}$) terminated her search. We define $V(x_n)$, with x_n being a termination state, as

$$V^\pi(x_n) = d(x_n, \pi(x_n)) + \beta^G \delta \ , \tag{3.4}$$

with δ being the binary variable that indicates whether at least one agent $i \in \mathcal{T}$ has failed her search. Table 6 (in Appendix A) summarizes the notation used to define the MSCPS problem and the related MDP.

3.2. Policy representation

In Section 3.1, we introduced the policy function π which indicates for any encountered state x the action $\pi(x)$ to take, assuming that a state-action pair $(x, \pi(x))$ can apply to any agent. Alternatively, we can represent π as a set of single agent policies π^i by defining π^i only for states where agent i takes a decision. With λ being the deciding agent in state x, we let $\pi^i(x) = \pi(x)$ if $i = \lambda$ in state x and $\pi^i(x) = \emptyset$ if $i \neq \lambda$. We refer to this joint set of single-agent policies $\{\pi^i\}_{i \in \mathcal{D}}$ as the agent-based representation of π.

User-dependent single-agent policies: In general, the execution of a single-agent policy π^i is user-dependent and must be centrally coordinated as each agent's action depends on other agents' observations. In state x, the deciding agent λ is aware of the full state x, i.e., aware of all other availability realizations observed by all other agents. From a single-agent perspective, the station selection for λ depends on whether and where other agents terminated. For all states x corresponding to λ located at v, selected actions $\pi^i(x)$ might not be identical. We let Π^{dep} be the set that contains user-dependent policies $\pi = \{\pi^i\}_{i \in \mathcal{D}}$, with π^i being user-dependent.

User-independent single-agent policies: While coordinated search in general requires central coordination, it is possible to ensure a-priori coordination between agents without central coordination, i.e., preserving user-independent search policies. In this special case, coordination accounts for other agent's visit intentions when deriving an agent's search policy but excludes observation sharing during the search. In such a setting, an agent i, executing a search policy π, being located at station v with v being occupied, will always visit the same next station. We can formally describe this condition as

$$\pi^i(x) = \pi^i(\bar{x}),\ \forall (x, \bar{x}) \in \bar{\mathcal{X}} \times \bar{\mathcal{X}} \text{ st. } v^i = \bar{v}^i = v, \tag{3.5}$$

with v^i being the location of agent i in state x, $\bar{v}^i$ being the location of i in state $\bar{x}$, and $\bar{\mathcal{X}}$ being the subset of states possibly reachable from x_0 through π. We let Π^{ind} be the set that contains user-independent policies $\pi = \{\pi^i\}_{i \in \mathcal{D}}$, with π^i being user-independent.

3.3. MDP representation with a user-independent policy constraint

The MDP defined in Section 3.1 considers a user-dependent global policy that can be represented as a set of single-agent user-dependent policies (see Section 3.2). We show that by constraining single-agent policies to be user-independent, this global MDP representation simplifies to a set of single-agent MDPs, which allows to easily plan each agent's solution

decentrally. We introduce the representation of a single-agent MDP in Section 3.3.1 accordingly. Here, the objective function equals the sum of single-agent MDP objective functions with an extra penalty cost. In this case, a single-agent policy π^i translates to an ordered sequence of station visits $C^i = (v_0^i, \ldots, v_n^i)$ with i starting at v_0^i, following the vertices in sequence, and terminating either the search at any first available charging station $v \in C^i$, or unsuccessfully at v_n^i. Section 3.3.2 then focuses on representation equivalence.

3.3.1. Single-agent MDP notation:

Analogously to Guillet et al. (2022), we model each driver's individual search process as a single-agent finite-horizon MDP. Let $\mathcal{S}$ be the (single-agent) state space and $x \in \mathcal{S}$ be a state defined as $x = (C^i, a)$, with $C^i = (v_0^i, ..., v_k^i)$ being the sequence of visited stations, and a being the realized availability at the last visited station v_k^i of C. If $a = 0$, then the agent takes an action u that consists of the single station selection decision $u = (v)$, with $v \in \tilde{\mathcal{V}}, v \notin C^i$, and $\tilde{\mathcal{V}}$ being the set of reachable stations from v_k^i. In this case, the transition function $p_t^i(\acute{x}|x, u)$ describes the probability for the agent i to be in state $\acute{x}$ after having taken action u in state x. The immediate cost induced for taking action $u = (v)$ in state x results to the travel time $t_{v_k^i, v}$ between v_k^i and v, or to the penalty cost $\bar{\beta}^i$, if the agent already spent her time budget. If $a = 1$, then the agent successfully terminated her search at v_k^i and the agent-specific station usage cost $\gamma_{v_k^i}^i$ results.

Policy π^i is the function that maps the planned action u for agent i in each encountered state x, and defines the related search path $C^i(\pi^i) = (v_0^i, ..., v_n^i)$, with $\pi^i((v_0^i, ..., v_k^i), 0) = v_{k+1}^i \;\; \forall\, k \in [0, ..., n-1]$.

Agent i aims to find a policy π^i that minimizes her search cost $F^{\pi^i}((v_0^i), 0)$, with F being the single-agent value function, which is defined as

$$\begin{aligned} F^{\pi^i}(x) =& t_{v_k,v} + \sum_{\acute{x} \in \mathcal{S}} p_t^i(\acute{x}|x, \pi^i(x)) F^{\pi^i}(\acute{x}) \\ \Leftrightarrow F^{\pi^i}(C, 0) =& \, t_{v_k,v} + p_t^i((C', 1)|(C, 0), \pi^i(C, 0)).\gamma_v^i \\ &+ p_t^i((C', 0)|(C, 0), \pi^i(C, 0)) F^{\pi^i}(C', 0) \;, \end{aligned} \tag{3.6}$$

with $x = (C, 0)$ being a non-termination state, $\acute{x} = (C', a')$ and $C' = (v_0, ..., v_k, v)$.

Transition functions: To define our transition functions, we recall that station occupancy observations are persistent for the whole planning horizon and a-priori availability probabilities are identical for all agents by assumption, which leads us to the following observation: if the first station v visited by j is available, j stops and charges there, otherwise v is occupied and remains occupied during the search horizon. Accordingly, v will never be available to another agent i who intends to visit station v after j. More generally, the probability that v is available to agent i equals the probability that all agents j intending to visit v before i ($t_v^i \geq t_v^j$) will have found a free station before their expected visit to v, and that station v is available during the search.

Formally, we denote the probability that station v is available for agent i at time t

with $p_v^i(t)$. We let $\rho^i(t)$ be the probability that agent i found at least one station available among all stations v, from sequence $C^i = (v_1^i, ..., v_n^i)$, for which the visit time t_v^i is lower than t, and define $\rho^i(t)$ as

$$\rho^i(t) = 1 - \prod_{v \in C^i, t_v^i \leq t} (1 - p_v^i(t)). \tag{3.7}$$

Then, $p_v^i(t)$ reads

$$p_v^i(t) = p_v \prod_{j \in \mathcal{D}, j \neq i, t_v^j \leq t} \rho^j(t_v^j). \tag{3.8}$$

Note that both definitions are finitely nested as visits are ranked by agent's arrival time.

We now define user-dependent transition function as

$$\begin{aligned} p_t^i((C', 1)|(C, 0), (v)) &= p_v^i(t'), \\ p_t^i((C', 0)|(C, 0), (v)) &= 1 - p_v^i(t'), \end{aligned} \tag{3.9}$$

with $C := C^i$. We let C' be sequence C extended by station v, and t' be the accumulated driving time for sequence C'.

In a single-agent setting, transition functions are independent of other agents planned actions and simplify to

$$\begin{aligned} p_t((C', 1)|(C, 0), (v)) &= p_v, \\ p_t((C', 0)|(C, 0), (v)) &= 1 - p_v. \end{aligned} \tag{3.10}$$

3.3.2. System evaluation cost:

With F^{π^i} being the single-agent value function for agent i, we now introduce a cost that jointly evaluates all single-agent policies. Let $\alpha^i = F^{\pi^i}(x_0^i)$ be the cost that explicitly evaluates the expected value of the policy cost assigned to i in its initial state. Cost α^i can be decomposed (cf. Guillet et al. 2022) as

$$\alpha^i = (1 - \rho^i)\beta^i + A^i(\pi^i) \ . \tag{3.11}$$

We let $\bar{\rho}^i(\pi, k)$ be the probability that agent i fails in finding at least one free station in $C^i_{[0:k]}$, while ρ^i denotes the probability to find at least one free station in the whole sequence C^i. Here we define $A^i(\pi^i)$ as

$$A^i(\pi^i) = \sum_{k=0}^{n-1} [t_{v_k^i, v_{k+1}^i} \bar{\rho}^i(\pi^i, k)] + \sum_{k=1}^{n} \gamma_{v_k^i}^i p_{v_k^i}^i (C_{[0:k-1]}) \bar{\rho}^i(\pi^i, k-1) \ . \tag{3.12}$$

We define the cost α^π that jointly evaluates the system policy $\pi = \{\pi^i\}_{i \in \mathcal{D}}$, i.e., the set of single-agent user-independent policies, as

$$\alpha^\pi = \sum_{i \in \mathcal{D}} \alpha^i + (1 - \prod_{i \in \mathcal{D}} \rho^i)\beta^G, \tag{3.13}$$

with α^π being the sum of all expected single-agent MDP costs and a cost $(1 - \prod_{i \in \mathcal{D}} \rho^i)\beta^G$ that penalizes the system with respect to the number of agents that could not successfully finish their search. Quantity $(1 - \prod_{i \in \mathcal{D}} \rho^i)$ represents the likelihood that at least one agent did not successfully terminate her search.

We now show that for such a policy π, the joint cost α^π equals the value function V^π evaluated in the initial global state x_0.

Proposition 1. *Let policy π be a set of single-agent user-independent policies, such that $\pi \in \Pi^{ind}$. Then*

$$\alpha^\pi = V^\pi(x_0),$$

with V^π being defined in Equation 3.3 and α^π being defined in Equation 3.13.

Proposition 1 simplifies the representation of the centralized MDP to a set of single-agent MDPs, which enables us to devise decentralized online algorithms in Section 4.

4. Online solution planning

In the following, we present our online heuristics to process sequentially revealed charging requests, i.e., drivers entering the system are unknown ahead of time. Accordingly, the set of agents $\mathcal{D}$ is initially empty and we update $\mathcal{D}$ each time a new charging request enters the system.

4.1. Static policy planning

For static policy planning, we focus on decentralized decision-making settings (see Section 2). We plan each agent's search path with a modified version of the stochastic search algorithm developed in Guillet et al. (2022), by taking into account the latest available information, i.e., the latest shared visit intentions or the latest availability observations. In the following, we first briefly outline our algorithm (Section 4.1.1) in its basic variant without any information-sharing, before we detail the required changes to account for observation-sharing (Section 4.1.2), visit intention-sharing (Section 4.1.3), or both (Section 4.1.4).

4.1.1. No information-sharing (DEC)

This setting corresponds to a (fully-decentralized) single-agent setting, in which each agent is unaware of any prior requested search paths and availability observations. In practice, this setting equals planning routes on individual non-communicating navigation devices or with a stateless navigation service platform, which does not retain information about past requests. Here, each agent aims to minimize her individual cost α^i, with α^i being defined based on a single-agent MDP with *user-independent* transition functions (see Equation 3.10). As shown in Guillet et al. (2022), a multi-label setting algorithm with a heuristic dominance criterion, which we refer to as LH, can efficiently solve this problem setting.

In general, a multi-label setting algorithm propagates partial policies to find a cost-optimal policy, maintaining a list of all explored and non-dominated partial policies in cost-increasing order. A partial policy π_v^i describes a given search path starting from the initial vertex v_0

and ending at v. We associate each partial policy π_v^i with a label L_v^i, associated to vertex v and agent i, defined as $L_v^i = (t_v^i, A_v^i, \rho_v^i, \alpha_v^i)$. A label L_v^i consists of the accumulated driving time t_v^i, the partial cost A_v^i, the overall probability of success ρ_v^i, and the total cost for agent i (see Equation 3.11). We recall that A_v^i and ρ_v^i result from the decomposition of cost α_v^i (see Section 3.3.1). At each exploration step, the algorithm retrieves the minimum-cost partial policy π_v^i and propagates her related label L_v^i to all unvisited vertices $\acute{v}$, reachable from v. For each vertex $\acute{v}$, the algorithm discards the propagated label $L_{\acute{v}}^i$ if it is dominated by any other label at vertex $\acute{v}$, and otherwise discards any labels that $L_{\acute{v}}^i$ may dominate. Specifically, considering two partial policies π_1^i and π_2^i for agent i that end with the same vertex visit v and their associated labels $L_v^i(\pi_1^i)$ and $L_v^i(\pi_2^i)$, we say that $L_v^i(\pi_1^i)$ dominates $L_v^i(\pi_2^i)$ ($L_v^i(\pi_1^i) \succ L_v^i(\pi_2^i)$), if

$$1 - \rho_v^i(\pi_1) \leq 1 - \rho_v^i(\pi_2) \;, \tag{4.1}$$

$$A_v^i(\pi_1) \leq A_v^i(\pi_2) \tag{4.2}$$

are true.

Finally, the labeling procedure returns the – non-dominated – minimum-cost label, that describes the search policy π^i with minimum cost α^i. The used dominance relation does not guarantee the optimality of π^i in a single agent setting but provides close to optimal solutions with significant runtime savings compared to an exact dominance relation (cf. Guillet et al. 2022). For a detailed pseudo-code of this algorithm, we refer to Appendix C. We solve each incoming request with this algorithm and refer to its sequential application as hierarchical label-setting, denoted with HLH.

4.1.2. Occupancy observation-sharing (DEC-O)

In this setting, each agent i knows about occupied stations observed by other agents prior to her search. To account for this knowledge, we remove observed occupied stations from an agent's action space because occupied stations cannot be freed up during the remaining planning horizon. Accordingly, an action u for i consists of the single station selection decision $u = (v)$ with $v \in \tilde{\mathcal{V}}, v \notin C^i, v \notin \mathcal{O}$, $\tilde{\mathcal{V}}$ being the set of reachable stations from v_k^i, and $\mathcal{O}$ being the set of observed occupied stations. To account for this modified action space, we reduce the charging station network graph to unvisited stations whenever we compute a new search path. Accordingly, this allows us to use the HLH algorithm for the no-information-sharing setting to compute each agent's search path.

4.1.3. Visit-intentions sharing (DEC-I)

In this setting, an agent i knows all station visit intentions of agents who started their search prior to i, i.e., all $\pi^j \; \forall j \in \tilde{\mathcal{D}}$, with $\tilde{\mathcal{D}}$ being the set of preceding agents. Here, we define cost α^i based on a single-agent MDP with *user-dependent* transition functions (see Equation 3.9). Given i is the i^{th} requesting agent, we let $\pi_{i-1} = \{\pi^j\}_{\forall j \in \tilde{\mathcal{D}}}$ be the joint

policy of the (sub)system prior to i's request, i.e., the joint set of all single-agent policies for agents in $\{1, ..., i-1\}$. Then, the probability of station v to be available to i results from Equation 3.8, by replacing $\mathcal{D}$ by $\tilde{\mathcal{D}}$

$$p_v(t) = p_v \prod_{j \in \tilde{\mathcal{D}}, j \neq i, t_v^j \leq t_v^i} \rho^j(t) \ , \tag{4.3}$$

with $\rho^j(t)$ being defined in Equation 3.7.

Each agent i uses the information about other agent's visit intentions to individually optimize her own search path, which may occur at the detriment of the agents that started their search earlier. In some cases, agent i may bypass another agent j by visiting the intended stations of j before j's expected visit times.

Collaborative intention-sharing: To avoid the selfish use of intention-sharing, we design a hierarchical solution method such that agents minimize their search times without compromising other agents' success. Here, agent i does not optimize her search path with respect to her individual cost; instead she optimizes her search path with respect to the cost of the subsytem that includes herself and already planned policies π_i of other agents. We refer to this collaborative implementation as HLH-c. To optimize the agent's policy with respect to the subsystem cost, we compute n candidate policies for agent i using our LH algorithm, which queues the evaluated policies by cost-increasing order. Out of these candidate policies, we then select the policy that yields the lowest subsystem cost.

Formally, let $\Gamma^i = \{\pi_k^i : \forall k \in [0, n]\}$ be the set that contains these n candidate policies for agent i. Then, we (i) compute Γ^i using LH, evaluate the joint cost of the subsystem policy π_{i-1} and the newly planned policy, i.e., $\pi_i = \pi_{i-1} \circ \{\pi^i\}$ for all $\pi^i \in \Gamma^i$, and (ii) select π^{i*} among Γ^i that minimizes α^{π_i} as

$$\pi^{i*} = \underset{\pi^i \in \Gamma^i}{\arg\min}\, \alpha^{\pi_{i-1} \circ \{\pi^i\}} \ . \tag{4.4}$$

If agents are heterogeneous, an agent i needs to know other agents' parameters $\bar{\beta}^i$ and γ_v^i to compute the exact value $\alpha^{\pi_{i-1} \circ \{\pi\}}$. In practice, these parameters can be either shared or approximated.

4.1.4. Intentions & Occupancy observations sharing (DEC-IO):

In this setting, an agent i combines both knowledge about past agents' observations and visit intentions when planning her search path at time t^i. Similar to DEC-O, we remove occupied stations from an agent's action space, and similar to DEC-I, transition functions are user-dependent. In this setting, we do not need to account for an agent j's remaining visit intentions, if j started earlier than i and successfully finished her search already. If this agent j is not terminated yet at planning time t^i, we know that all stations visited by j before t^i are occupied, such that we only account for j's visit intentions that would occur later than t^i. Accordingly, we truncate j's search path C^j to the stations not visited yet at t^i

and refer to the truncated sequence as $\bar{C}^j$. We obtain availability probabilities by replacing C^j by $\bar{C}^j$ in Equations 3.7&3.8.

4.2. Dynamic policy planning

In the following, we describe the methodology for dynamic policy planning.

Focusing on a centralized-decision maker, we utilize two different algorithms to dynamically solve the large-scale MDP introduced in Section 3.1. The first algorithm is a rollout algorithm (RO) with a one-step decision rule as described in Goodson et al. (2017), which is known to provide high-quality solutions in similar settings. This rollout algorithm explores the MDP solution tree partially, using a base-policy to approximate the value function. In each state, the algorithm selects the action that yields the lowest approximated cost. The second algorithm bases on a dynamic implementation of our HLH-c algorithm (see Section 4.1.3). Instead of selecting the next best station visit based on a partial MDP solution tree exploration, this algorithm (re)computes an agent's individual search path using the latest observations and visit intentions available at each decision step. We then use the first station visit of the recomputed search path as the next station visit. We refer to this second algorithm as LH-RO and note that it combines dynamic and offline planning similar to the work of Ulmer et al. (2019).

Rollout algorithm (RO): Figure 2 details the pseudo-code of our algorithm which dynamically solves the MDP defined in Section 3.1. We initialize the set of active and terminated agents, the set of observed stations, and the vector that describes the state of each agent (l.1). A new request or a new station visit triggers a new decision epoch (l.3). In case this is not the first decision epoch, we update the status of the last deciding agent from $s^\lambda = \mathrm{d}$ to $s^\lambda = \mathrm{r}$ and her assigned station to v^*. We assume that the system state x gets implicitly updated upon each new decision epoch. The current agent λ can observe v^λ as available and successfully terminate her search (l.7&8), as occupied (l.9), or may start her search, such that we add λ to $\mathcal{J}$ (l.10&11). In case v^λ is not available, the deciding agent λ must select the next best station v^* to visit. For each feasible station that λ may visit (l.13) and for both possible states $\acute{x}^\mathrm{f}$ and $\acute{x}^\mathrm{s}$, the function `greedyCost`$(\acute{x})$ approximates the related cost-to-go $V^{\tilde{\pi}}(\acute{x})$ by following the greedy base-policy $\tilde{\pi}$ until a termination state or until K decision epochs are reached. Given the explicit values of $V^{\tilde{\pi}}(\acute{x}^\mathrm{s})$ and $V^{\tilde{\pi}}(\acute{x}^\mathrm{f})$, the best station $\pi(x) = v$ minimizes the cost-to-go (l.16,17&18) as

$$\pi(x) = \underset{u=(v)\in\mathcal{U}(x)}{\arg\min}\ d(x,u) + p_v V^{\tilde{\pi}}(\acute{x}^\mathrm{s}) + (1-p_v)V^{\tilde{\pi}}(\acute{x}^\mathrm{f})\ . \tag{4.5}$$

If there is no reachable station within λ's remaining time budget (l.19), λ has failed her search. The search continues over the overall planning horizon as long as new decision epochs are triggered (l.2&3).

Dynamic HLH-c algorithm (LH-RO): We now use the dynamic variant of the HLH-c algorithm to solve the CEN setting and detail the pseudo code in Figure 3. Upon each agent's request, the algorithm plans a user-independent policy and dynamically refines it at each agent's decision epoch using the latest updated information, i.e., station occupancy observations and visit intention updates.

Similar to the online rollout algorithm, we initialize our variables and compute a visit recommendation each time an agent starts her search or visits an occupied station. We add the agent to the set of terminated agents if it finds an available station (l.7&8). In addition, we initialize the set Π that contains for each active agent $i \in \mathcal{J}$ her last computed user-independent policy. The procedure `getBestPaths` (l.15) executes the single-agent label-setting algorithm LH that returns the n best candidate single-agent policies for λ. We then

Figure 2: Online rollout algorithm

1: $\mathcal{J} \leftarrow \emptyset$, $\mathcal{T} \leftarrow \emptyset$, $\mathcal{O} \leftarrow \emptyset$, $\mathbf{x} \leftarrow \mathbf{0}$, $x \leftarrow (\mathcal{J}, \mathcal{T}, \mathcal{O}, \mathbf{x})$
2: **while** True **do**
3: **if** `newEpochTriggered`() **then**
4: **if** $\mathcal{J} \neq \emptyset \wedge s^\lambda ==$ d **then**
5: $s^\lambda \leftarrow$ r, $v^\lambda \leftarrow v^*$, $t^\lambda \leftarrow t^\lambda + t_{v_k,v^*}$
6: $\lambda \leftarrow$ `decidingAgent`(), $v^\lambda \leftarrow$ `observedStation`()
7: **if** $a_{v^\lambda} == 1$ **then**
8: $\mathcal{O}$.add(v^λ), $\mathcal{T}$.add(λ), $\mathcal{J}$.pop(λ), $s^\lambda \leftarrow$ f
9: **else**
10: **if** $\lambda \notin \mathcal{J}$ **then**
11: $\mathcal{J}$.add(λ), $s^\lambda \leftarrow$ d
12: $v^* \leftarrow 0$, $Q^* \leftarrow \infty$
13: **for** $v \in \mathcal{U}(x)$ **do**
14: **for** $\acute{x}^{\mathrm{s}}, \acute{x}^{\mathrm{f}} \in T(x, v)$ **do**
15: $V^s \leftarrow$ `greedyCost`($\acute{x}^{\mathrm{s}}$), $V^f \leftarrow$ `greedyCost`($\acute{x}^{\mathrm{f}}$)
16: $Q \leftarrow t_{v_k,v} + (1 - p_v)V^f + p_v V^s$
17: **if** $Q < Q^*$ **then**
18: $Q^* \leftarrow Q$, $v^* \leftarrow v$
19: **if** $v^* == 0$ **then**
20: $s^\lambda \leftarrow$ t

Figure 3: Label-based heuristic (LH-RO)

1: $\mathcal{J} \leftarrow \emptyset$, $\mathcal{T} \leftarrow \emptyset$, $\mathcal{O} \leftarrow \emptyset$, $\mathbf{x} \leftarrow \mathbf{0}$, $x \leftarrow (\mathcal{J}, \mathcal{T}, \mathcal{O}, \mathbf{x})$, $\Pi \leftarrow \emptyset$
2: **while** True **do**
3: **if** `newEpochTriggered` **then**
4: **if** $\mathcal{J} \neq \emptyset \wedge s^\lambda ==$ d **then**
5: $s^\lambda \leftarrow$ r, $t^\lambda \leftarrow t^\lambda + t_{v^\lambda,v^*}$, $v^\lambda \leftarrow v^*$,
6: $\lambda \leftarrow$ `decidingAgent`(), $v^\lambda \leftarrow$ `observedStation`(), Π.pop(π^λ)
7: **if** $a_{v^\lambda} == 1$ **then**
8: $\mathcal{O}$.add(v^λ), $\mathcal{T}$.add(λ), $\mathcal{J}$.pop(λ), $s^\lambda \leftarrow$ f,Π.pop(π^λ)
9: **else**
10: **if** $\lambda \notin \mathcal{J}$ **then**
11: $\mathcal{J}$.add(λ), $s^\lambda \leftarrow$ d
12: Π.pop(π^λ)
13: $\pi^* \leftarrow 0$, $\alpha^* \leftarrow \infty$
14: $T \leftarrow \bar{T}^\lambda - t^\lambda$
15: **for** $\pi \in$ `getBestPaths`($v^\lambda, T, \Pi, \mathcal{O}$) **do**
16: $\alpha \leftarrow$ `getCost`($\Pi \circ \pi$)
17: **if** $\alpha < \alpha^*$ **then**
18: $\alpha^* \leftarrow \alpha$, $v^* \leftarrow \pi(x)$, $\pi^* \leftarrow \pi$
19: **if** $v^* == 0$ **then**
20: $s^\lambda \leftarrow$ t
21: **else**
22: Π.add(π^*)

select policy π^* that minimizes the subsystem cost $\alpha^{\Pi \circ \pi^*}$ (l.16,17&18) as

$$\pi^* = \arg\min_{\pi \in \Gamma^\lambda} \alpha^{\Pi \circ \pi}, \tag{4.6}$$

with Γ^λ being the set that contains the n best policies for λ, and with Π being the set of individual policies for all agents except λ. We let the procedure `getCost` (l.16) compute the value of $\alpha^{\Pi \circ \pi}$. Finally, as long as unvisited stations are left, the minimum cost policy π^* provides the next station visit v^* for λ from the current state x (l.19) with λ being located at $v_0^\lambda := v^\lambda$. Since π^* maps to a search path $(v_0^\lambda, v_1^\lambda, ..., v_n^\lambda)$, v^* corresponds to the first unvisited station, i.e., $v^* := v_1^\lambda$.

In the decentralized decision-making settings with observation-sharing (DEC-O-dyn), we compute an agent's search path using the algorithm developed for the static planning setting DEC-O. However, we recompute the initially planned search path each time the agent visits an occupied station, using the latest observations shared by all agents

4.3. Observation validity

For the sake of completeness, we discuss how to relax station occupancy persistency in the following. To do so, we assume that availability probabilities of already visited stations can recover over time following a time-dependent exponential function (cf. Guillet et al. 2022) defined as

$$p_v^r(\Delta_v^i) = p_v(1 - e^{-(\frac{\mu_v}{p_v})(\Delta_v^i)}) \,, \tag{4.7}$$

with Δ_v^i being the elapsed time since i visited v for the last time. Here, $\frac{1}{\mu_v}$, denotes the average time station v remains occupied, and p_v the probability that v is available prior to any visit. Both values remain constant over the total planning horizon. To account for time-dependent recovering functions, we consider observed stations as candidate stations with an availability probability recovered according to Equation 4.7. We note that to reduce the computation overhead, one could chose to exclude latest observed stations from candidate stations, formally if $\Delta_v^i \leq T_{\text{thres}}$.

In this paper, we assume that agents can only passively communicate about occupied charging stations. These observations could be deduced from sharing GPS trace data, if a vehicle drives by a station without stopping, then the station is assumed to be occupied. However, future work could assume agents being able to communicate actively about station availability when they leave the station after having (re)charged their car. In this case, the station is available upon the agent's departure and the probability availability decreases from 1 to her initial value p_v, analogously to Equation 4.7 as

$$p_v^r(\Delta_v^i) = p_v + (1 - p_v)(e^{-(\frac{\mu_v}{p_v})(\Delta_v^i)}). \tag{4.8}$$

Moreover, Equation 4.8 could be used in a setting that allows an agent not to select the first (or next) available station, while reporting about the availability status of the non-selected stations.

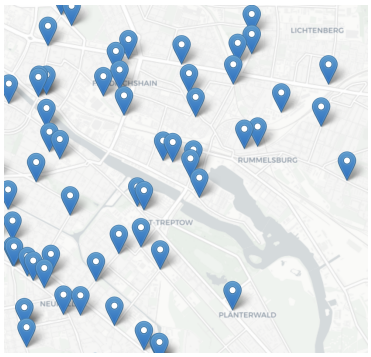
5. Experimental design

To analyze the effectiveness of the different coordination strategies, we conduct extensive numerical simulation experiments on real-world test instances for the city of Berlin (see Figure 4). In the following, we first detail our instance generation, before we describe additional benchmark algorithms, and elaborate on the metrics used to evaluate our algorithms' performance.

Instance generation: Besides the charging station availability distribution, the ratio of candidate stations per number of drivers and the departure time horizon are the main factors that impact our results. Accordingly, we vary the number of drivers, the global and individual search area dimensions, and the planning horizon to create a diverse set of scenarios as follows. To account for varying spatial overlaps in between multiple drivers' search areas as well as for a varying number of drivers, we randomly draw departure locations within a radius of $r^s \in \{100, 300, 700\}$ meters for a total number of $N \in \{2, ..., 10\}$ drivers. Additionally, we consider two different driver search radii of $\bar{S} = 1$ km and $\bar{S} = 2$ km. Table 2 specifies the available number of stations that result from varying those parameters. Moreover, we account for varying temporal overlaps between multiple drivers by equally distributing the drivers' search start time within a varying time horizon $t^s \in \{0, 1, 5, 15\}$ min. Independent of those characteristics, drivers have a search time budget of $\bar{T} = 5$ min. Utilizing a full-factorial design, we thus obtain 216 different test instances for our studies.

To analyze the impact of the varying charging station availability, we consider a low and a high charging station availability scenario. We generate those scenarios based on probability distributions centered on an expected mean d_a with $d_a = 0.25$ for the low-availability scenario (*low-25*) and $d_a = 0.60$ for the high-availability scenario (*high-60*). For each instance and availability scenario, we perform 100 simulation runs and use the same realized availability values to compute simulated estimates over all test instances.

Figure 4: **Charging stations distribution in a part of Berlin**



Charging station network in the city center of Berlin, Germany used to build the respective instance graph. The Image was created using Folium, which is data licensed under MIT License.

Table 2: **Number of stations depending on the search area dimension**

$\bar{S}$	r^s	N								
		2	3	4	5	6	7	8	9	10
1000	100	11	13	13	13	13	13	13	13	13
	300	11	13	14	15	16	17	18	18	18
	700	10	13	15	14	16	18	22	22	21
2000	100	22	23	23	23	23	23	23	23	23
	300	23	25	27	28	29	30	31	31	31
	700	26	31	36	40	40	35	52	53	40

We analyzed the sensitivity of our algorithms with respect to the penalty term β^G in a preliminary study (see Appendix D.1). In this study, we observed only little sensitivity of the results to β^G, such that there exists only a minor trade-off between system robustness and quality performances, that is best addressed by setting $\beta^G = 700$ min.

We further consider agents' heterogeneity to account for both heterogeneous and homogeneous use cases. We focus the main results discussion on the homogeneous agents use-case, because a service provider does not want to bias the search towards single agents by considering heterogeneous parameters in practice. Accordingly, we set the station's utilization cost $\gamma_v^i = 0$ for all drivers $i \in \mathcal{D}$ and all stations $v \in \mathcal{V}$, and consider an identical time budget $\bar{T}$ and search radius $\bar{S}$ for all agents. We additionally briefly analyze the impact of heterogeneous agents' time budget and search radius to complete our analyses.

Benchmark algorithms: We evaluate the quality performances of the RO and LH-RO in the centralized setting (CEN). Since the MDP is too large to be solved optimally, we benchmark both policies against a myopic greedy algorithm G and a deterministic offline solution OFF, as suggested in Powell (2009).

In G, we greedily decide on the next station visit v^* for the deciding agent λ in each decision epoch based on a cost combining the driving time from its current station v^λ to available stations and the individual driver's time-based penalty weighted by stations' availability probability. Formally,

$$v^* = \arg\min_{v \in \bar{\mathcal{V}}} t_{v^\lambda, v} + (1 - p_v)\bar{\beta} \,, \tag{5.1}$$

with $\bar{\mathcal{V}}$ being the set of candidate stations that agent λ can visit.

In OFF, we assume the charging demand, i.e., all drivers' departure time and location, to be known for the overall planning horizon. We then compute for each realization $k \in [0, ..., 100]$ of simulated stations availability, the minimum-cost assignment of drivers to stations, using a weighted bipartite graph $G' = (\mathcal{V}', \mathcal{A}')$, with $(v, i) \in \mathcal{A}'$ such that $i \in \mathcal{D}$ and $v \in \mathcal{V}_{\text{avail}}(k)$ is the set of available stations for realization k. Let the weight for arc (v, i) be the driving time from the agent i's start location to station v, such that $w_{v,i} = t_{v_0^i, v}$. We add dummy station vertices ν if $|\mathcal{V}_{\text{avail}}| \leq |\mathcal{D}|$, such that $w_{\nu,i} = \beta \; \forall i \in \mathcal{D}$ and we add dummy driver vertices ι if $|\mathcal{V}_{\text{avail}}| \geq |\mathcal{D}|$, such that $w_{v,\iota} = 0 \; \forall v \in \mathcal{V}_{\text{avail}}$. We then solve the resulting assignment problem with the Karp algorithm (Karp 1980).

Performance evaluation: For each test instance, we successively compute the search path (static planning) or next station visit (dynamic planning) for all drivers, according to their departure order and the selected setting. We then evaluate the performance from both a driver and a system perspective, based on 100 simulation runs for both *low-25* and *high-60* scenarios. From a driver perspective, we compute the realized search time $\hat{t}^k$ for each simulation run k, which allows us to compute the simulated estimate of a driver i's individual

cost $\hat{\alpha}^i$ as

$$\hat{\alpha}^i = \sum_{k=0}^{100} \hat{t}^k + \hat{\delta}^k \bar{\beta} / 100 \ , \tag{5.2}$$

with $\hat{\delta}^k$ being the binary variable that indicates whether the k^{th} search was successful. We obtain the driver's success rate $\hat{\rho}^i$ as

$$\hat{\rho}^i = \sum_{k=0}^{100} \hat{\delta}^k / 100 \ . \tag{5.3}$$

From a system perspective, we compute the simulated estimate of the expected system cost $\hat{\alpha}$ as

$$\hat{\alpha} = \sum_{i \in \mathcal{D}} \hat{\alpha}^i + (1 - \prod_{i \in \mathcal{D}} \hat{\rho}^i) \beta^G \ . \tag{5.4}$$

The quantity $\hat{\rho} = \prod_{i \in \mathcal{D}} \hat{\rho}^i$ describes the realized system success rate, i.e., the simulated estimate of the likelihood that all drivers successfully finished their search.

6. Results

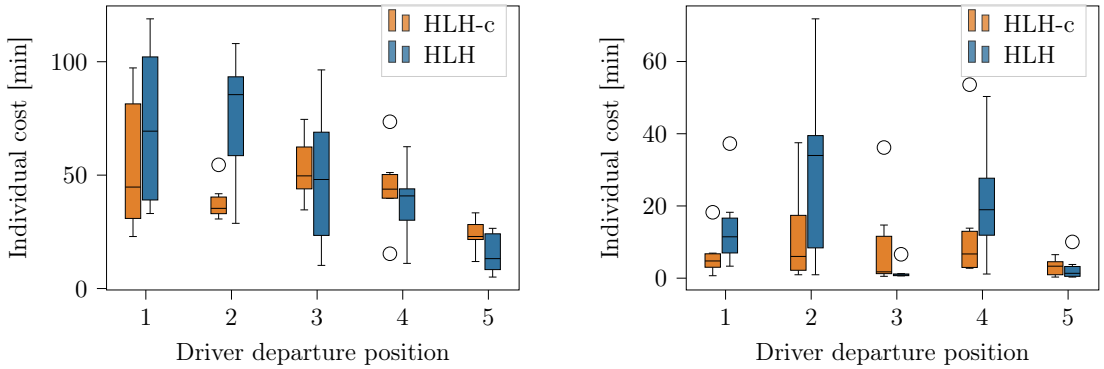
We first discuss our results from a system perspective (Section 6.1) before we focus on a driver perspective (Section 6.2).

6.1. System perspective

In the following, we first analyze the impact of collaboration in intention-sharing settings, before we focus on the impact of dynamic planning in observation-sharing settings. We then benchmark our algorithms for the centralized setting, and finally draw general conclusions on the performance of all possible algorithmic settings.

6.1.1. Collaboration in intention-sharing settings:

To analyze the benefit of collaboration, we compare the performances of HLH and HLH-c (see Section 4.1.3), in the DEC-I setting, with respect to individual drivers' costs $\hat{\alpha}$ (see Equation 5.2) and the solution's fairness. Here, we consider a solution to be fair if each driver obtains a similar cost α^i independent of her departure position. Figures 5a and 5b show the distribution of individual drivers' cost depending on their departure order, obtained with the collaborative algorithm (HLH-c), respectively the non-collaborative algorithm (HLH) for $t^s = 0$ min, in low- and high-availability scenarios for all instances with $N = 5$ drivers. We note that five simultaneously active drivers represent a very likely real-world scenario that we observed in practice. For each setting, Table 5c compares the difference between the highest and the lowest value (median, mean, 1$^{\text{st}}$ and 3$^{\text{rd}}$ quartiles, max, min) obtained for these individual drivers' cost distributions. As can be seen in Figure 5, solutions obtained with HLH-c outperform the solutions obtained with HLH on average with respect to both the system cost $\hat{\alpha}$ and the solution's fairness. Table 5c shows that the departure position has a smaller impact on the results with than without collaboration. The difference between



(a) *low-25* scenario (b) *high-60* scenario

		Δ(mean)	Δ(median)	Δ(max)	Δ(min)	Δ(q1)	Δ(q3)
low-25	HLH-c	26.7	31.2	63.9	22.7	22.3	53.2
	HLH	60.1	72.3	92.4	28.0	50.2	78.0
	Δ_{ref} [%]	-56	-57	-31	-19	-56	-32
high-60	HLH-c	11.3	4.95	47.1	2.44	2.07	12.8
	HLH	28.5	33.1	65.2	3.02	11.4	38.3
	Δ_{ref} [%]	-60	-85	-28	-19	-82	-66

(c) Comparison of the distributions of driver's costs per departure position

Figure 5: Comparison of HLH and HLH-c in the DEC-I setting for $t^s = 0$ min

Note. Each subplot shows for each driver i the distribution of the realized individual cost $\hat{\alpha}^i$ depending on her departure position, over all test instances that correspond to $t^s = 0$ min, $r^s \in \{100, 300, 700\}$ m, $\bar{S} \in \{1000, 2000\}$ meters, for $N = 5$ drivers. In the table, we compute Δ as follows for each metric $m \in \{\text{mean}, \text{median}, \text{q1}, \text{q3}, \text{max}, \text{min}\}$: $\Delta(m) = \max_i(m(i)) - \min_i(m(i))$, with i being the departure position and $m(i)$ being the statistic of the costs distribution corresponding to the i^{th} departure position. We compute Δ_{ref} as follows: ${}^{\Delta_{\text{HLH-c}}(m) - \Delta_{\text{HLH}}(m)}/_{\Delta_{\text{HLH}}(m)}$.

highest and lowest mean, median, maximum, minimum, 1st quartile, 3rd quartile values decreases with collaboration. In particular, collaboration decreases the difference for mean individual costs by 56% in the *low-25* scenario and by 60% in the *high-60* scenario. In the analyzed case, drivers start their search closely one after another, such that the last navigated drivers have a high chance to start while the preceding drivers are still searching. Without collaboration, the latest drivers may use the visit intention information to their advantage by earlier visiting the stations targeted by the preceding drivers. Here, the collaborative procedure HLH-c yields lower realized system cost $\hat{\alpha}$ than HLH as it ensures that the use of visit intention information benefits all drivers by avoiding conflicts at the stations already included in their search paths. An additional analysis in Appendix D, show a similar trend for $t^s = 1$ min, but decreasing effects for larger departure time horizon, $t^s = 5$ and $t^s = 15$ min. In the latter case, we observe no significant benefit in a collaborative setting.

Result 1. *Visit intentions sharing can – and should – be used collaboratively. Without collaboration, visit intention may favor selfish solutions that negatively affect early departing drivers' search paths.*

6.1.2. Dynamic planning for decentralized observation-sharing:

We analyze the impact of dynamically re-planning solutions in the observation-sharing setting.

Table 3 compares the system cost $\hat{\alpha}$ for DEC-O and DEC-O-dyn, in low- and high-availability scenarios. As can be seen, the effectiveness of dynamic and static planning depends on the length of the drivers' departure time horizon: if t^s is small, drivers start their search almost simultaneously and cannot benefit from prior observations of preceding drivers. Here, dynamically sharing observations during the search significantly increases the available information for all drivers and thus leads to significant improvements, by decreasing cost $\hat{\alpha}$ on average by 8% and up to 53%. If t^s is large, subsequently searching drivers benefit from observations shared by prior drivers already in a static planning approach. Accordingly, the benefit of dynamic observation-sharing decreases with an average cost decrease limited to 2%. Here, DEC-O even outperforms DEC-O-dyn in some cases (e.g., $N = 4$ & $t^s \in \{5, 15\}$).

Result 2. *Without intention-sharing, dynamic observation-sharing in addition to static observation sharing improves the system performances for short departure time horizons by decreasing cost $\hat{\alpha}$ by 8% on average.*

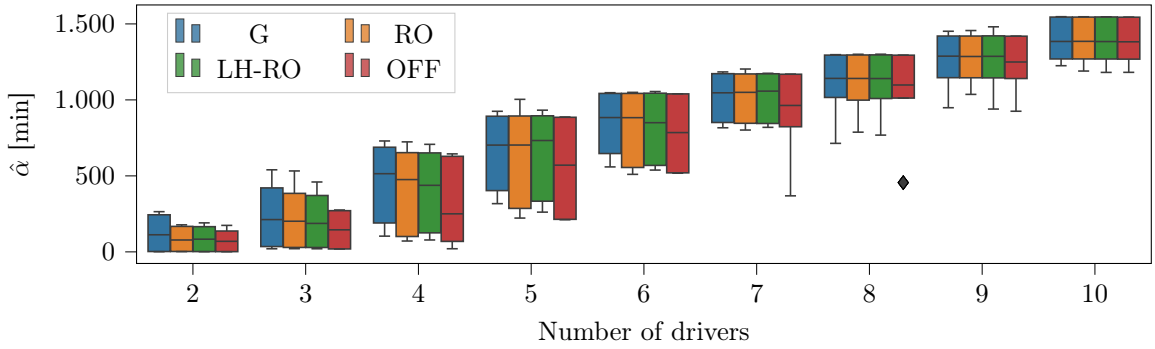
6.1.3. Centralized planning:

In the following, we compare the performance of RO's and LH-RO's policies to the greedy (G) and offline (OFF) benchmark described in Section 5. Here, G provides an upper bound to RO and LH-RO, which allows to analyze the performance gain by looking ahead (RO & LH-RO) rather than myopically deciding (G). Contrary, OFF provides a lower bound for each availability realization, as all uncertain information – station availability and future charging demand – is known, allowing to study an artificial perfect information setting. Figure 6 shows the detailed distribution of the realized cost $\hat{\alpha}$ for RO, LH-RO, G, and OFF,

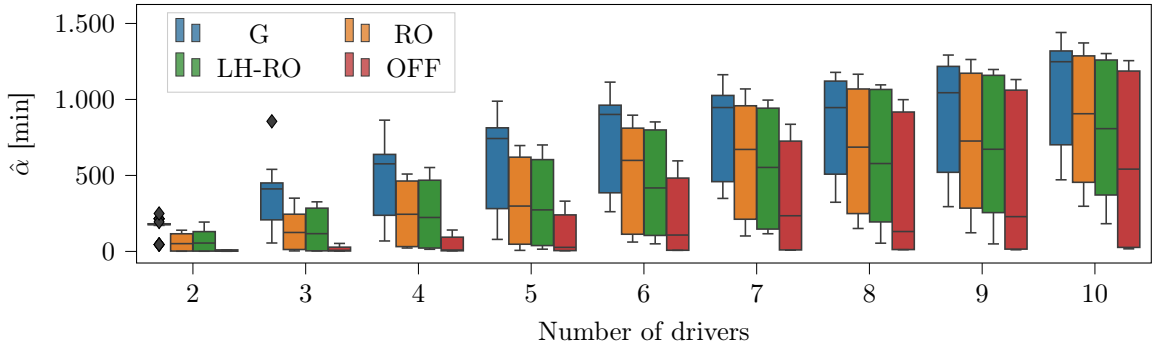
Table 3: System cost comparison between static and dynamic decentralized policy planning strategies

	low-25				*high-60*			
n	t^s=0	t^s=1	t^s=5	t^s=15	t^s=0	t^s=1	t^s=5	t^s=15
2	-13.4	-6.94	0.09	0.09	-7.14	-1.54	0.00	0.00
3	-27.7	-21.9	-7.67	-1.16	-53.1	-37.2	0.65	0.20
4	-21.7	-9.37	-4.42	5.30	-15.5	-34.6	33.1	27.7
5	-14.0	-11.3	-2.81	-0.75	10.4	-25.2	-12.7	7.62
6	-10.2	-4.52	-1.90	-0.58	-22.4	-3.84	-4.56	0.75
7	-6.14	-2.12	-1.47	0.77	0.86	3.87	-5.12	1.45
8	-7.96	-3.94	-2.07	-0.13	-6.32	-12.9	-5.01	-5.31
9	-6.80	-5.75	0.18	-0.10	-11.2	-11.1	-2.75	-4.72
10	-6.38	-3.78	-1.51	0.27	-5.21	-3.58	-3.56	-5.50

The table shows $\hat{\alpha}$'s relative improvement $\Delta[\%] = {}^{(\hat{\alpha}_{\text{dyn}} - \hat{\alpha}_{\text{stat}})}/_{\hat{\alpha}_{\text{stat}}}$ of the dynamic setting to the static setting for DEC-O. A negative Δ means the dynamic setting outperforms the static counterpart.



(a) $\bar{S} = 1000$ meters



(b) $\bar{S} = 2000$ meters

Figure 6: Distribution of the realized system cost $\hat{\alpha}$ in CEN for RO, LH-RO, G, and OFF per number of drivers, separated for small and large search areas

for a varying number of drivers and search radii $\bar{S}$.

As can be seen, both RO and LH-RO outperform the myopic policies (G) by decreasing cost $\hat{\alpha}$ by 14% (RO) and 16% (LH-RO) on average. The cost reduction depends however on the search radius $\bar{S}$. For small search areas ($\bar{S} = 1000$ m), both RO and LH-RO decrease the cost obtained with G by 2% on average, while the perfect-information setting OFF yields a 9% cost decrease on average. We observe that the benefits of using RO, LH-RO, or even OFF, decrease with the number of drivers as in this case, there is only little room for improvement over the myopic policy, due to the very limited number of candidate stations available for all drivers. For larger search areas ($\bar{S} = 2000$ m), all non-myopic settings achieve a significantly higher cost reduction of 29% (RO), 34% (LH-RO) and 63% (OFF) compared to G. According to LH-RO and RO's performances, we decide to use LH-RO in the CEN setting.

Result 3. *On average, the LH-RO algorithm decreases the cost $\hat{\alpha}$ obtained by RO by 3% and the cost obtained by G by 16% in a centralized setting. Specifically, for larger search areas ($\bar{S} = 2000$ m), LH-RO achieves a cost reduction of 7% (compared to RO) and 34% (compared to G).*

6.1.4. General performance evaluation:

We compare the performances of all decentralized settings, DEC, DEC-I, DEC-O, DEC-IO and DEC-O-dyn, the centralized setting CEN, and a decentralized myopic benchmark,

DEC(N), that greedily computes search paths in a decentralized way, to reflect drivers' behavior without any search assistance.

Figure 7 provides a detailed comparison of all settings with respect to the system cost $\hat{\alpha}$ averaged over all test instances. As can bee seen, an advanced search strategy without coordination (DEC) already decreases the system cost obtained with a naive search strategy (DEC(N)) by 15% on average. With coordination, the DEC-O setting decreases the cost compared to DEC(N) by 26% in a static setting and by 30% in a dynamic setting (DEC-O-dyn), while DEC-I and DEC-IO lead to a 37% decrease, and CEN yields a 39% cost decrease. Comparing uncoordinated and coordinated settings, we note that DEC-O-d decreases the cost obtained in DEC by 18% while both DEC-I and DEC-IO decrease it by 26%, and CEN leads to a 28% decrease. The static observation-sharing setting DEC-O performs on average worse than both static intention-sharing settings due to the algorithm's sensitivity to the departure time horizon t^s (see Section 6.1.2) and yields only a 13% cost decrease. Both DEC-I and DEC-IO show performances very close to CEN, which indicates that a decentralized and static setting performs nearly as well as a dynamic and centralized setting, as long as drivers share intentions. Figure 8 visualizes two static search paths computed in the DEC and DEC-I settings, for $N = 5$ drivers, $\bar{S} = 2000$ m, $r^s = 300$ m and $t^s = 0$ min: the benefit of coordination translates into routes with fewer overlapping stations.

Result 4. *Compared to an uncoordinated setting (DEC), decentralized static intention-sharing (DEC-I, DEC-IO) reduces $\hat{\alpha}$ on average by 26%. A centralized dynamic setting (CEN) leads to a slightly higher cost reduction of 28%.*

Figure 9 compares DEC, DEC-I, DEC-O, DEC-IO, DEC-O-dyn and CEN with respect to cost $\hat{\alpha}$ in low-availability scenarios. We divide plots between short departure time horizons ($t^s \in \{0, 1\}$ min) and large departure time horizons ($t^s \in \{5, 15\}$ min), as well as between small search areas ($\bar{S} = 1000$ m) and large search areas ($\bar{S} = 2000$ m).

Numerical results show that a larger search area ($\bar{S} = 2000$ m) decreases the system cost $\hat{\alpha}$ for coordinated settings by at least 30% in *low-25* on average and by at least 90% in *high-60*

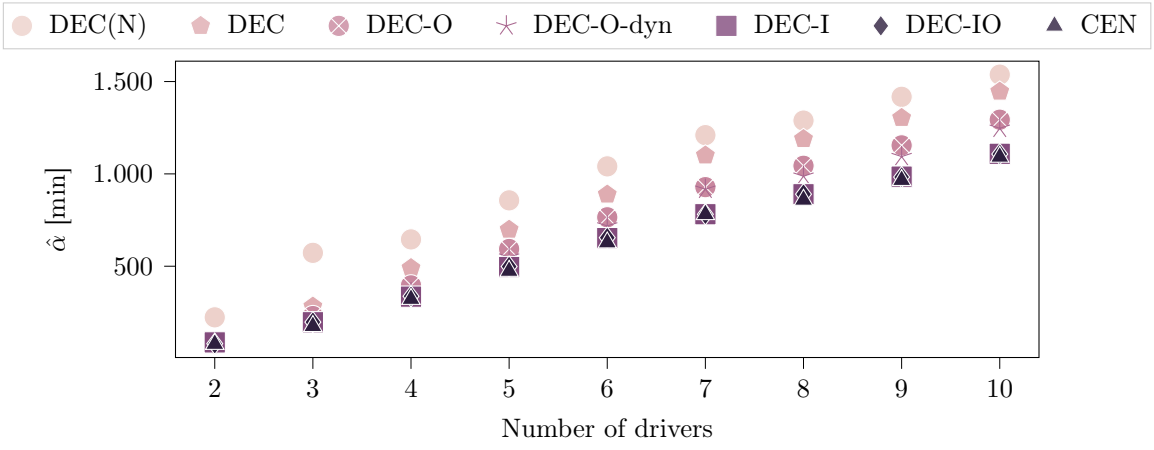
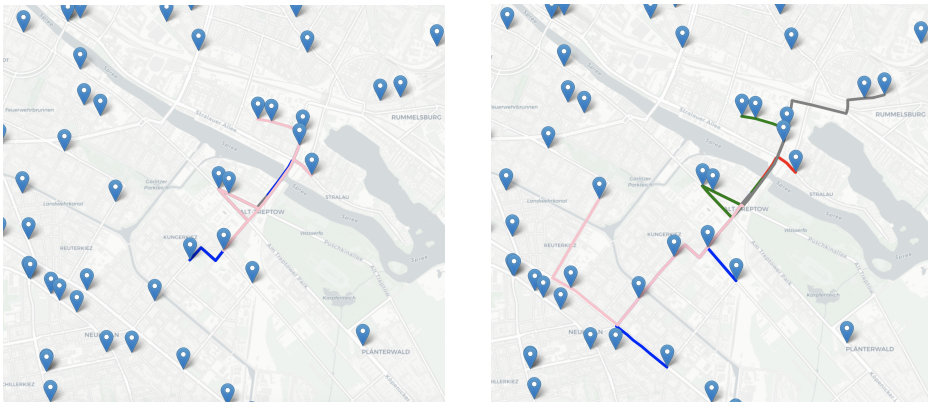


Figure 7: Average realized system cost $\hat{\alpha}$ for all information-sharing settings and both selfish settings, aggregated over all instances, per number of drivers



(a) DEC (b) DEC-I

Figure 8: Visual representation of uncoordinated (DEC) and coordinated (DEC-I) search routes

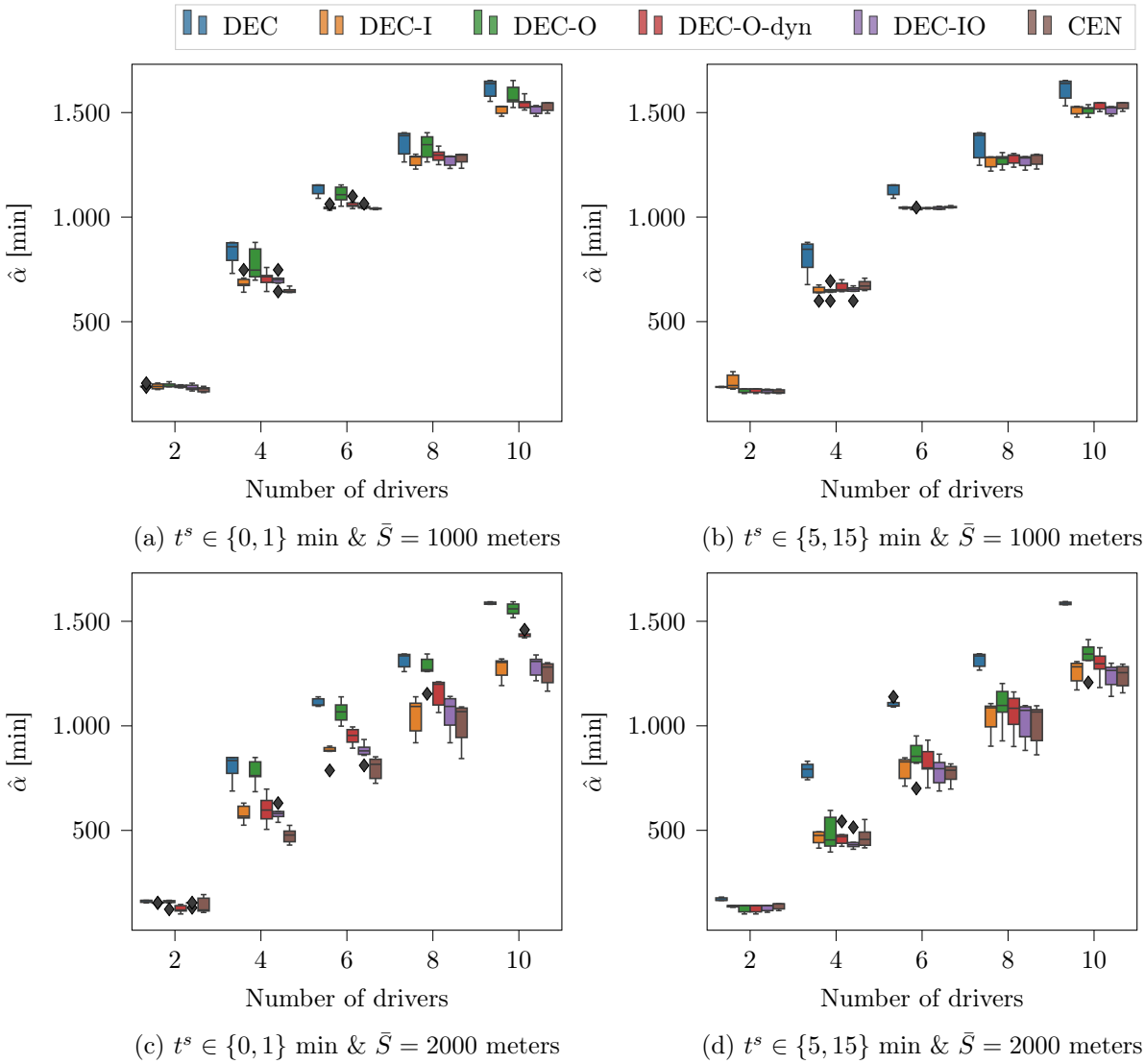
scenarios on average. In contrast, $\hat{\alpha}$ remains constant, or in some cases increases without coordination.

For short departure time horizons (see Figures 9a&9c), we observe that pure intention-sharing (DEC-I) slightly outperforms combined observation- and intention-sharing (DEC-IO) for static policies, mostly when $\bar{S} = 2000$ m. In this case, DEC-IO tends to provide lower individual cost solutions for the early departing drivers than DEC-I due to the additional observation information. Here, improving the solutions of early drivers worsens the solution quality of subsequent drivers due to the limited number of possible station visits. In contrast, DEC-I achieves better performances from a system perspective by negatively affecting the early drivers' search, such that subsequent drivers can obtain higher-quality solutions. In observation-sharing settings, results confirm that dynamic policies (DEC-O-d) outperform static policies (DEC-O).

For large departure time horizons (see Figures 9b&9d), DEC-IO improves over DEC-I and even outperforms CEN for smaller search areas. Notably, DEC-O performs very similar to the other coordinated settings in this case. As the departure time horizon increases, the likelihood that searches temporally overlap decreases, which in turn decreases the benefits of intention-sharing in addition to observation-sharing. We further observe that DEC-O slightly outperforms DEC-O-dyn in this case for small search areas ($\bar{S} = 1000$ m). We note that in this case, DEC-O is well suited for a practical implementation as it has lower computational requirements than DEC-IO or CEN.

Figure 23 in Appendix D contains similar analyses for high-availability scenarios. While these results show similar trends in general, we note that the benefit of dynamic observation-sharing in a decentralized setting (DEC-O-dyn) is less consistent in this scenario for smaller departure time horizons.

Result 5. *Sharing occupancy observations in addition to intentions is not beneficial for almost simultaneous searches, i.e., $t^s \in \{0, 1\}$ min. In this case, observation-sharing improves the early departing drivers' solution to the detriment of succeeding drivers, which worsens*



(a) $t^s \in \{0, 1\}$ min & $\bar{S} = 1000$ meters

(b) $t^s \in \{5, 15\}$ min & $\bar{S} = 1000$ meters

(c) $t^s \in \{0, 1\}$ min & $\bar{S} = 2000$ meters

(d) $t^s \in \{5, 15\}$ min & $\bar{S} = 2000$ meters

Figure 9: Comparison of decentralized and centralized decision-making in low-availability scenarios

the total system performance.

Result 6. *For larger departure time horizons, i.e., $t^s \in \{5, 15\}$ min, the decentralized observation-sharing setting (DEC-O) performs similar as the decentralized information-sharing settings (DEC-I, DEC-IO) and as the centralized setting (CEN).*

6.2. Single-driver perspective

In the following, we evaluate how coordination impacts an individual driver's solution, before we analyze the impact of driver heterogeneity.

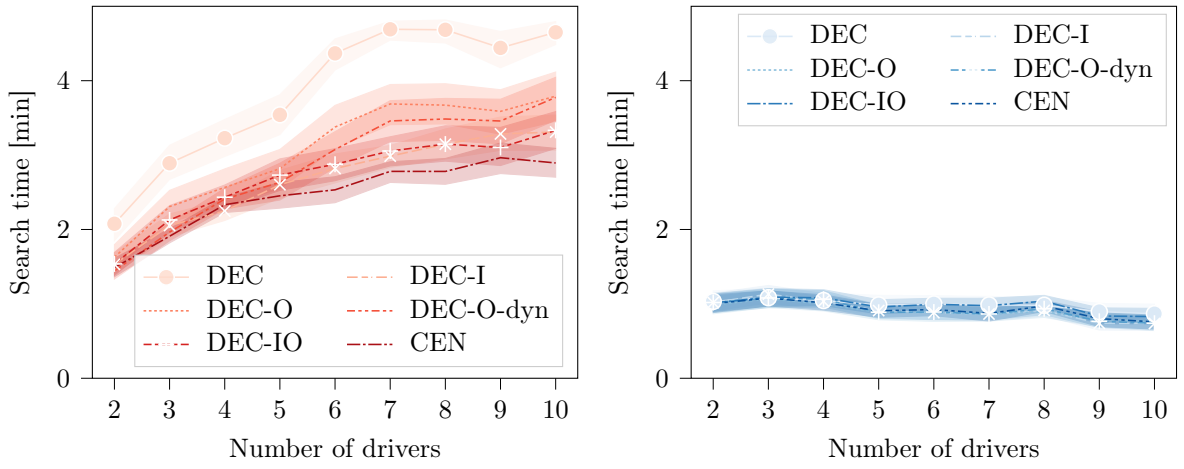
6.2.1. Drivers' benefits of coordination:

We refer to the uncoordinated solutions (DEC) as selfish solutions, in which a driver obtains her solution independently, without additional information and in her own best interest.

First, we analyze the worst and best realized solutions obtained among all drivers for each test instance. For all decentralized settings (DEC-I, DEC-O, DEC-IO, DEC-O-dyn) as well as for the centralized dynamic setting (CEN), we compare the results to the uncoordinated setting (DEC). We then analyze the impact of a driver's departure position on her individual solution, before we analyze each driver's success rate and search time deviation between the selfish and any coordinated solution.

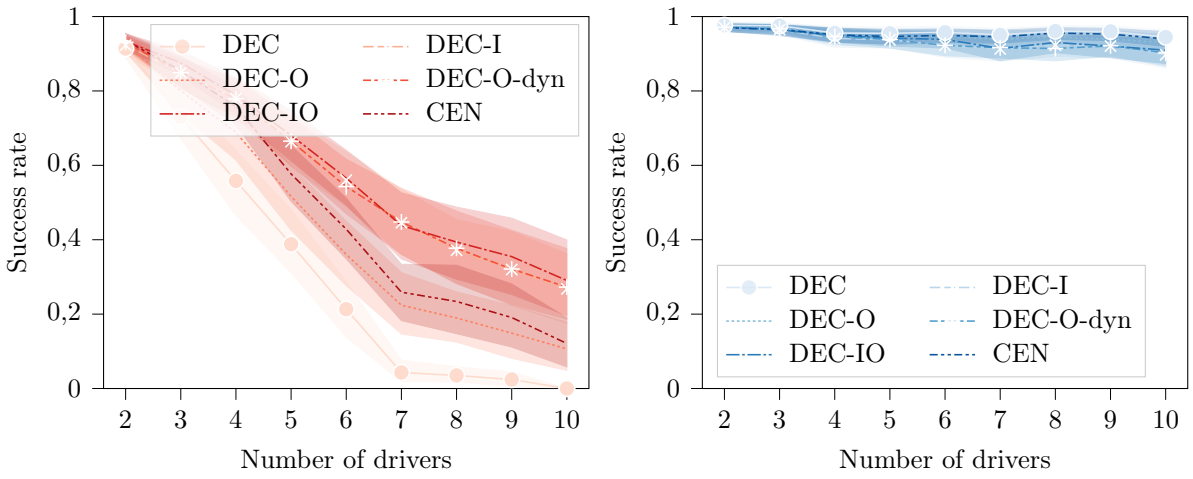
Figure 10 shows the mean value of all lowest and highest realized individual search times and a corridor corresponding to a 95% confidence interval over all test instances of the low-availability scenario for each analyzed setting. Figure 11 analogously shows mean values of all lowest and highest realized individual success rates. As can be seen, the lowest search times are comparable with and without coordination, such that a selfish solution does not improve the best case scenario with respect to individual search times.

However, all coordinated settings decrease the maximal individual search times, in particular by 30 % in DEC-I and DEC-IO, and by 35% in CEN. Analyzing performances with



(a) Worst individual search times (b) Best individual search times

Figure 10: Distribution of the worst and best search times in the low-availability scenario



(a) Worst individual success rates (b) Best individual success rates

Figure 11: Distribution of the worst and best success rate in the low-availability scenario

respect to individual success rates (see Figure 11), there exists a minor trade-off between the best and the lowest individual success rates. While the highest individual success rate appears to be slightly higher in a selfish environment, the lowest success rate increases significantly in all coordinated settings, by up to 30 % (0.02 to 0.32). We note that CEN yields the most robust solutions, by providing lowest worst individual search times and highest worst individual success rates.

Result 7. *Coordinated searches outperform selfish searches, because they significantly improve the worst-possible solution that a driver may obtain while preserving her best-possible solution.*

Figure 12 shows the individual costs $\hat{\alpha}^i$ obtained for all drivers i depending on their departure position, averaged over all values of $r^s \in \{100, 300, 700\}$ for test instances with ten drivers. Figure 12a shows results for a short departure time horizon ($t^s = 1$ min), while Figure 12b details results for a larger departure time horizon ($t^s = 15$ min).

As can be seen, coordination reduces a driver's search cost (nearly) independent of her departure position. In some specific cases (e.g., for the first driver with $t^s = 15$ min), a driver may obtain a higher individual cost with coordination than without. However, these larger costs occur at the benefit of a smaller spread between the worst and the best solution that any driver may obtain. Moreover, our results show that individual solutions are more homogeneous for shorter departure time horizons, as searches take place almost simultaneously. With a larger departure time horizon, early drivers are privileged against succeeding drivers as they have more chances to find a free station before parallel competing searches start.

Result 8. *At the exception of early departing drivers, the individual solution obtained by a driver with coordination outperforms the one obtained without coordination, independent of the driver's departure position.*

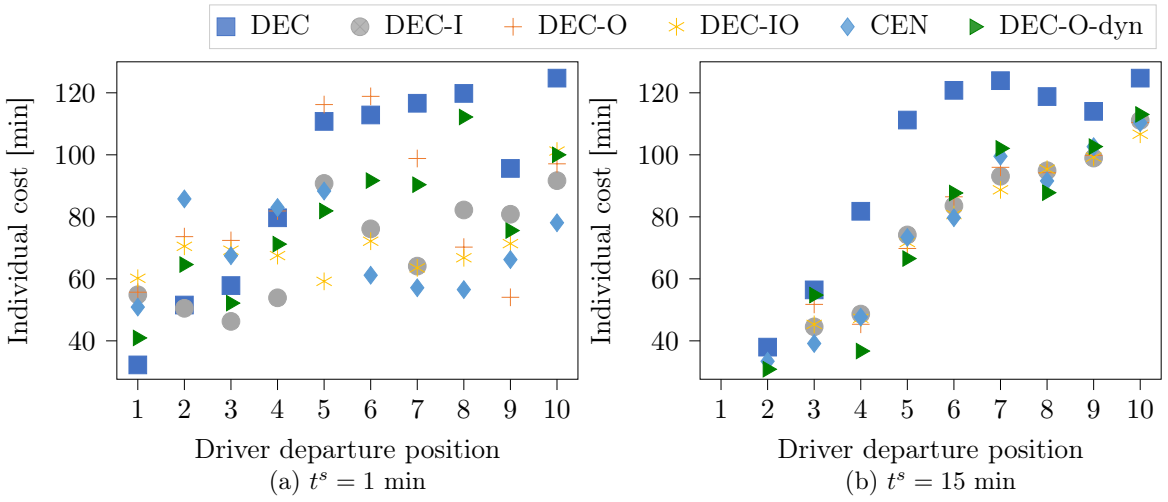


Figure 12: Single-agent cost ordered by departure times in a low-availability scenario (*low-25*) with larger search radius ($\bar{S} = 2000$ m, $N = 10$ drivers)

Table 4 shows the relative individual search time deviation ($\hat{\Delta}t_{rel}$) and the absolute individual success rate deviation ($\hat{\Delta}\rho$) per number of drivers N and search radius $\bar{S}$ for all instances, averaged over all r^s, t^s, and availability values. As can be seen, all coordinated settings allow a driver to reduce her search time and increase her search reliability on average. Specifically, a driver may save 2% (DEC-O-d), 3% (CEN), 8% (DEC-O, DEC-IO) and 9% (DEC-I) of her search time, while she can increase her success rate by 0.05 (DEC-O), 0.06 (DEC-O-d) and 0.09 (DEC-I, DEC-IO, CEN). In line with the system-perspective evaluation, DEC-I, DEC-IO, and CEN yield the best performances from a driver-perspective too, such that we further detail results only for these settings.

There exists a trade-off between search time savings and the reliability improvement. For small search areas ($\bar{S}$=1000 m), drivers save 8% (CEN) or 12% (DEC-I, DEC-IO) of their search times on average, and up to 23% (DEC-I, DEC-IO) with a high number of drivers ($N = 9$). The success rate increase is limited to 0.05 on average (DEC-I, DEC-IO, CEN). For larger search areas ($\bar{S} = 2000$ m), the time gain decreases. Drivers may only save 5% of their search times on average (DEC-I, DEC-IO) or even slightly increase their search time by 2% (CEN). In this case, however, the reliability gain significantly increases, with drivers increasing their success rates on average by 0.12 (DEC-I, DEC-IO) or 0.13 (CEN), and up to 0.30 (CEN with $N = 10$).

Table 4: Individual search time and reliability improvements with coordination

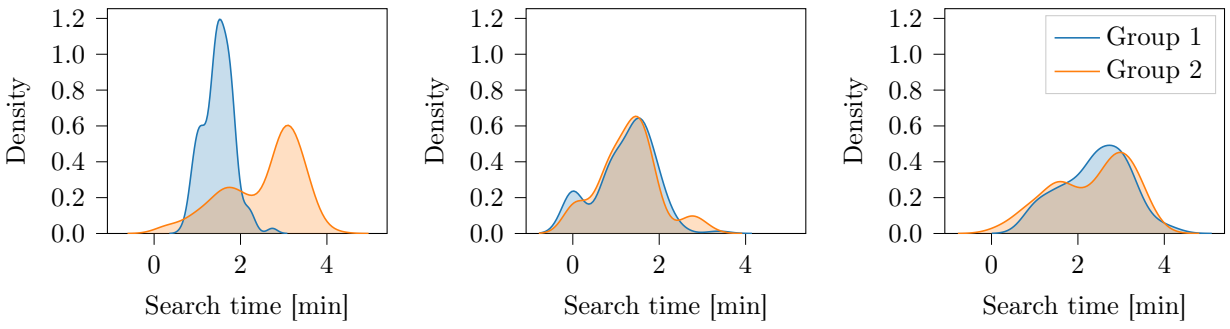
		Av. search time deviation $\hat{\Delta}t_{rel}$					Av. success rate deviation $\hat{\Delta}\rho$				
	N	DEC-I	DEC-O	DEC-O-d	DEC-IO	CEN	DEC-I	DEC-O	DEC-O-d	DEC-IO	CEN
$\bar{S} = 1000$ m	2	2.21	2.06	2.20	2.12	1.71	0.00	0.00	0.00	0.00	0.00
	3	1.40	2.39	-2.33	2.21	-2.33	0.01	0.01	0.01	0.01	0.01
	4	3.72	4.52	1.21	2.98	0.08	0.03	0.02	0.02	0.03	0.03
	5	5.86	7.27	4.52	4.49	2.23	0.05	0.03	0.04	0.05	0.05
	6	16.8	10.5	9.13	15.5	12.0	0.07	0.04	0.06	0.07	0.08
	7	18.0	13.6	7.26	15.7	9.88	0.09	0.05	0.06	0.09	0.08
	8	21.3	14.5	9.85	21.3	15.4	0.08	0.04	0.05	0.08	0.08
	9	23.4	18.0	11.8	23.4	14.9	0.07	0.04	0.06	0.07	0.08
	10	21.1	17.7	7.32	21.4	20.5	0.09	0.05	0.08	0.09	0.09
$\bar{S} = 2000$ m	2	1.99	2.11	2.39	2.11	2.50	0.00	0.00	0.00	0.00	0.00
	3	3.79	3.02	-1.83	4.00	-1.73	0.01	0.01	0.02	0.02	0.02
	4	3.50	4.07	0.35	2.82	-0.34	0.03	0.02	0.03	0.03	0.04
	5	1.97	4.87	-3.50	-0.40	-3.37	0.06	0.03	0.05	0.06	0.07
	6	7.68	7.92	0.15	6.77	1.01	0.10	0.05	0.08	0.10	0.11
	7	14.9	11.0	4.17	13.3	8.70	0.16	0.09	0.10	0.16	0.17
	8	1.36	4.61	-9.01	0.03	-9.16	0.19	0.09	0.13	0.19	0.21
	9	2.28	6.54	-6.56	0.51	-13.3	0.23	0.12	0.16	0.23	0.25
	10	9.38	7.43	-2.38	7.84	-3.01	0.29	0.13	0.18	0.29	0.30

The table compares Δt_{rel} and $\Delta\rho$, with $\hat{\Delta}t_{\text{rel}}$ [%] and $\hat{\Delta}\rho$ their respective average over all test instances, computed as: $\Delta t_{\text{rel}} = -1/n(\sum_{i=0}^{n}(\hat{t}^i_{\text{setting}} - \hat{t}^i_{\text{DEC}})/\hat{t}^i_{\text{DEC}})$ and $\Delta\rho = -1/n(\sum_{i=0}^{n}\hat{\rho}^i_{\text{DEC}} - \hat{\rho}^i_{\text{setting}})$, with n being the number of drivers. The evaluated setting outperforms DEC for positive values.

6.2.2. Impact of driver heterogeneity:

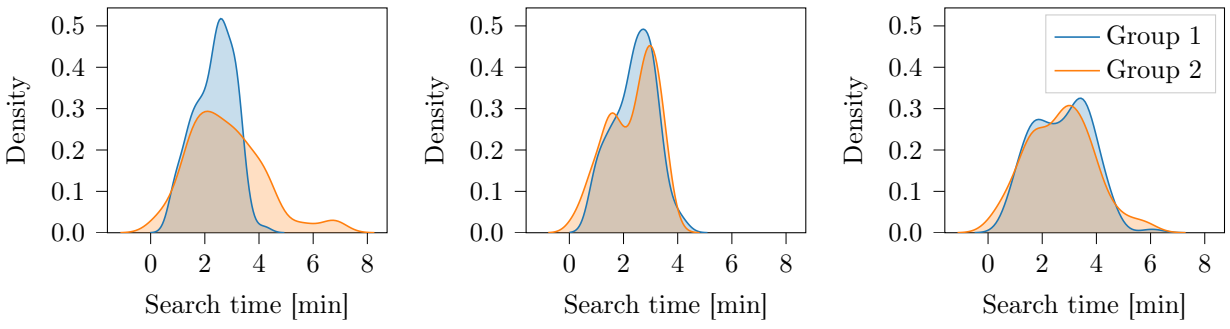
To analyze the impact of driver heterogeneity, we split drivers into two distinct groups: the first group contains all drivers with an odd departure position, while the second group contains all drivers with an even departure position. Figure 13 shows the impact of drivers with heterogeneous search radii in the CEN setting by analyzing three cases: in the first case (13a), drivers of the first group have a smaller search radius ($\bar{S} = 1000$ m), while drivers of the second group have a larger search radius ($\bar{S} = 2000$ m). In the second case (13b), drivers of both groups have a smaller search radius ($\bar{S} = 1000$ m) while in the third case (13c), drivers of both groups have a larger search radius ($\bar{S} = 2000$ m). Accordingly, drivers are heterogeneous in Figure 13a and homogeneous in Figures 13b&13c. Analogously, Figure 14 shows the impact of smaller and larger time budgets. In the first case (14a), drivers from the first group have a smaller time budget ($\bar{T} = 5$ min) while drivers from the second group have a larger time budget ($\bar{T} = 10$ min), in the second case (14b) all drivers have a smaller time budget whereas in the third case (14c), all drivers have a larger time budget. We average results for test instances with $N = 10$ drivers.

As can be seen, the distribution of individual search times in homogeneous settings is (nearly) independent of the respective group. This is not the case in heterogeneous settings. Here, drivers that perform a more constrained search, i.e., have a smaller search radius (see Figure 13a) or a smaller time budget (see Figure 14a), obtain lower search times compared to



(a) Heterog. (b) Homog. ($\bar{S} = 1000$ m) (c) Homog. ($\bar{S} = 2000$ m)

Figure 13: Impact of heterogeneous search radii on drivers' search times



(a) Heterog. (b) Homog. ($\bar{T} = 5$ min) (c) Homog. ($\bar{T} = 10$ min)

Figure 14: Impact of heterogeneous time budgets on drivers' search times

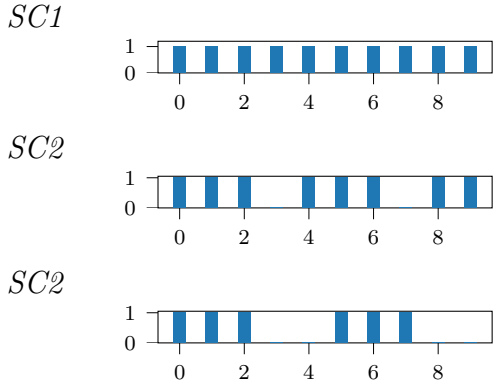
drivers that perform a less constrained search, especially in the first case. To reduce potential visit overlap, drivers with a larger search radius increase their chances of finding unoccupied stations by visiting far-distanced stations that are less affected by potential overlaps, which then contributes to increase their search times. Finally, we note that performing searches with homogeneous parameters appears to be preferable from a practitioner's perspective, in order to provide fair and consistent service to all customers.

Result 9. *Time budget or search radius heterogeneity favors drivers with lower time budget and lower search radius.*

6.2.3. Coordination in longer planning horizons:

In the following, we analyze the impact of coordination in a longer planning horizon, i.e., a three-hour departure time horizon, by comparing three charging demand distribution scenarios, see Figure 15. In *SC1*, drivers are steadily entering the system, with 5 requests per 15 minutes time interval. In *SC2*, each fourth request is skipped, whereas in *SC3*, every fourth and fifth requests are skipped. The first scenario reflects a homogeneous demand distribution, while the other scenarios reflect heterogeneous demand distributions, similar to peak demand that arises in practice. Table 5 compares the relative cost improvement obtained on average between coordinated (CEN) and uncoordinated decentralized (DEC) planning in both short ($t^s = 15$ minutes) and long ($t^s = 180$ minutes) planning horizon settings, depending on the radius of the area in which all agents start their search ($r^s \in \{100, 300, 700\}$ meters). Figure 16 further compares the average individual drivers cost obtained in the long planning horizon setting with coordination (CEN) and without (DEC) depending on the driver's departure times, the search radius ($\bar{S} \in \{1000, 2000\}$ meters) and the distribution scenario (*SC1* or *SC3*).

Figure 15: Requests distribution scenario



The figure shows for each scenario the requests distribution pattern, with each (missing) bar corresponding to a (missing) request.

Table 5: Relative cost improvement for CEN over DEC

Horizon	r^s	*SC1*	*SC2*	*SC3*
$t^s = 15$ min	100 m	-38.3%	-41.3%	-40.3%
	300 m	-38.3%	-42.6%	-49.0%
	700 m	-17.2%	-11.8%	-25.7%
$t^s = 180$ min	100 m	-21.7%	-26.5%	-29.6%
	300 m	-22.6%	-26.2%	-29.7%
	700 m	-23.6%	-25.8%	-27.7%

The table shows the relative improvement of the realized system cost $\hat{\alpha}$, computed as follows: $\Delta[\%] = (\hat{\alpha}_{\text{CEN}} - \hat{\alpha}_{\text{DEC}})/\hat{\alpha}_{\text{DEC}}$ of the CEN setting to the DEC setting in percentages, averaged over all instances corresponding to $\bar{S} \in \{1000, 2000\}$ m, and both *low-25* and *high-60* scenarios, for each value of r^s.

As can be seen, coordination significantly improves the overall search performances in longer horizon settings. Comparing the results of the short and long planning horizon settings (cf. Table 5), our results show that cost savings in the long planning horizon setting decrease for a limited departure area ($r^s \in \{100, 300\}$ meters), but increase when drivers are initially better distributed over the search area (i.e., $r^s = 700$ meters). When accounting for a long planning horizon, the overall station availability in the system decreases slightly due to an increasing number of drivers entering the system and possibly blocking charging stations for a longer period of time. However, additional information related to stations getting freed can be shared, such that the results illustrate the trade-off that exists between the performance loss due to the availability decrease and the performance gain due to the information increase. Our results further show that a decreased charging demand (i.e., *SC3*) increases the cost reduction obtained with coordination in long planning horizons, with an up to 46% $\hat{\alpha}$ cost decrease in the *high-60* scenario. While the system performances increase for a larger search radius, in-line with short planning horizons results, Figure 16 shows that decreasing the requests frequency has however a larger positive impact for a smaller search radius.

Finally, we observe that long planning horizon results reveal cyclic patterns for individual drivers cost (see Figure 16). These patterns show a decreasing amplitude for larger charging demand heterogeneity or search radii. This effect results from early drivers having a higher chance of reaching closely related charging stations and thus obtaining lower individ-

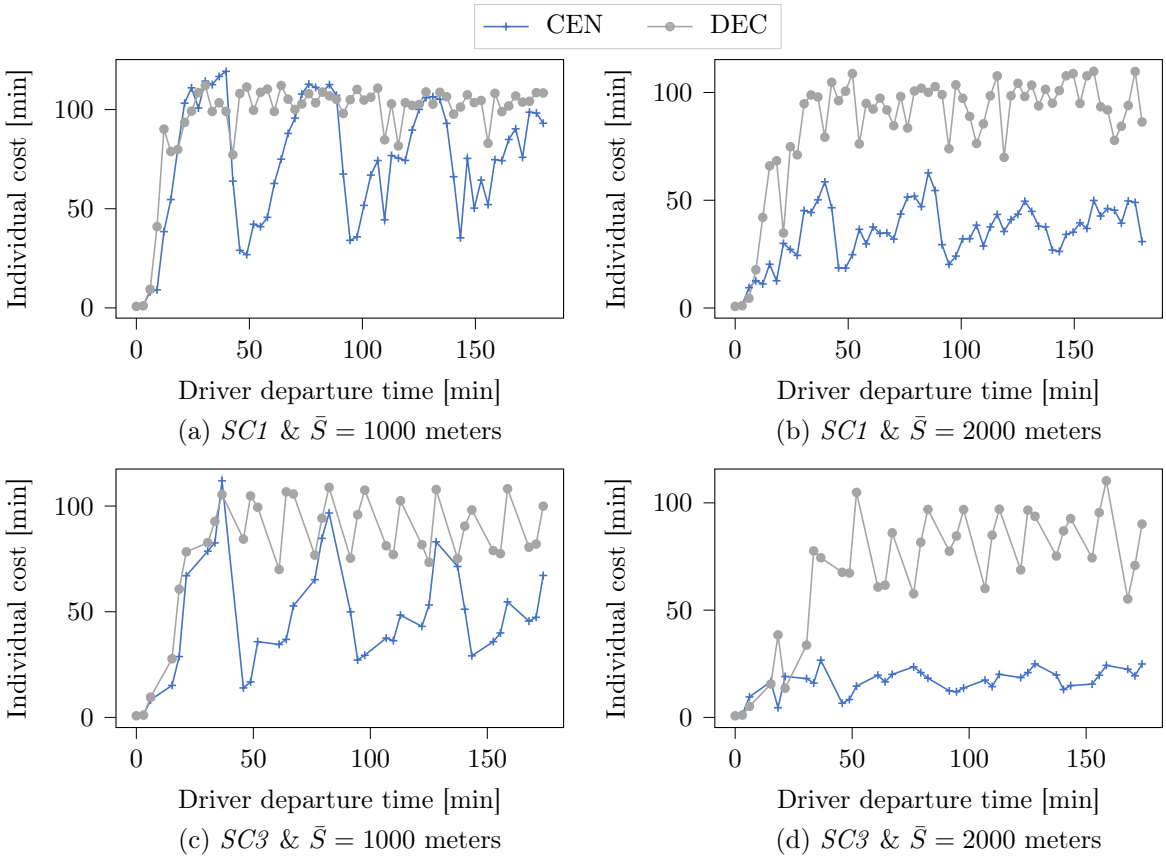


Figure 16: Comparison of DEC and CEN in high-availability scenarios for a three-hours planning horizon

ual cost than succeeding drivers. When these stations are freed after $\Delta T^{v}_{\text{charge}}$ minutes, they get revisited first with a higher recovered probability to be available to succeeding drivers, which replicates the pattern. With a smaller overlap between drivers' searches, which can be realized either by larger search radii or larger temporal disconnect between drivers entering the system, the amplitude of these patterns decreases as a smaller overlap increases the chance for each driver to reach an unoccupied closely related station. Figure 24 in Appendix D.4 shows complementary analyses for the low-availability scenario. Similar to short planning horizon results, the cost reduction obtained in CEN compared to DEC decreases in low-availability scenarios, especially for a small search radius with $\bar{S} = 1000$ meters.

7. Conclusion

In this paper, we studied the multi-agent charging station search problem in stochastic environments, which we define as a single-decision maker MDP. We showed that by constraining agents' individual policies to be executed independent of each other, we can simplify the global MDP representation to a set of single-agent MDPs. We then introduced several online algorithms that solve centralized and decentralized decision-making settings, applicable to static and dynamic policies, and different levels of information-sharing. Specifically, we analyzed the benefits of intention-sharing, i.e., drivers sharing their planned visits, observation-sharing, i.e., drivers sharing observed occupancies of charging stations, or both.

Using a case-study for the city of Berlin, we analyzed the benefits of coordination between multiple agents' search: our results show that coordination increases the system performance while individually benefiting each driver in general. Analyzing the performance from a system perspective, our results show that a static decentralized coordination strategy achieves a 26% cost decrease as long as drivers share visit intentions. A centralized coordination strategy requires higher computational load but achieves only 2% additional cost decrease. We further highlight the benefit of enforcing collaboration in intention-sharing settings. Moreover, our results show that observation-sharing performs on average worse than intention-sharing setting. However, observation-sharing requires less data and computational resources, and may be used to derive more accurate availability probabilities, which makes it interesting for practitioners. We show that from a driver perspective coordination may save up to 23% of a driver's search time, while increasing her search's success rate by 9% on average. We find that coordinated searches outperform uncoordinated searches for individual best and worst case scenarios. Finally, we observe similar effects in longer planning horizon settings.

Finally, one comment on our study is in order. We assumed a non-adversarial setting such that agents always follow their navigation device visit recommendations, which may however be challenged by drivers behavior in practice. If drivers deviate from the recommended visits, intention-sharing might become misleading. Analyzing competitive searches within a game-

theoretical setting by relaxing the non-adversarial assumption opens a new avenue for further research in this context.

Acknowledgments

This work was partially funded by the German Federal Ministry for Economic affairs and Energy within project iMove (01ME16003B).

References

Al-Kanj, L., J. Nascimento, W. B. Powell. 2020. Approximate dynamic programming for planning a ride-hailing system using autonomous fleets of electric vehicles. *European Journal of Operational Research* **284**(3) 1088 – 1106.

Arndt, T., D. Hafner, T. Kellermeier, S. Krogmann, A. Razmjou, M. S. Krejca, R. Rothenberger, T. Friedrich. 2016. Probabilistic routing for on-street parking search. *24th Annual European Symposium on Algorithms (ESA 2016)*, vol. 57. 6:1–6:13.

Bourgault, F., T. Furukawa, H. F. Durrant-Whyte. 2003. Coordinated decentralized search for a lost target in a bayesian world. *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*.

Chung, T. H., J. W. Burdick. 2008. Multi-agent probabilistic search in a sequential decision-theoretic framework. *2008 IEEE International Conference on Robotics and Automation*.

Dai, W., G. Sartoretti. 2020. Multi-agent search based on distributed deep reinforcement learning. Tech. rep., National University of Singapore.

Deloitte. 2020. Electric vehicles, setting a course for 2030. https://www2.deloitte.com/uk/en/insights/focus/future-of-mobility/electric-vehicle-trends-2030.html. Last accessed: 02.25.2021.

Goodson, J. C., B. W. T Thomas, J. W. Ohlmann. 2017. A rollout algorithm framework for heuristic solutions to finite-horizon stochastic dynamic programs. *European Journal of Operational Research* **258**(1) 216–229.

Guillet, M., G. Hiermann, A. Kröller, M. Schiffer. 2022. Electric vehicle charging station search in stochastic environments. *Transportation Science* **56**(2) 483–500.

Guo, Q., O. Wolfson. 2018. Probabilistic spatio-temporal resource search. *GeoInformatica* **22**(1) 75–103.

Jafari, E., S. D. Boyles. 2017. Multicriteria stochastic shortest path problem for electric vehicles. *Networks and Spatial Economics* **17**(3) 1043–1070.

Karp, R. M. 1980. An algorithm to solve the m x n assignment problem in expected time o(mn log n). *Networks* **10** 143–152.

Kullman, N., M. Cousineau, J. Goodson, J. Mendoza. 2022. Dynamic ridehailing with electric vehicles. *Transportation Science* **56**(3) 775–794.

Kullman, N., J. C. Goodson, J. E. Mendoza. 2021. Electric vehicle routing with public charging stations. *Transportation Science* **55**(3) 637–659.

Lauinger, D., F. Vuille, D. Kuhn. 2017. A review of the state of research on vehicle-to-grid (v2g): Progress and barriers to deployment. *Working paper, Ecole Polytechnique Fédérale de Lausanne* .

McKinsey. 2020. The road ahead for e-mobility. https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/the-road-ahead-for-e-mobility. Last accessed: 02.25.2021.

Myersdorf, D. 2020. Ultra-fast charging batteries and the cure for ev charge anxiety. https://www.intelligenttransport.com/transport-articles/112928/charging-anxiety/. Last accessed: 02.25.2021.

Powell, W. B. 2009. What you should know about approximate dynamic programming. *Naval Research Logistics (NRL)* **56**(3) 239–249.

Qin, K. K., W. Shao, Y. Ren, J. Chan, F. D. Salim. 2020. Solving multiple travelling officers problem with population-based optimization algorithms. *Neural Computing and Applications* **32**(16) 12033–12059.

Schmoll, S., M. Schubert. 2018. Dynamic resource routing using real-time dynamic programming. *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. 4822–4828.

Schuller, A., C. Stuart. 2018. "from cradle to grave: e-mobility and the energy transition. https://www.volkswagen-newsroom.com/en/stories/combatting-charging-anxiety-5107. Last accessed: 02.25.2021.

Sweda, T. M., I. S. Dolinskaya, D. Klabjan. 2017. Adaptive routing and recharging policies for electric vehicles. *Transportation Science* **51**(4) 1326–1348.

Tang, H., M. Kerber, Q. Huang, L. Guibas. 2013. Locating lucrative passengers for taxicab drivers. *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*.

Ulmer, M., J. C. Goodson, D. Mattfeld, Hennig M. 2019. Offline–online approximate dynamic programming for dynamic vehicle routing with stochastic requests. *Transportation Science* **53**(1) 185–202.

Volkswagen. 2019. Combatting "charging anxiety". https://www.volkswagen-newsroom.com/en/stories/combatting-charging-anxiety-5107. Last accessed: 02.25.2021.

Wong, E., F. Bourgault, T. Furukawa. 2005. Multi-vehicle bayesian search for multiple lost targets. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*.

You, P., Z. Yang, M. Chow, Y. Sun. 2016. Optimal cooperative charging strategy for a smart charging station of electric vehicles. *IEEE Transactions on Power Systems* **31**(4) 2946–2956.

A. Problem notation

Table 6: Notation

Notation used to define the MSCPS problem	
$\mathcal{G} = (\mathcal{V}, \mathcal{A})$	Complete charging station graph
$\mathcal{D}$	Set of agents
t_0^i	Departure time of agent i
v_0^i	Start vertex of agent i
$\bar{S}^i$	Maximal distance allowed between any vertex and the start vertex v_0^i for agent i
$\bar{T}^i$	Time budget for agent i
T	Planning horizon
a_v	Binary random variable modeling the availability of v
p_v	Initial probability that charging station v is available before any visit
$t_{v,\acute{v}}$	Driving time on arc $(v, \acute{v})$
$\bar{\beta}^i$	Termination penalty cost at an occupied station v for agent i
γ_v^i	Termination cost at an available station v for agent i
Notation used to define the MDP	
$\mathcal{X}$	(Multi-agent) State Space
$\mathcal{U}$	(Multi-agent) Action Space
$\mathcal{J}$	Set of active drivers in state $x \in \mathcal{X}$
$\mathcal{T}$	Set of terminated drivers in state $x \in \mathcal{X}$
$\mathcal{O}$	Set of visited stations in state $x \in \mathcal{X}$
v^i	Station assigned to driver i in state $x \in \mathcal{X}$
t^i	time when i reaches v^i
$s^i \in \{\mathrm{d}, \mathrm{f}, \mathrm{t}, \mathrm{r}\}$	status of i in state $x \in \mathcal{X}$
$d(x, u)$	immediate cost induced by taking action $u \in \mathcal{U}(x)$ in state $x \in \mathcal{X}$
π	System policy
V^π	Value function
π^i	Single-agent i policy
δ	Binary variable indicating if at least one driver failed her search in termination state $x_n \in \mathcal{X}$
Π^{dep}	Set of policy π with *user-dependent* single-agent policies
Π^{ind}	Set of policy π with *user-independent* single-agent policies

B. Proof of Proposition 1

To prove Proposition 1, we seek to show that

$$\forall \pi \in \Pi^{\text{ind}} \text{ , } \alpha^{\pi} = V^{\pi}(x_0) \tag{B.1}$$

holds, i.e., that the expected MDP cost defined in Section 3.1 for a user-independent policy π (see Section 3.2) can be expressed as a function of all expected single-agent MDP costs (see Section 3.3). We recall that a user-independent policy can be expressed as a set of single-agent policies, with $\pi = \{\pi^i, i \in \mathcal{D}\}$ and π^i being represented by a sequence of station visits $C^i = (v_0^i, ..., v_n^i)$.

To show that Equation B.1 holds, we first reformulate the policy-specific cost function $V^{\pi}(x)$ for any state $x \in \mathcal{S}$. We then show by recursion that it holds for any state, which allows us to finally prove the equality relation for the initial state x_0.

Step 1: Given decision epoch t with deciding agent $\acute{\lambda}$, we explicitly define for both the partially successful state $\acute{x}^{\text{s}}$ and the unsuccessful state $\acute{x}^{\text{f}}$ the policy cost function $V^{\pi}(\acute{x}^{\text{s}})$ and $V^{\pi}(\acute{x}^{\text{f}})$. $\mathcal{J}'_{\text{f}}$, respectively $\mathcal{J}'_{\text{s}}$ represent the set of non-terminated drivers at $\acute{x}^{\text{f}}$, respectively $\acute{x}^{\text{s}}$. We consider for all drivers $i \in \mathcal{J}'_{\text{f}}$ their truncated policies, i.e., actions prescribed from their last visited stations in decision epoch t. We represent agent i's truncated policy by the sequence of station visits $\bar{C}^i = (\bar{v}_0^i, ..., \bar{v}_l^i)$, with $\bar{v}_0^i$ being the assigned station to i in x. With a slight abuse of notation, we refer to $\bar{v}_k^i$ as v_k^i in the remainder of this proof.

In the following, we let indices k describe the time-ordered visit events of stations included in $\bar{C}^i$ for all drivers in $\mathcal{J}'_{\text{f}}$ for both $\acute{x}^{\text{f}}$ and $\acute{x}^{\text{s}}$. Let $n(\mathcal{J}'_{\text{f}})$ be the maximum number of all possible visit events, considering all stations defined by all respective $\bar{C}^i$ such that $n = \sum_{i \in \mathcal{J}'_{\text{f}}} |\bar{C}^i|$, with $|\bar{C}^i|$ being the length of $\bar{C}^i$. We define k^i as the index in sequence $\bar{C}^i$ of the last station visited by i in stage k. The binary variable $\delta_{i,k}$ indicates whether i observes an occupied station and has enough time to select her next station to visit at stage k. In this case, we let $t(i,k)$ indicate the travel time for i from the station visited in stage k to the next planned station, i.e., $t(i,k) = t_{v_{k^i}^i, v_{k^i+1}^i}$. We let $\rho_{[k:l]}^i$ be the likelihood for i to get a least one free station among $\bar{C}^i[k^i, l^i]$, with $\rho_{[k:l]}^i = 1 - \prod_{v \in \bar{C}^i[k^i:l^i]}(1 - p_v^i)$ and p_v^i being the user-dependent availability probability. Binary variables δ'_{s}, respectively δ'_{f}, indicate whether at least one driver failed her search already, i.e., $s =$ 't', in state $\acute{x}^{\text{s}}$, respectively in state $\acute{x}^{\text{f}}$.

We now explicit the expected cost of value function V from states $\acute{x}^{\mathrm{s}}$ and $\acute{x}^{\mathrm{f}}$ as follows

$$V^{\pi}(\acute{x}^{\mathrm{s}}) = \sum_{k=0}^{n-1} \sum_{i \in \mathcal{J}'_{\mathrm{s}}} \bar{\rho}^{i}_{[0:k]} t(i,k) \delta_{i,k} + \sum_{i \in \mathcal{J}'_{\mathrm{s}}} \bar{\rho}^{i}_{[0:n]} \bar{\beta} + (\delta'_{\mathrm{s}} + (1-\delta'_{\mathrm{s}})(1 - \prod_{i \in \mathcal{J}'_{\mathrm{s}}} \rho^{i}_{[0:n]}))\beta^{G} \,,$$
$$V^{\pi}(\acute{x}^{\mathrm{f}}) = \sum_{k=0}^{n-1} \sum_{i \in \mathcal{J}'_{\mathrm{f}}} \bar{\rho}^{i}_{[0:k]} t(i,k) \delta_{i,k} + \sum_{i \in \mathcal{J}'_{\mathrm{f}}} \bar{\rho}^{i}_{[0:n]} \bar{\beta} + (\delta'_{\mathrm{f}} + (1-\delta'_{\mathrm{f}})(1 - \prod_{i \in \mathcal{J}'_{\mathrm{f}}} \rho^{i}_{[0:n]}))\beta^{G} \,, \tag{B.2}$$

with

$$\sum_{k=0}^{n-1} \sum_{i \in \mathcal{J}'_{\mathrm{f}}} \bar{\rho}^{i}_{[0:k]} t(i,k) \delta_{i,k} - \sum_{k=0}^{n-1} \sum_{i \in \mathcal{J}'_{\mathrm{s}}} \bar{\rho}^{i}_{[0:k]} t(i,k) \delta_{i,k} \overset{(*)}{=} \sum_{k=0}^{n-1} \bar{\rho}^{\acute{\lambda}}_{[0:k]} t(\acute{\lambda},k) \delta_{\acute{\lambda},k} \,, \tag{B.3}$$

using in $(*)$ that the subsequent visit decisions for all drivers but $\acute{\lambda}$ do not depend on whether $\acute{\lambda}$ observes an occupied station ($\acute{x}^{\mathrm{s}}$) or not ($\acute{x}^{\mathrm{f}}$) in decision epoch t, because all π^{i} are user-independent.

For brevity in notation, we show the case for $\gamma^{i}_{v} = 0 \; \forall i \in \mathcal{D}, v \in \mathcal{V}$, w.l.o.g., as one can apply the following transformation: (i) $t_{v^{i}_{k}, v^{i}_{k+1}} \leftarrow t_{v^{i}_{k}, v^{i}_{k+1}} + p_{v^{i}_{k+1}} \cdot \gamma^{i}_{v^{i}_{k+1}}$ and (ii) $V^{\pi}(\acute{x}^{\mathrm{s}}) \leftarrow V^{\pi}(\acute{x}^{\mathrm{s}}) + \gamma^{\acute{\lambda}}_{v}$.

Step 2: Equation B.2 holds for the last decision epoch in state $\acute{x}$ before the global termination state, in which only one non-terminated driver, for $\acute{x}^{\mathrm{f}}$ and no driver for $\acute{x}^{\mathrm{s}}$ ($n = 1$) remains. We now show that if Equation B.2 holds for t, then it also holds for $t-1$ and thus by recursion for any possible decision epoch, e.g., in the initial state with $t = 0$.

Let us consider all states x that can lead to $\acute{x}^{\mathrm{s}}$ or $\acute{x}^{\mathrm{f}}$ following policy π, such that $p(\acute{x} \in \{\acute{x}^{\mathrm{f}}, \acute{x}^{\mathrm{s}}\} | x, \pi(x)) = 1$. Let λ be the deciding agent for these states and $t-1$ be the decision epoch accordingly. For all these states, $\acute{\lambda}$ is unterminated, as it is either deciding in $\acute{x}^{\mathrm{f}}$ or successfully finished in $\acute{x}^{\mathrm{s}}$. Without loss of generality, let us shift indexes such that $k := k+1$ in the summations of $V(\acute{x}^{\mathrm{s}})$ and $V(\acute{x}^{\mathrm{f}})$. The first station visit from $\acute{x}^{\mathrm{f}}$ or $\acute{x}^{\mathrm{s}}$ in epoch t becomes the second station visit from states x^{f} or x^{s} in preceding epoch $t-1$.

For brevity, we show the case in which $\acute{\lambda}$ *is not terminated*, i.e., $\acute{\lambda} \in \mathcal{J}'_{\mathrm{f}}$ and $\mathcal{J}'_{\mathrm{f}} = \mathcal{J}'_{\mathrm{s}} \bigcup \acute{\lambda}$. In this case, $\delta'_{\mathrm{s}} = \delta'_{\mathrm{f}} = \delta_{\mathrm{f}}$. If at least one agent terminated in $\acute{x}^{\mathrm{s}}$, at least one agent must have terminated in $\acute{x}^{\mathrm{f}}$ and vice-versa since $\acute{\lambda}$ did not terminate. Similarly, if no agent terminated in $\acute{x}^{\mathrm{s}}$, then no agent terminated in $\acute{x}^{\mathrm{f}}$ since $\acute{\lambda}$ did not terminate and vice-versa. In all cases, the termination condition holds for the previous unsuccessful state x^{f}.

$$
\begin{aligned}
V^{\pi}(x^f) \overset{(*)}{=} & t(\lambda,0) + p_{v^{\acute{\lambda}}} V(\acute{x}^{\mathrm{s}}) + (1-p_{v^{\acute{\lambda}}}) V(\acute{x}^{\mathrm{f}}) \\
= & t(\lambda,0) + p_{v^{\acute{\lambda}}} (\sum_{k=1}^{n} \sum_{i \in \mathcal{J}_{\mathrm{s}}'} \bar{\rho}^{i}_{[1:k]} t(i,k) \delta_{i,k} + \sum_{i \in \mathcal{J}_{\mathrm{s}}'} \bar{\rho}^{i}_{[1:n+1]} \bar{\beta}) \\
& + (1-p_{v^{\acute{\lambda}}}) (\sum_{k=1}^{n} \sum_{i \in \mathcal{J}_{\mathrm{f}}'} \bar{\rho}^{i}_{[1:k]} t(i,k) \delta_{i,k} + \sum_{i \in \mathcal{J}_{\mathrm{f}}'} \bar{\rho}^{i}_{[1:n+1]} \bar{\beta}) \\
& + p_{v^{\acute{\lambda}}} (\delta_{\mathrm{s}}' + (1-\delta_{\mathrm{s}}')(1 - \prod_{i \in \mathcal{J}_{\mathrm{s}}'} \rho^{i}_{[1:n+1]})) \beta^{G} \\
& + (1-p_{v^{\acute{\lambda}}}) (\delta_{\mathrm{f}}' + (1-\delta_{\mathrm{f}}')(1 - \prod_{i \in \mathcal{J}_{\mathrm{f}}'} \rho^{i}_{[1:n+1]})) \beta^{G} \\
\overset{(**)}{=} & \sum_{k=0}^{n} \sum_{i \in \mathcal{J}_{\mathrm{s}}'} \bar{\rho}^{i}_{[0:k]} t(i,k) \delta_{i,k} + \sum_{i \in \mathcal{J}_{\mathrm{s}}'} \bar{\rho}^{i}_{[0:n+1]} \bar{\beta} \\
& + (1-p_{v^{\acute{\lambda}}}) (\sum_{k=1}^{n} \bar{\rho}^{\acute{\lambda}}_{[1:k]} t(\acute{\lambda},k) \delta_{\acute{\lambda},l} + \bar{\rho}^{\acute{\lambda}}_{[1:n+1]} \bar{\beta}) \\
& + p_{v^{\acute{\lambda}}} (1-\delta_{\mathrm{s}}')(1 - \prod_{i \in \mathcal{J}_{\mathrm{s}}'} \rho^{i}_{[1:n+1]}) \beta^{G} + (1-p_{v^{\acute{\lambda}}})(1-\delta_{\mathrm{f}}')(1 - \prod_{i \in \mathcal{J}_{\mathrm{f}}'} \rho^{i}_{[1:n+1]}) \beta^{G} \\
& + ((1-p_{v^{\acute{\lambda}}}) \delta_{\mathrm{f}}' + p_{v^{\acute{\lambda}}} \delta_{\mathrm{s}}') \beta^{G} \\
= & \sum_{k=0}^{n} \sum_{i \in \mathcal{J}_{\mathrm{f}}'} \bar{\rho}^{i}_{[0:k]} t(i,k) \delta_{i,k} + \sum_{i \in \mathcal{J}_{\mathrm{f}}'} \bar{\rho}^{i}_{[0:k]} \bar{\beta} + \\
& (1-\delta_{\mathrm{f}}) (p_{v^{\acute{\lambda}}} (1 - \prod_{i \in \mathcal{J}_{\mathrm{s}}'} \rho^{i}_{[1:n+1]}) + (1-p_{v^{\acute{\lambda}}})(1 - \prod_{i \in \mathcal{J}_{\mathrm{f}}'} \rho^{i}_{[1:n+1]})) \beta^{G} + \delta_{\mathrm{f}} \beta^{G} \\
\overset{(***)}{=} & \sum_{k=0}^{n} \sum_{i \in \mathcal{J}_{\mathrm{f}}} \bar{\rho}^{i}_{[0:k]} t(i,k) \delta_{i,k} + \sum_{i \in \mathcal{J}_{\mathrm{f}}} \bar{\rho}^{i}_{[0:n+1]} \bar{\beta} + (1-\delta_{\mathrm{f}})(1 - \prod_{i \in \mathcal{J}_{\mathrm{f}}} \rho^{i}_{[0:n+1]}) \beta^{G} + \delta_{\mathrm{f}} \beta^{G} \, .
\end{aligned}
\tag{B.4}
$$

We use in $(*)$ the recursive definition of V^{π} and in $(**)$ Equation B.3. In $(***)$, we exploit that $\mathcal{J}_{\mathrm{f}}' = \mathcal{J}_{\mathrm{f}}$, which follows from the following reasoning: if $\acute{\lambda}$ belongs to $\mathcal{J}_{\mathrm{f}}'$, then it is not terminated in x^{f} and accordingly belongs to $\mathcal{J}_{\mathrm{f}}$; as x^{f} is a partially unsuccessful state for λ, it follows that λ belongs to both $\mathcal{J}_{\mathrm{f}}$ and $\mathcal{J}_{\mathrm{f}}'$. We show that

$(***)$ holds, given the following equality

$$\begin{aligned}
&p_{v^{\acute{\lambda}}}(1-\prod_{i\in\mathcal{J}_{\mathrm{s}}'}\rho^i_{[1:n+1]})+(1-p_{v^{\acute{\lambda}}})(1-\prod_{i\in\mathcal{J}_{\mathrm{f}}'}\rho^i_{[1:n+1]})\\
=&p_{v^{\acute{\lambda}}}(1-\prod_{i\in\mathcal{J}_{\mathrm{s}}'}\rho^i_{[1:n+1]})+(1-p_{v^{\acute{\lambda}}})(1-\rho^{\acute{\lambda}}_{[1:n+1]}\cdot\prod_{i\in\mathcal{J}_{\mathrm{s}}'}\rho^i_{[1:n+1]})\\
=&1-(\prod_{i\in\mathcal{J}_{\mathrm{s}}'}\rho^i_{[1:n+1]})(\rho^{\acute{\lambda}}_{[1:n+1]}+p_{v^{\acute{\lambda}}}-\rho^{\acute{\lambda}}_{[1:n+1]}p_{v^{\acute{\lambda}}})\\
\overset{(****)}{=}&1-(\prod_{i\in\mathcal{J}_{\mathrm{s}}'}\rho^i_{[0:n+1]})(\rho^{\acute{\lambda}}_{[0:n+1]})\\
=&1-(\prod_{i\in\mathcal{J}_{\mathrm{f}}'}\rho^i_{[0:n+1]}),
\end{aligned}\tag{B.5}$$

using in $(****)$ that $\forall i\in\mathcal{J}_{\mathrm{s}}^{'}\ \rho^i_{[0:n+1]}=\rho^i_{[1:n+1]}$ and that

$$(1-\rho^{\acute{\lambda}}_{[0:n+1]})=(1-\rho^{\acute{\lambda}}_{[1:n+1]})(1-p_{v^{\acute{\lambda}}})\Leftrightarrow\rho^{\acute{\lambda}}_{[0:n+1]}=\rho^{\acute{\lambda}}_{[1:n+1]}+p_{v^{\acute{\lambda}}}-\rho^{\acute{\lambda}}_{[1:n+1]}p_{v^{\acute{\lambda}}}\ .$$

Finally, we have $n(\mathcal{J}_{\mathrm{f}}):=n(\mathcal{J}_{\mathrm{f}}')+1=n+1$.

We can analogously show that B.2 holds for $V^{\pi}(x^{\mathrm{f}})$ when $\acute{\lambda}$ is terminated in $\acute{x}_{\mathrm{f}}$. In this case, $\mathcal{J}_{\mathrm{s}}'=\mathcal{J}_{\mathrm{f}}'$, $\mathcal{J}_{\mathrm{f}}'=\mathcal{J}_{\mathrm{f}}\bigcup\acute{\lambda}$, $\delta_{\mathrm{f}}'=1$, and $\delta_{\mathrm{s}}'=\delta_{\mathrm{f}}$. Given that $\prod_{i\in\mathcal{J}_{\mathrm{f}}}\rho^i_{[0:n+1]}=p_{v^{\acute{\lambda}}}\cdot\prod_{i\in\mathcal{J}_{\mathrm{f}}'}\rho^i_{[1:n+1]}$, we can verify by substitution that

$$\begin{aligned}
p_{v^{\acute{\lambda}}}(1-\delta_{\mathrm{s}}')(1-\prod_{i\in\mathcal{J}_{\mathrm{s}}'}\rho^i_{[1:n+1]})\beta^G+(1-p_{v^{\acute{\lambda}}})(1-\delta_{\mathrm{f}}')(1-\prod_{i\in\mathcal{J}_{\mathrm{f}}'}\rho^i_{[1:n+1]})\beta^G&\\
+((1-p_{v^{\acute{\lambda}}})\delta_{\mathrm{f}}'+p_{v^{\acute{\lambda}}}\delta_{\mathrm{s}}')\beta^G=&\\
(\delta_{\mathrm{f}}+(1-\delta_{\mathrm{f}})(1-\prod_{i\in\mathcal{J}_{\mathrm{f}}}\rho^i_{[0:n]}))\beta^G\ .&
\end{aligned}\tag{B.6}$$

We analogously show that B.2 holds for $V^{\pi}(x^s)$.

Step 3: Finally, we have

$$\begin{aligned}
V^{\pi}(x_0)&\overset{(*)}{=}\sum_{k=0}^{n-1}\sum_{i\in\mathcal{J}_0}\bar{\rho}^i_{[0:l]}t(i,k)\delta_{i,k}+\sum_{i\in\mathcal{J}_0}\bar{\rho}^i_{[0:n]}\bar{\beta}+(1-\prod_{i\in\mathcal{J}_0}\rho^i_{[0:n+1]})\beta^G\\
&\overset{(**)}{=}\sum_{i\in\mathcal{J}_0}(\alpha^i)+(1-\prod_{i\in\mathcal{J}_0}\rho^i_{[0:n]})\beta^G\ ,
\end{aligned}\tag{B.7}$$

with $n=n(\mathcal{J}_0)$. In the initial state x_0, $\delta=0$ holds, because no driver has terminated in the initial state by assumption. We use in $(*)$ Equation B.2 and in $(**)$ the definition of α^i.

Finally, we justified that for $\pi\in\Pi^{\mathrm{ind}}$, with x_0 as initial state at the decision epoch

$t = 0$ and $\mathcal{J}_0 = \mathcal{D}$, that

$$\begin{aligned} V^{\pi}(x_0) &= \sum_{i \in \mathcal{D}} (\alpha^i) + (1 - \prod_{i \in \mathcal{D}} \rho^i_{[0:n]}) \beta^G \\ &= \sum_{i \in \mathcal{D}} F^{i^{\pi^i}}(x^i_0) + (1 - \prod_{i \in \mathcal{D}} \rho^i_{[0:n]}) \beta^G. \end{aligned} \tag{B.8}$$

This concludes the proof.

C. Pseudo code of the LH algorithm

Figure 17 shows the pseudo-code of the multi-label setting algorithm introduced in Guillet et al. (2022). We introduce the set of active labels $\mathcal{L}^a$ and let L^i_0 be the initial label corresponding to an agent's start location. We let $\mathcal{F}_{v,\acute{v}}(L)$ be the function that returns the propagated label L' (with a partial policy ending at $\acute{v}$) of L (with partial policy ending at v). The cost $\alpha(L)$ is the cost associated with label L. Function $\delta^+(L)$ returns a set of tuples $(v, \acute{v})$ which denotes all feasible physical successor locations $\acute{v} \in \mathcal{V}$ for a label L whose partial policy ends at $v \in \mathcal{V}$.

D. Additional numerical results

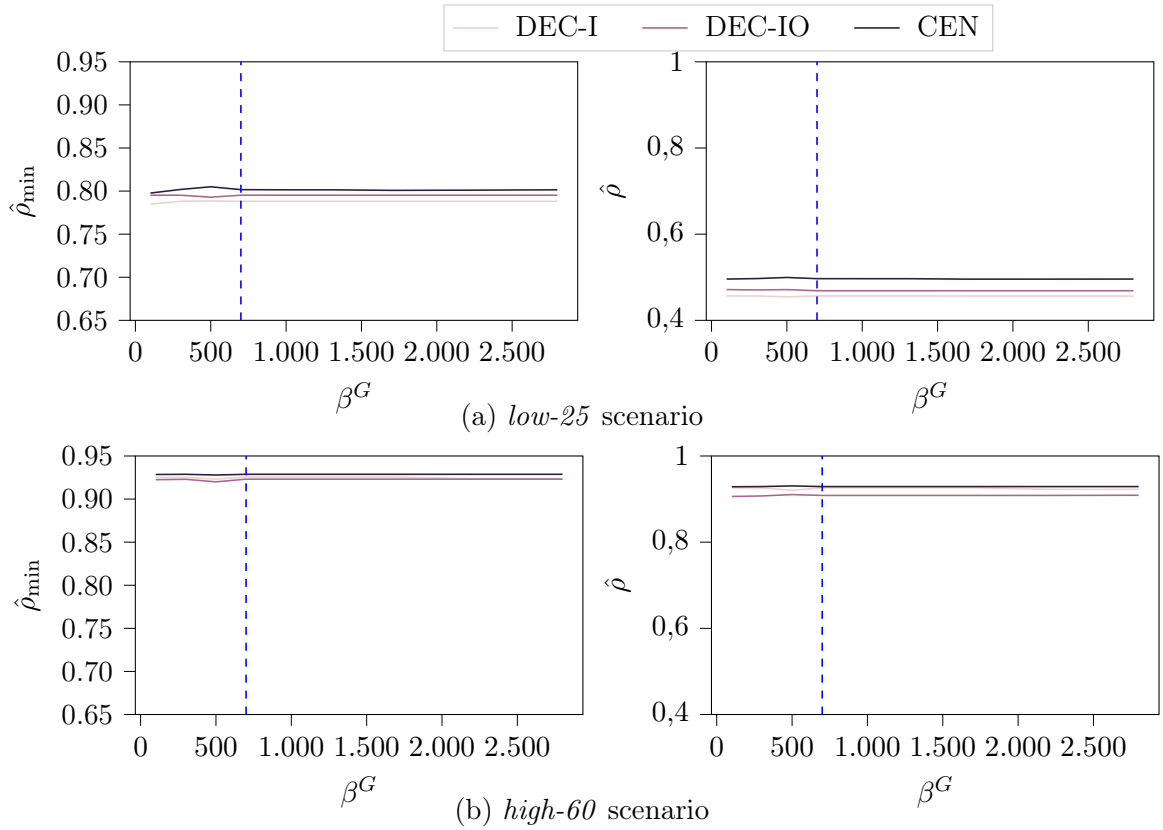
In this section, we substantiate our results discussion with additional analyses.

D.1. Sensitivity analysis

In the following, we analyze the sensitivity of the algorithmic settings DEC-I, DEC-IO, and CEN to the value of parameter β^G, and recall that DEC, DEC-O, and DEC-O-d are insensitive to this parameter.

Figure 17: LH algorithm.

$\mathcal{L}^a \leftarrow \{L^i_0\}$, $L^* \leftarrow L^i_0$
while $\mathcal{L}^a \neq \emptyset$ **do**
  $L \leftarrow$ `costMinimumLabel(`$\mathcal{L}^a$`)`
  $\mathcal{L}^a \leftarrow \mathcal{L}^a \setminus \{L\}$
  for $(v, \acute{v}) \in \delta^+(L)$ **do**
    $L' \leftarrow \mathcal{F}_{v\acute{v}}(L)$
    if `isNotDominated(`$L', \mathcal{L}^a$`)` **then**
      `dominanceCheck(`$\mathcal{L}^a, L'$`)`
      $\mathcal{L}^a \leftarrow \mathcal{L}^a \cup \{L'\}$
      if $(\delta^+(L') = \emptyset) \wedge \alpha(L') < \alpha(L^*)$ **then**
        $L^* \leftarrow L'$
return L^*



(a) *low-25* scenario

(b) *high-60* scenario

Figure 18: Impact of β^G on system and worst success rates

We compute $\rho_{\min}$ as follows: $\hat{\rho}_{\min} = \min_{i \in \mathcal{D}} \hat{\rho}^i$ with $\mathcal{D}$ being the set of drivers considered for each instance. We average values over all instances corresponding to a *low-25* or *high-60* scenario that correspond to $t^s = 300$ m.

As the value of cost $\hat{\alpha}$ depends on the value of β^G, we analyze the impact on β^G-independent metrics: Figure 18 evaluates the impact of β^G on the system success rate $\hat{\rho}$ and the worst individual success rate $\hat{\rho}_{\min}$ that a driver may obtain, while Figure 19 evaluates the impact of β^G on the average search time $\hat{t}$ and the largest search time $\hat{t}_{\max}$ that a driver may obtain in a successful search.

As can be seen, the impact of β^G on both the success rate and the search time is low. In both cases, results show opposite effects in low- and high-availability scenarios. While a value of β^G lower than 700 min yields lower search times (worst and average case) in a low-availability scenario for DEC-IO, it also yields larger search times in the worst and average case. We note that values of β^G larger than 700 have only a marginal impact for all settings and set $\beta^G = 700$ min.

D.2. Collaboration in intention-sharing settings

Similar to Figure 5, Figures 20,21&22 compare the individual driver's costs depending on their departure order, obtained with the collaborative procedure (HLH-c) and the non-collaborative procedure (HLH) for $t^s = 1$ min, respectively $t^s = 5$ min, and $t^s = 15$ min, in low and high-availability scenarios for instances with $N = 5$ drivers.

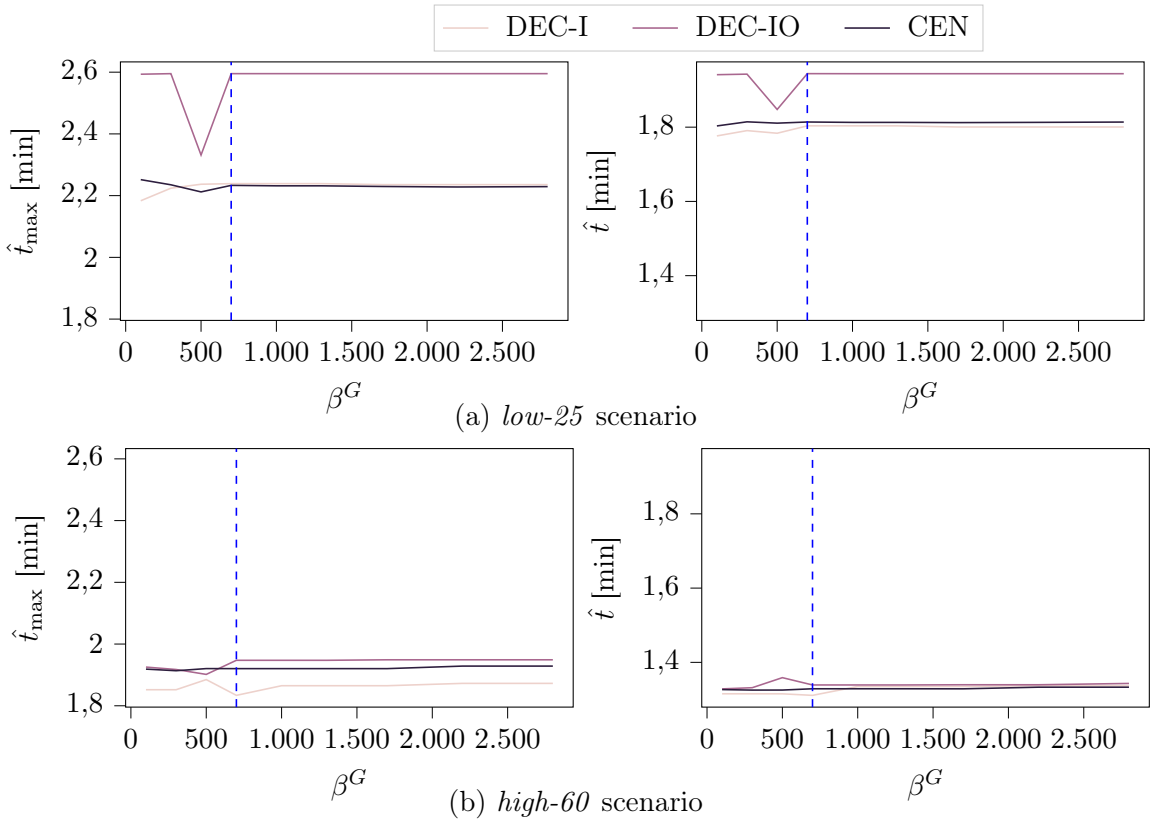


Figure 19: Impact of β^G on on average and worst search times

We compute $\hat{t}_{\max}$ as follows: $\hat{t}_{\max} = \max_{i \in \mathcal{D}} \hat{t}^i$ with $\mathcal{D}$ being the set of drivers considered for each instance. We average values over all instances corresponding to a *low-25* or *high-60* scenario and that correspond to $t^s = 300$ m.

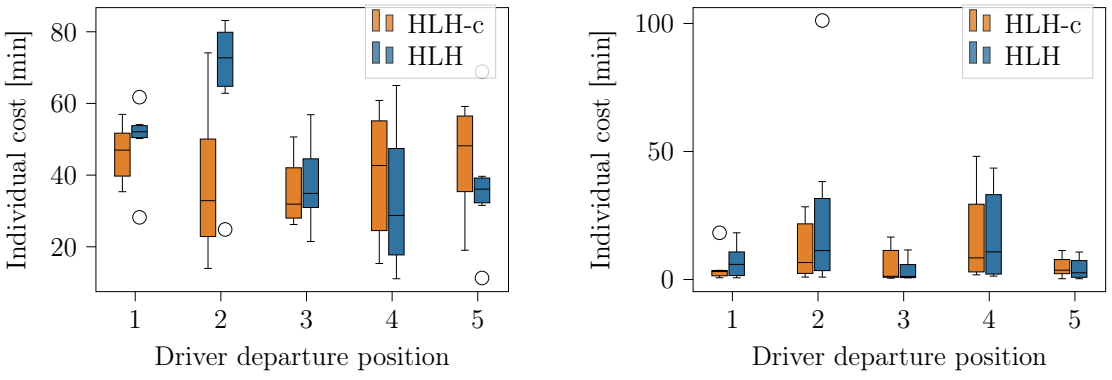
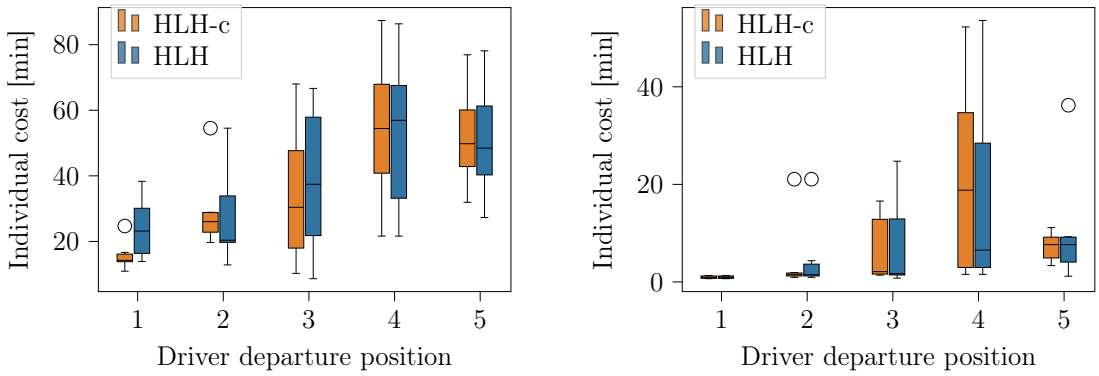


Figure 20: Comparison of individual cost $\hat{\alpha}^i$ obtained with HLH and HLH-c in the DEC-I setting for $t^s = 1$ min

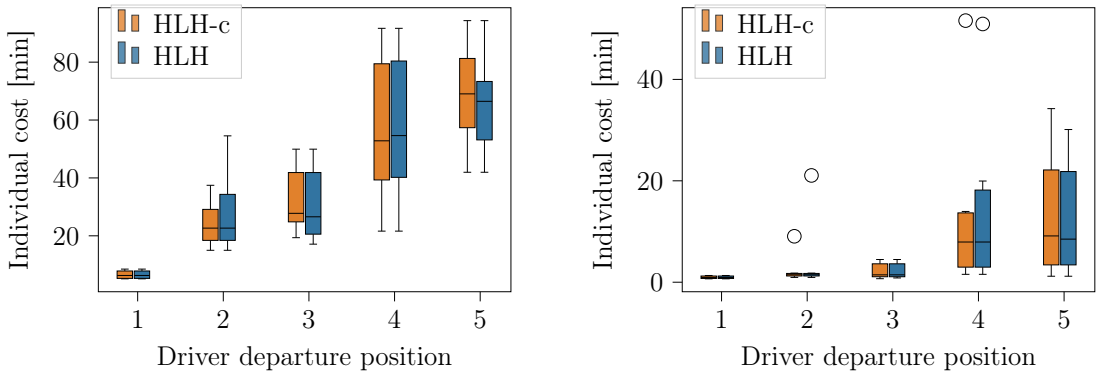
Each plot shows for each driver i, depending on her departure position, the distribution of the realized individual cost $\hat{\alpha}^i$ over all test instances that correspond to $t^s = 1$ min, $r^s \in \{100, 300, 700\}$ m, $\bar{S} \in \{1000, 2000\}$ m, and for $N = 5$ drivers.



(a) *low-25* scenario (b) *high-60* scenario

Figure 21: Comparison of individual cost $\hat{\alpha}^i$ obtained with HLH and HLH-c in the DEC-I setting for $t^s = 5$ min

Each plot shows for each driver i, depending on her departure position, the distribution of the realized individual cost $\hat{\alpha}^i$ over all test instances that correspond to $t^s = 5$ min, $r^s \in \{100, 300, 700\}$ m, $\bar{S} \in \{1000, 2000\}$ m, and for $N = 5$ drivers.



(a) *low-25* scenario (b) *high-60* scenario

Figure 22: Comparison of individual cost $\hat{\alpha}^i$ obtained with HLH and HLH-c in the DEC-I setting for $t^s = 15$ min

Each plot shows for each driver i, depending on her departure position, the distribution of the realized individual cost $\hat{\alpha}^i$ over all test instances that correspond to $t^s = 15$ min, $r^s \in \{100, 300, 700\}$ m, $\bar{S} \in \{1000, 2000\}$ m, and for $N = 5$ drivers.

As can be seen, collaboration significantly increases the solution's fairness for $t^s = 1$ min, in particular in low-availability scenarios. However for a larger t^s, collaboration affects marginally the individual solution that a driver may obtain, independent of her departure position.

D.3. General performance evaluation

Analogously to Figure 9, Figure 23 compares all decentralized settings (DEC, DEC-I, DEC-O, DEC-IO, DEC-O-dyn) with the centralized setting (CEN) with respect to cost $\hat{\alpha}$ in high-availability scenarios. We split results between short departure time horizons ($t^s \in \{0, 1\}$ min) and large departure time horizons ($t^s \in \{5, 15\}$ min), and between a small search

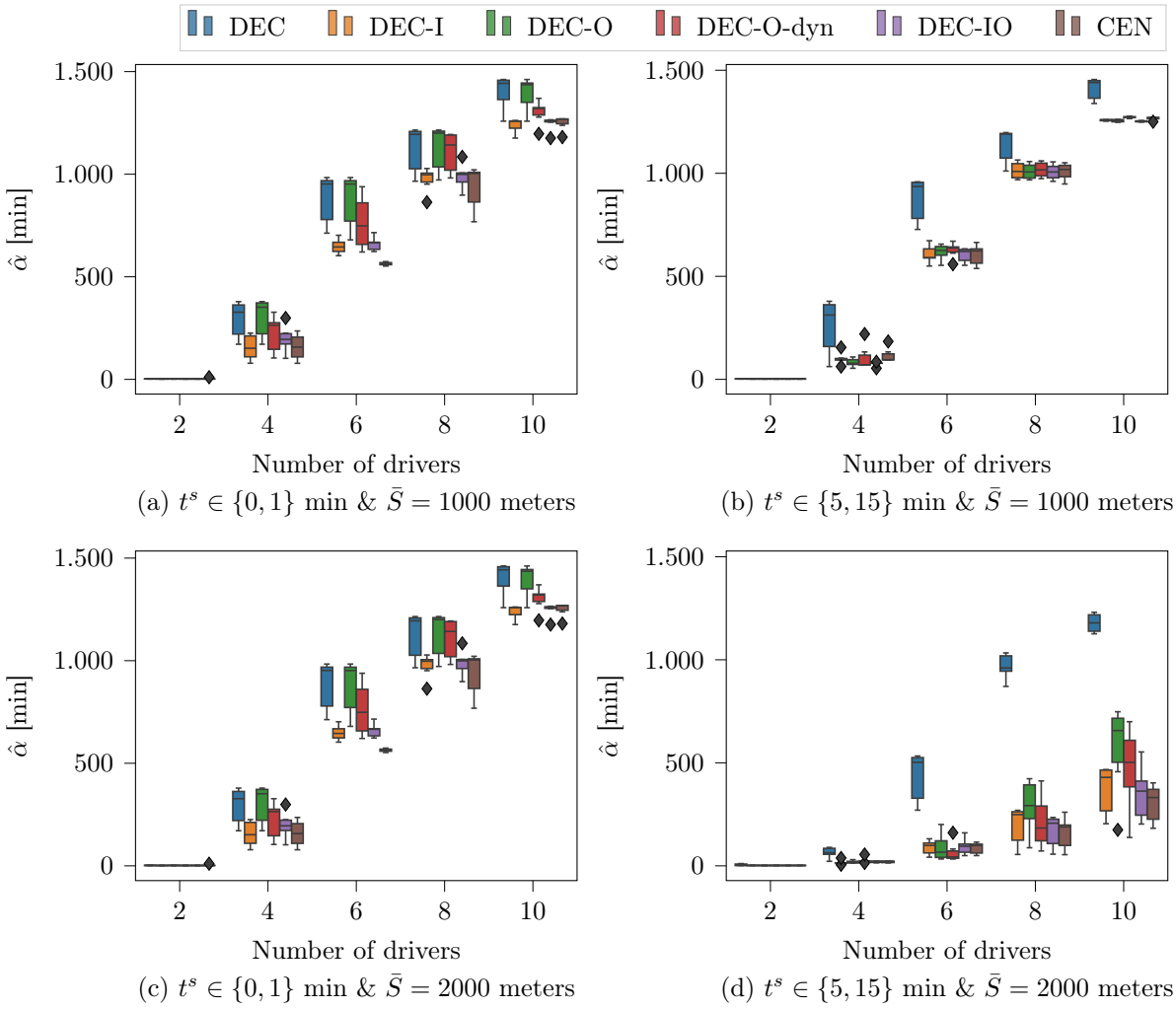


Figure 23: Comparison of decentralized and centralized decision-making in high-availability scenarios

radius ($\bar{S} = 1000$ m) and a large search radius ($\bar{S} = 2000$ meters). Overall we observe similar result trends as for low-availability scenarios. However, the difference between the improvement obtained with coordination in small search areas and the improvement obtained in large search areas is significantly higher than for low-availability scenarios. Here, the benefit of deviating from overlapping selfish solutions increases due the higher number of alternative available candidate stations compared to a low-availability scenario. We additionally note that the amplitude of the cost distribution obtained in the DEC-O-dyn setting for large search radius instances (see Figure 23a) is larger than in low-availability scenarios.

D.4. Coordination with a long planning horizon:

Similar to Figure 16, Figure 24 shows individual drivers' cost depending on their departure time, but in a low-availability scenario. Coordination generally improves a non-coordinated system, but with a less visible impact than in the high-availability scenario case, in-line with the main results discussion (cf. Figure 12). In Figure 24a, a lower search radius $\bar{S} = 1000$

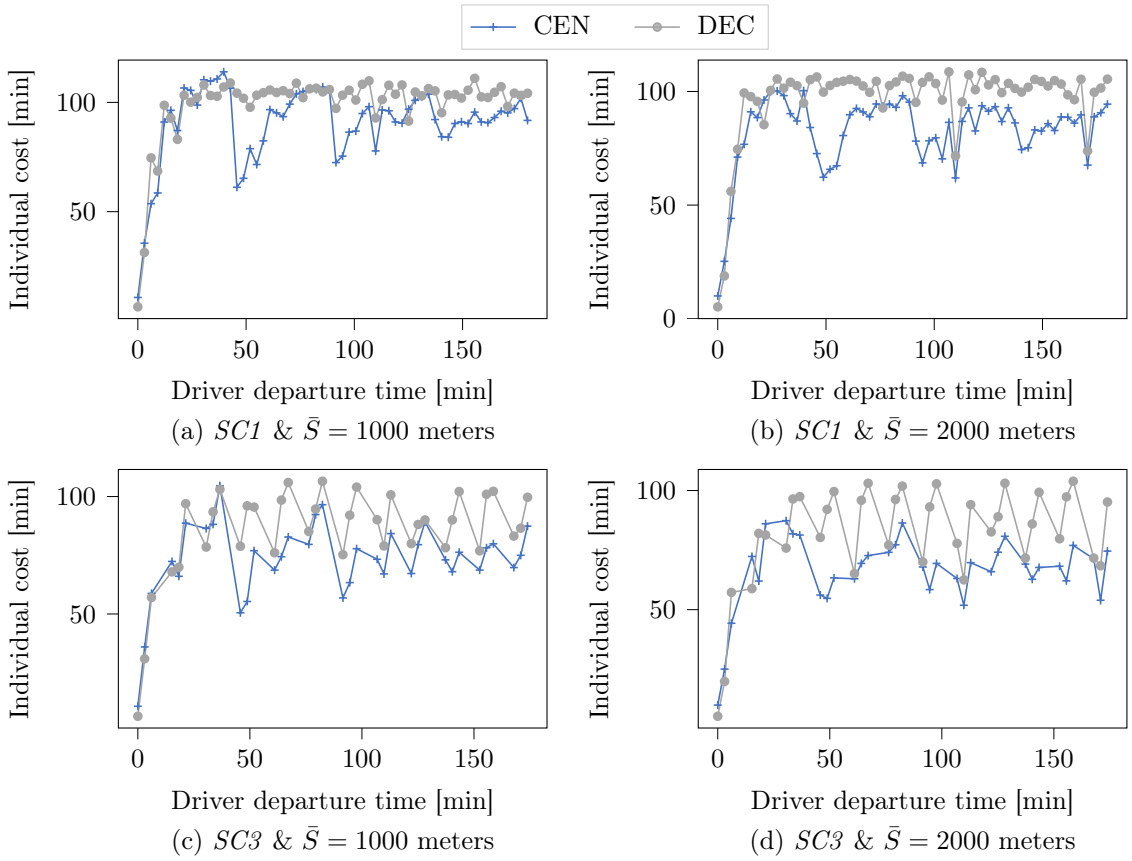


Figure 24: Comparison of DEC and CEN in high-availability scenarios for a three-hours planning horizon

Each subplot shows the individual realized cost $\hat{\alpha}_i$ for each considered setting, per number of drivers N, averaged over all values of $r^s \in \{100, 300, 700\}$ meters.

meters combined with *low-25* increases the number of drivers that compete for a very limited number of expectedly available stations, which significantly limits the overall coordination potential. In contrast to high-availability scenarios, results show a visible system transition from the initial state to congested states.

5 Coordinating charging request allocation between self-interested navigation service platforms

This chapter is based on an working article published as:

Guillet M., Schiffer M. (2022). Coordinating charging request allocation between self-interested navigation service platforms. *https://arxiv.org/pdf/2208.09530.pdf*.

Abstract

Current electric vehicle market trends indicate an increasing adoption rate across several countries. To meet the expected growing charging demand, it is necessary to scale up the current charging infrastructure and to mitigate current reliability deficiencies, e.g., due to broken connectors or misreported charging station availability status. However, even within a properly dimensioned charging infrastructure, a risk for local bottlenecks remains if several drivers cannot coordinate their charging station visit decisions. Here, navigation service platforms can optimally balance charging demand over available stations to reduce possible station visit conflicts and increase user satisfaction. While such fleet-optimized charging station visit recommendations may alleviate local bottlenecks, they can also harm the system if self-interested navigation service platforms seek to maximize their own customers' satisfaction. To study these dynamics, we model fleet-optimized charging station allocation as a resource allocation game in which navigation platforms constitute players and assign potentially free charging stations to drivers. We show that no pure Nash equilibrium guarantee exists for this game, which motivates us to study VCG mechanisms both in offline and online settings, to coordinate players' strategies toward a better social outcome. Extensive numerical studies for the city of Berlin show that when coordinating players through VCG mechanisms, the social cost decreases on average by 42% in the online setting and by 52% in the offline setting.

1. Introduction

Battery-powered vehicles support the shift towards more sustainable mobility systems, especially if coupled with the overall increased use of public transportation systems. After a slow start for EV adoption, the private electric vehicles market recently showed an encouraging growth, with global adoption doubling in 2020 and 2021, and car manufacturers planning to launch more than 100 new electric vehicle models by 2024. However, studies reveal strong adoption heterogeneity, at national and regional scales. In 2020, Norway had a (BH)EV market share of more than 70% (cleantechnica 2021), whereas Romania's EV market share was less than 10% (AVER 2021). In the latter case, the adoption is thus far hindered by poor or missing charging infrastructure. In general, charging-related anxieties caused by limited infrastructure availability dissuade conventional vehicle drivers from switching to electric vehicles. Drivers may experience so-called *range anxiety*, i.e., the fear of running out of battery, or *charge anxiety*, i.e., the fear of not finding any available and non-broken public charging station. While building new charging stations may alleviate these anxieties, operating over-dimensioned charging infrastructure in low charging demand areas can be highly cost-inefficient (Nelder and Rogers 2019). Overcoming this chicken-and-egg dilemma, i.e., deciding whether to pursue EV adoption or charging infrastructure first, can be mitigated with appropriate infrastructure planning and operation (cf. Enlit 2022).

From a short-term perspective, drivers' anxieties related to charging in an undersized infrastructure can be addressed via advanced operational planning, e.g., a navigation system that reliably guides EV drivers until they find an available station, anticipating possibly blocked, unreachable, or out-of-order stations. To foster EV adoption from a long-term perspective, regulatory measures, e.g., a ban on conventional vehicles in city centers or financial incentivization are necessary, but must be complemented by planning solutions. Such solutions include developing the charging infrastructure with new equipment, consolidating existing urban infrastructure (e.g., street lamp charging), or increasing the reliability of real-time availability information through detection sensors or cameras.

The expected growing charging demand can nevertheless create new bottlenecks, i.e., charging station congestion, if demand is misaligned with charging infrastructure capacities. Even with an increased charging infrastructure coverage, several EV drivers may receive identical charging station recommendations from navigation services platforms, which creates station utilization conflicts due to long recharging times. Here, fleet-optimized recommendations – enabled by requests centralization – can better distribute the charging demand over the available charging stations and increase the overall users' satisfaction (cf. Guillet and Schiffer 2022).

Such a fleet-optimization may work very well in presence of a single platform but finds its limits with the presence of several self-interested navigation platforms, each seeking to maximize their own drivers' utilities. In this case, the overall outcome of the drivers' assignment to stations may be far from optimal due to conflicting charging station allocations in between

platforms. Accordingly, solely improving the infrastructure with no further charging demand coordination at the system level may worsen the EV charging experience by decreasing the likelihood of finding an available station within a minimum amount of time. If a system composed of multiple navigation services platforms cannot reach any good equilibrium, a centralized non-profit-driven authority, e.g., a municipality, may aim to improve the system towards a socially efficient outcome, using financial incentives or penalties.

To understand the dynamics of such a setting, we introduce the fleet charging station allocation (FCSA) game in which several self-interested navigation platforms aim to individually optimize the assignment of available stations to their EV drivers, i.e., their charging demand. In this context, the scope of this paper is twofold. First, we use a game-theoretical framework to analyze the dynamics and bottlenecks that may arise in this setting due to non-cooperative driver assignments. We show that no guarantee for a pure Nash equilibrium (PNE) exists, which motivates our second scope: deriving mechanisms to stabilize and coordinate the charging station allocation, studying the impact of aligning platforms' interests from a platform but also from a driver perspective.

1.1. Related Work

Our work relates to indivisible resource allocation problems for electric vehicles (EVs) and other related applications, e.g., parking utilization, which we review in the following. We first discuss resource allocation games, particularly with a focus on equilibrium analyses, before we review works that apply mechanism design to resource allocation problems.

Congestion games, initially introduced in Rosenthal (1973), model non-cooperative atomic resource allocation between homogeneous agents. Here, a player's strategy is defined as a subset of the available resources, whose utilization cost depends on the total number of players selecting the resource. This class of games naturally applies to load-balancing (see, e.g., Even-Dar et al. 2003, Goemans et al. 2006), network design (see, e.g., Anshelevich et al. 2008), or internet routing (see, e.g., Secci et al. 2011). Congestion games fall into the category of potential games (cf. Monderer and Shapley 1996), i.e., games that admit a potential function, which describes the payoff variation between two different strategies independent of the players. Such games possess a guaranteed PNE existences. Milchtaich (1996)'s work extends Rosenthal (1973)'s work by showing the existence of a PNE in congestion games with player-specific payoff functions or weighted players, but restricted singleton strategies. Ackermann et al. (2009) later show that strategies that consist of the bases of a matroid defined on the resources set still guarantee the PNE'existence for games analyzed in Milchtaich (1996).

Most EV-related games deal with charging capacity allocation in grid networks, focused on optimizing grid operators' revenues (Tushar et al. 2012), reducing energy peak load through load and capacity-dependent energy prices (Sheikhim et al. 2013), or mitigating the impact of charging vehicles on transportation networks (Sohet et al. 2021). Ayala et al. (2011) analyze

related competitive parking slot assignment problems, and show the existence of a PNE when each player, i.e., driver, must select at most one parking spot for which a successful utilization depends on her and other players' driving distances to all parking spots. He et al. (2015) further extend the result of Ayala et al. (2011) by analytically characterizing equilibrium parking slot assignments.

While the FCSA game studied in this paper resembles a congestion game, its payoff functions are user-specific and depend on the number of other players selecting an identical resource only under some conditions: while all drivers competing for the same resource affect each other's payoff in a classical congestion game, in the FCSA game only drivers that are closer to utilizing the resource affect the payoff function of drivers that are further away from utilizing the resource. Moreover, congestion games have not yet been studied in the context of charging station allocation via intermediate self-interested participants.

To deal with inefficient equilibria, mechanism design (MD) theory aims to design games in such a way that a socially desirable outcome is reached (Nisan et al. 2007). A third party, a so-called principal, centrally allocates goods or resources to its participants, who must reveal their preferences on the resources before paying a price for the received resource. In this context, the well-known VCG mechanism (Vickrey 1961, Clarke 1971, Groves 1973) defines a pricing scheme that ensures revelation truthfulness by aligning participants and the overall system interests in an offline setting. As an offline allocation may be hard to realize in practice, Parkes and Singh (2003) propose two online VCG mechanisms, namely delayed and online VCG, that utilize a Markov decision process (MDP) to derive the minimal expected cost allocation, while ensuring that expected participant utilities align with the expected system interest. They show the Bayesian-Nash truthfulness of these mechanisms and apply them successfully to WiFi pricing (Friedman and Parkes 2003). As the pricing scheme relies on an optimal policy argument, the authors later discuss approximately efficient online MD using ϵ-efficient policies. More recently, Stein et al. (2020) proposed a reinforcement learning-based mechanism that guarantees strategy-proofness when the resource allocation problem is solved online. Focusing on EV charging, Rigas et al. (2020) apply standard VCG pricing schemes to achieve efficient charge scheduling of self-interested EV drivers by minimizing the charging demand imbalance at a station. Gerding et al. (2011) design a mechanism to solve the online charge scheduling problem, blind to future charging demand. Several works have studied mechanisms in the related context of parking slot allocation. Xu et al. (2016) derive a trading mechanism to share private parking slots during office hours in big cities between self-interested owners, to remedy limited public parking spot availability. Zou et al. (2015) extend the parking slot assignment game of Ayala et al. (2011) by applying VCG mechanisms to a publicly-owned parking facility that aims to maximize the social welfare, both in static and dynamic settings. Wang et al. (2020) derive an optimal allocation pricing scheme based on a Demange-Gale-Sotomayor-based mechanism, for reservable and non-reservable parking resources in a city.

In summary, most EV-related settings focus on charge scheduling problems with homogeneous self-interested EV drivers, while this work focuses on navigation platforms as self-interested participants. Similarly, work on parking slot allocation problems focus on directly incentivizing drivers to truthfully report their valuations, but not on intermediaries that could balance aggregated parking demand over available slots.

1.2. Aims and Scope

To close the research gaps outlined above, we introduce a game-theoretical setting to study the dynamics of self-interested navigation platforms that aim to best balance their own clients' charging requests among available charging stations, focusing first on equilibrium analysis, and second on mechanism design. We formalize the resulting charging station allocation problem as a novel game, and apply both offline and online VCG mechanisms to ensure socially desirable outcomes in idealized but also in practical settings. Specifically, our contribution is three-fold. First, we define the FCSA game in a perfect-information setting as a new resource allocation game, and show that this game does not possess a guaranteed PNE, but also no approximated PNE with a sufficiently small approximation factor. Second, we coordinate players by applying VCG pricing in both an offline and an online setting, accounting for weighted players. We extend the delayed VCG mechanism to a weighted delayed VCG mechanism, and implement a data-driven online assignment policy. Finally, we conduct extensive simulation-based experiments to analyze the benefits of our coordinated allocation mechanism. Our results show that players' coordination via the VCG mechanism can decrease the social cost on average by 52% in an offline setting and by 42% in an online setting when using our data-driven assignment policy. Our results further show that a player with a small share of drivers has a greater interest in participating in the VCG mechanism than a player with a larger share of drivers. A player's payoff relative to its number of drivers decreases when its share of managed drivers increases.

1.3. Organization

The remainder of this paper is structured as follows. Section 2 details our FCSA game setting and corresponding equilibrium analyses. In Section 3, we introduce the VCG mechanisms, which we apply in both offline and online settings. Section 4 describes our experimental design, while we discuss our numerical results in Section 5. Section 6 concludes this paper and provides an outlook for future research. For the sake of conciseness, we shift all proofs to the Appendix.

2. Problem Setting

We focus on an non-cooperative offline resource allocation problem, where several navigation platforms aim to optimally assign their EV drivers to available charging stations, such that no visit conflict arises for their drivers and the total travel time for drivers to assigned stations is minimal. In the following, we consider a perfect-information setting, in which each platform is aware of overall charging demand, and of accurately reported real-time charging station availabilities. Note that such a perfect-information setting is unrealistic in practice but allows us to analyze whether stable players' strategies exist at least in an idealized setting. Each platform optimally assigns free stations to its navigated EV drivers, who accordingly drive to the stations to recharge their vehicles. To focus on the dynamics between the navigation platforms, we assume that electric drivers do not deviate from their assigned stations. In case of a conflicting assignment, i.e., if two or more platforms assign one of their drivers to the same station, the station availability is guaranteed for the driver with the earliest arrival time. The other assigned drivers fail their search, which induces a penalty cost for the respective platform.

In the following, we formalize our problem setting as a resource allocation game, show that a PNE guarantee does not exist, and discuss the limits of refinement equilibrium concepts within this context.

2.1. Fleet Charging Station Allocation game

We now define the FCSA game, in which navigation service platforms constitute the players. A player's strategy is an assignment of charging stations to its drivers, while its payoff describes the cost of the assignment, corresponding to the sum of the time needed by each driver to travel to its assigned station. We assume that if multiple drivers are assigned to the same station, all non-closest drivers must pay an extra penalty cost that represents the discomfort for failing to find a free charging station. We further restrict the set of reachable stations for a driver by setting a maximal driving time, and assume a deterministic driver behavior as reasoned in the section's introduction.

Formally, we consider a set of players $\mathcal{N}$, a set of drivers $\mathcal{D}$, and a set of available charging stations $v \in \mathcal{V}$. We denote with $\mathcal{V}' = \mathcal{V} \bigcup \{v_0\}$ the set of charging stations extended with an artificial station v_0, which represents a non-feasible assignment of a driver to any physical station. Such non-feasible assignments may occur when the number of drivers exceeds the number of available stations. We let $t_{k,v}$ be the driving distance for driver k to charging station v. Each player must serve a charging demand, that corresponds to a set of driver locations, denoted by $\mathcal{D}_i = \{k_i \in \mathcal{D}\}$, $\forall i \in \mathcal{N}$. A player's pure strategy s_i is an allocation function of charging stations to drivers, with $s_i : \mathcal{D} \mapsto \mathcal{V}'$. We restrict the stations that a driver k may visit to the stations reachable within $\bar{S}$ meters, i.e., $s_i(k) \in \{v : v \in \mathcal{V}', \gamma(k, v) \leq \bar{S}\}$, with $\gamma(k, v)$ being the distance from driver k to station v. A player can assign a physical

station at most once to its drivers, but can assign an artificial vertex v_0 as many times as needed, formally

$$\forall k, k' \in \mathcal{D}_i,\ (s_i(k) = s_i(k')) \wedge (s_i(k) \neq v_0) \Rightarrow (k = k') \ . \tag{2.1}$$

A player must further assign a real station to each of its drivers if possible, i.e., at most $|\mathcal{D}_i| - |\mathcal{V}|$ drivers can be assigned to the artificial station v_0. We let the strategy profile $s = (s_i)_{i \in \mathcal{N}}$ be the vector of strategies of all players, s_{-i} be the vector of strategies for all players but i, and S_i be the strategy space for player i. We denote with $u_i(s)$ the payoff for player i when strategy profile s is played, which corresponds to the assignment cost of s_i, resulting from the sum of its individual driver's payoff functions c_i, given the other competing players' strategies s_{-i}. Formally, we define $u_i(s)$ as

$$u_i(s) = \sum_{k \in \mathcal{D}_i} c_i(k, s) \ , \tag{2.2}$$

with

$$c_i(k,s) = \begin{cases} \bar{\beta}, & \text{if } s_i(k) = v_0 \\ t_{k,s_i(k)}, & \text{if } \forall k' \in \mathcal{D}_j \ \ \forall j \in \mathcal{N},\ j \neq i\ (s_j(k') = s_i(k)) \Rightarrow (t_{k',v} \geq t_{k,v}) \\ t_{k,s_i(k)} + \bar{\beta} & \text{otherwise} \end{cases} \tag{2.3}$$

An individual driver k's payoff $c_i(k, s)$ ensures that, in case of conflicting driver to station assignments, all but the earliest arriving drivers assigned to a station v must pay a penalty cost $\bar{\beta}$. The penalty $\bar{\beta}$ represents the discomfort for failing the charging station search. Then, the goal of each player i is to find a strategy s_i that minimizes $u_i(s_i, s_{-i})$, given that the assignment cost depends on other players' selected strategies.

We consider the pure-strategy profile s^* to be a PNE when it holds that

$$\forall i \in \mathcal{N}, s_i^* \in \arg\min_{s_i \in S_i} u_i(s_i, s_{-i}^*) \ , \tag{2.4}$$

i.e., no player can unilaterally decrease its payoff by changing its strategy. The sum of all players' payoffs defines the social cost.

From a system-perspective, i.e., in an idealized setting with collaborative players, the goal is to find a strategy profile that minimizes the social cost. We refer to the minimum total cost as the social optimum, defined as

$$S_g = \min_{s \in \mathcal{S}} \sum_{i \in \mathcal{N}} \sum_{k \in \mathcal{D}_i} c_i(k, s) \ . \tag{2.5}$$

With A_v being the assignment of drivers to station v, $s(k)$ being the station assigned to driver k, and strategy profile s, we can reformulate the social optimum as

$$S_g = \min_{s \in \mathcal{S}} \sum_{v \in \mathcal{V}} \sum_{k \in A_v} t_{k,s(k)} + (|A_v| - 1) \cdot \bar{\beta} \ . \tag{2.6}$$

2.2. Equilibrium analysis

To analyze possible stable outcomes of the FCSA game, we discuss the FCSA game's equilibrium properties in the following. Here, we note that a Nash equilibrium corresponds to a

strategy profile such that no player has an incentive to unilaterally deviate from its current strategy. Moreover, there always exists a Nash equilibrium for a strategy profile of mixed strategies, i.e., a profile that corresponds to a probability distribution over pure strategies. However, mixed strategies do not reflect the behavior of a player in practice, as a player assigns a set of physical stations to its drivers once, which makes the interpretation of mixed strategy Nash Equilibria difficult for real-world analyses. In contrast, a PNE allows for more interpretable outcomes but is not guaranteed to exist in the FCSA game.

Proposition 1. *The FCSA game does not always possess a PNE.*

Proof. We refer to Appendix A for a proof of Proposition 1. □

For games with non-guaranteed PNE existence, ρ-equilibria with pure strategies constitute an interesting alternative to interpret the outcome of a game. Specifically, a ρ-equilibrium corresponds to a near-stable state in which each player cannot decrease its payoff by more than the absolute factor $\rho \geq 0$. In practice, such an equilibrium corresponds to each player accepting to slightly deviate from the best solution it may obtain if its deviation stabilizes the game outcome. Formally, this corresponds to a profile s, such that

$$\forall i \in \mathcal{N}, \forall s'_i \in \mathcal{S}_i \ \ , u_i(s'_i, s_{-i}) > u_i(s) - \rho \ . \tag{2.7}$$

However, we can assume that such equilibrium refinement is only of practical interest if a player's payoff marginally increases when the player deviates from its best response, i.e., if ρ is sufficiently small. In Proposition 2, we show that the existence of an equilibrium with ρ being sufficiently small cannot be guaranteed, as we can construct instances in which for any set of strategy profiles at least one player can significantly decrease its payoff by deviating from its current strategy.

Proposition 2. *The existence of a ρ-PNE with $\rho < \bar{\beta} - \Delta_t$, with $\Delta_t = t_{max} - t_{min}$ being the difference between the largest and the lowest driving times between a driver and a station, cannot be guaranteed for the FCSA game.*

Proof. We refer to Appendix A for a proof of Proposition 2. □

3. Charging station allocation mechanism

In Section 2, we show that no pure PNE guarantee exists for the FCSA game, which motivates our interest to better coordinate the different players via MD to steer the system towards more socially desirable outcomes. To realize such a mechanism, we consider the existence of a white label operator (WLO) who decides on the system-optimal allocation of stations to drivers across several players, considering station preferences reported by all players. In practice, the WLO might be an inter-charge operator or a municipality that allocates, e.g., via a charging slot booking system, a subset of stations to the navigation platforms, who then assign the stations to their drivers.

Formally, we consider $m = |\mathcal{V}|$ charging stations constituting m indivisible resources to be concurrently allocated among $|\mathcal{N}|$ players. Each player has a cost valuation for each bundle of stations, that corresponds to the minimum cost assignment between its drivers and the bundle's stations. Valuations are non-additive, e.g., if a bundle contains more stations than drivers, removing the non-assigned station from the bundle preserves the valuation. While the valuation is expressed in units of time in the following, it can be transposed to monetary units in practice.

Observation 1. *As the principal already knows where stations are located, it can compute a player's cost valuation if it also knows where a player's drivers are located.*

Accordingly, it suffices that a player reports the locations of its drivers to the principal rather than explicitly communicating its cost valuation for every bundle.

Observation 2. *Reporting drivers' locations is linear in the number of drivers owned by the player.*

Accordingly, reporting drivers' locations requires much less information-sharing between a player and the principal, than if a player would report its cost valuation for any possible combination of stations.

The principal computes the allocation of stations to drivers that minimizes the sum of all players' bundle cost valuation and requires in return that each player pays a price for the station assignment. Each player aims to minimize its payoff, which corresponds to the sum of its cost valuation for the received station allocation and the price decided by the principal. We assume that players may lie about their preferences if lying can decrease their payoff compared to truthfully reporting their preferences. The goal of the principal is then to design a pricing rule, such that all players have an incentive to truthfully report their drivers' locations. To ensure players' truth-telling behavior, we apply the well known VCG pricing scheme which ensures that it is in a player's best interest to reveal its true preference.

In the following, we first analyze the VCG pricing scheme from an offline perspective in Section 3.1, to derive an upper bound on the system efficiency improvement that can be reached. We then study how to design an online VCG mechanism for a practical implementation in Section 3.2.

3.1. Offline charging demand

We now develop an offline mechanism to ensure truthful reporting of the players' information, i.e., the locations of their drivers. Focusing on an offline setting, we assume all players' information to be simultaneously reported. The principal allocates stations based on the revealed information, and each player accordingly receives a subset of stations and the corresponding drivers assignment for each of these stations. In Section 3.1.1, we discuss the mechanism for unweighted players, whereas we discuss how to account for weighted players in Section 3.1.2.

3.1.1. Unweighted VCG Mechanism

In the following, we adapt the VCG mechanism for payoff minimization and social cost minimization. We consider a setting in which each player $i \in \mathcal{N}$ has some hidden information θ_i, representing the set of its drivers' locations that it chooses to accurately reveal or not. We denote with $\theta = (\theta_i)_{i \in \mathcal{N}}$, the vector that contains all players' information. Let $a \in A$ be the set of alternatives that correspond to assignments of stations to drivers, which the principal computes and communicates to the players. Here, a real station can be assigned only once to a driver and a driver may only be assigned to a station if the driving distance is less than its maximum search radius $\bar{S}$. The artificial station vertex v_0 can be assigned to multiple drivers and induces a penalty cost for each assigned driver.

We introduce the function $f(\theta)$ that minimizes the sum of all players' valuations based on their reported information θ, and refer to it as the social choice function

$$f(\theta_1, ..., \theta_n) = \arg\min_{a \in A} \sum_i v_i(\theta_i, a) \ . \tag{3.1}$$

The valuation $v_i(\theta_i, a)$ of player i for alternative a corresponds to the sum of assignment costs of drivers to stations given the player's information θ_i, formally

$$v_i(\theta_i, a) = \sum_{k \in \mathcal{D}_i} c(k, a) \ , \tag{3.2}$$

with $c(k, a)$ being the cost of assigning driver k to station $a(k)$

$$c(k, a) = \begin{cases} \bar{\beta}, & \text{if } a(k) = v_0 \\ t_{k,a(k)}, & \text{else} \ . \end{cases} \tag{3.3}$$

Within an allocation a, each driver $k \in \mathcal{D}$ receives an assigned station $a(k) \in \mathcal{V} \bigcup v_0$. Using Equation 3.2, we can further express the social choice function as

$$f(\theta_1, ..., \theta_n) = \arg\min_{a \in A} \sum_{k \in \mathcal{D}} c(k, a) \ . \tag{3.4}$$

Note that the social choice function minimizes the social cost as defined in Equation 2.6, with the additional constraint that no more than one driver can be assigned to each physical station.

Adapting the VCG mechanisms for payoff minimization leads to non-positive VCG prices, which implies a positive transfer of money from the player to the system. Accordingly, the VCG pricing rule is given by

$$p(\theta_{-i}) = h_{-i} - \sum_{j \neq i} v_j(\theta_j, a) \ , \tag{3.5}$$

with

$$h_{-i} = \min_{b \in A} \sum_{j \neq i} v_j(\theta_j, b) \ , \tag{3.6}$$

and with $a = f(\theta)$, such that each player's payoff function results to

$$u_i = v_i(\theta_i, a) - p(\theta_{-i}) = \sum_{j \in \mathcal{N}} v_j(\theta_j, a) - h_{-i} \ . \tag{3.7}$$

Using VCG prices, the principal aligns each player's interest with the overall system interest

as the term $\sum_{j \neq i} v_j(\theta_j, a)$ in Equation 3.7 aligns the player payoff with the social cost (cf. Equation 2.6). Thus, a player minimizes its payoff when telling the truth, i.e., the pricing scheme guarantees incentive-compatibility (cf. Nisan et al. 2007). VCG prices further ensure that a player does not have an incentive to assign different drivers to each of the stations contained in its received subset (see Proposition 3).

Proposition 3. *A Player has no incentive to deviate from the drivers assignment to stations prescribed by the allocation a chosen by the principal, when telling the truth.*

Proof. We refer to Appendix A for a proof of Proposition 3. □

Moreover, VCG prices ensure that truth-telling minimizes a player's payoff even when it has the possibility to deviate from the received assignment (see Proposition 4). A player that misreports information on its drivers may receive a suboptimal stations assignment, which it may improve by modifying the assignment of its drivers to the stations contained in the received subset of stations. However, the resulting assignment will still yield higher payoff than when truthfully reporting its information and not deviating from the prescribed assignment.

Proposition 4. *A Player has no incentive to lie on its revealed information, even when it can deviate from the realized assignment by the principal*

Proof. We refer to Appendix A for a proof of Proposition 4. □

3.1.2. Weighted VCG Mechanism

The presented VCG mechanism leads to a minimal social cost but does not ensure solution fairness for a single player, e.g., it may lead to an outcome with one player having none of its drivers assigned at all, while another player may have all of its drivers assigned. Accordingly, a player might be better off in some scenarios by not participating in the system-based allocation unless it gets prioritized. To mitigate these issues, we now study a VCG mechanism, in which players are weighted to better balance the resulting individual player's assignments. We note that a VCG mechanism with weighted players guarantees truthfulness (cf. Roberts 1979).

To formalize this setting, we detail

$$f(\theta_1, ..., \theta_n) = \arg\min_{a \in A} \sum_i w_i \cdot v_i(a) \ , \tag{3.8}$$

with w_i being the weight associated to player i.

In this setting, the weighted pricing rule is

$$p^{\mathrm{w}}(\theta_{-i}) = h_{-i} - \sum_{j \neq i} \frac{w_j}{w_i} v_j(\theta_j, a) \ , \tag{3.9}$$

with h_{-i} being a cost independent of the allocation obtained by i, and defined as

$$h_{-i} = \min_{b \in A} \sum_{j \neq i} \frac{w_j}{w_i} v_j(\theta_i, b) \tag{3.10}$$

to ensure positive money transfer from the players to the system.

Weights optimization: Assuming a weighted players VCG mechanism, the weights can be defined a-priori based on the players' characteristics, e.g., the number of controlled drivers. We can also optimally derive weight values such that each player benefits from participating in the centrally optimized system. Deciding whether to participate in the system or not relates to a game, where each player can {opt-in, opt-out} of the system. By opting in, the player receives the VCG allocation and pays the related price, whereas by opting out, the player selfishly assigns stations to drivers, possibly conflicting with other players' assignments. We assume that if one player opts out, then all players are forced to opt out and selfishly assign their charging demand, as the noise created by the opted-out participant prevents computing an optimal system assignment for the remaining participants. Thus, a player has only an incentive to opt in if its resulting payoff with VCG is lower than its payoff without VCG. Accordingly, our goal is to derive weights such that all players profit from participating in the VCG mechanism.

The resulting weights optimization problem can be formalized as follows: let $b = (b_i)_{i\in\mathcal{N}}$ be the selfish strategy profile for all players, and let $u_i(b)$ be the respective payoffs for each player i with strategy profile b_i. As we seek to derive the best possible weights, we adapt the pivot rule such that the minimization part of h_{-i} for player i is independent of the a-priori undefined weights for all players but i, and let

$$h_{-i} = \min_{b\in A} \sum_{j\neq i} \frac{1}{w_i} v_j(\theta_i, b) \ . \tag{3.11}$$

Equation 3.11 still guarantees the price to be non-positive such that the mechanism does not transfer a positive amount of money to its players. We denote with OPT_{-i} the social optimum realized when i does not exist, as

$$\mathrm{OPT}_{-i} = \min_{b\in A} \sum_{j\neq i} v_j(\theta_i, b) \ . \tag{3.12}$$

We let $w = (w_i)_{i\in\mathcal{N}}$, with $1 \leq w_i \leq W$, and W being the maximal weight value; and define the weights optimization problem as

$$\min_{x\in\mathcal{A},w} \sum_{i\in\mathcal{N}} w_i \cdot \sum_{k\in\mathcal{D}_i} \sum_{v\in\mathcal{V}\cup\mathcal{V}_0} x_{kv} t_{kv} \tag{3.13}$$

$$\sum_{k\in\mathcal{D}} x_{kv} \leq 1 \ \forall v \in \mathcal{V}\cup\mathcal{V}_0 \tag{3.14}$$

$$\sum_{v\in\mathcal{V}\cup\mathcal{V}_0} x_{kv} = 1 \ \forall k \in \mathcal{D} \tag{3.15}$$

$$\sum_{j\in\mathcal{N}} w_j \cdot \sum_{k\in\mathcal{D}_j} \sum_{v\in\mathcal{V}\cup\mathcal{V}_0} x_{kv} t_{kv} - \mathrm{OPT}_{-i} \leq w_i \cdot u_i(b) \ \ \forall i \in \mathcal{N} \tag{3.16}$$

The objective (3.13) minimizes the weighted social cost. The first two constraints (3.14&3.15) are assignment constraints, while the last constraint (3.16) ensures that the resulting payoff within the mechanism, i.e., $v_i(\theta_i, x) - p_i^{\mathrm{w}}(\theta_{-i})$, is better for each player compared to the payoff when acting selfishly.

3.2. Online charging demand

An offline setting as described in Section 3.1 is not applicable in practice, because it is not possible to delay the station assignment when charging requests arrive in the system. Against this background, we extend the offline VCG mechanism to an online setting to allow for immediate assignment decisions by implementing a delayed VCG mechanism similar to Parkes et al. (2004). Section 3.2.1 introduces the mechanism for both unweighted and weighted players, while Section 3.2.2 describes our online allocation policies.

3.2.1. Delayed VCG Mechanism

In this online setting, different platforms interact with the principal and sequentially reveal their drivers' locations to the principal during the planning horizon. We assume that requests must be served immediately. Then, the goal of the principal is to make decisions over time that minimize the expected total cost assignment of all drivers in the system. In this setting, each player makes a delayed payment at the end of the planning horizon, that depends on the realized assignment.

The information vector θ_i of each player is sequentially revealed, such that $\theta_i = (\theta_i^k)_{k \in \mathcal{D}_i}$, with k being the k^{th} driver belonging to player i. We let $\theta^t := \theta_i^k$ be the information revealed at time t, identifying the location of the requesting driver k and the player i it belongs to. We denote with $\Theta = \{\theta_i : i \in \mathcal{N}\}$ the set of player's information vectors and we define a state x_t as the vector that describes the history of decisions and of revealed information such that $x_t = (\theta^0, ..., \theta^t, a_0, ..., a_{t-1})$. We define policy $\pi = (\pi_1, ..., \pi_t)$ as the sequence of decisions made in each epoch, with $a_t := \pi_t(x_t)$, being the station assigned to driver θ^t by the principal. Each assignment decision induces an immediate cost corresponding to the time required by the associated driver to reach the station, $t_{\theta^t, \pi_t(x_t)}$. At time t, a player's cost is the sum of all of its driver's travel times or penalties. A player's cost sequentially increases until the planning horizon ends.

From the principal's perspective, the sequential station assignment problem can be modeled as an MDP, whose solution policy corresponds to the policy π. Accordingly, we introduce an MDP defined by a policy π, the state space X, and transition functions $p(x'|x, a)$ that describe the probability that the system in state x will transition to post-decision state x' after having taken action a. As previously introduced, a state x describes the current stations' assignment of existing drivers in the system and an action a represents the assignment decisions realized by the principal.

We define the immediate cost $d_t^i(x_t, a_t)$ for each player i as

$$d_t^i(x_t, a_t) = \begin{cases} c(\theta^t, a_t), & \text{if } i = \hat{i} \\ 0, & \text{else} \end{cases} \tag{3.17}$$

with $\hat{i}$ being the player associated to request θ^t, using the station assignment cost $c(\theta^t, a_t)$ as defined in (3.3). Let the total immediate cost be $d_t(x_t, a_t) = \sum_{i \in \mathcal{N}} d_t^i(x_t, a_t)$. Further, we

denote by

$$d^i_{<T}(\theta, \pi) = \sum_{t=0}^{T} d^i_t(x_t = (\theta^0, ..., \theta^t, a_0, ..., a_{-1}), \pi_t(x_t)) \tag{3.18}$$

the accumulated cost for player i from $t = 0$ until T, with reported information θ. We can then define the MDP value function V^π as the expected value of the summed costs over all decision epochs

$$V^\pi(x_t) = E_\pi[\sum_{\tau=t}^{T} d_t(x_\tau, \pi_\tau(x_\tau))] \ , \tag{3.19}$$

which we can also recursively express as

$$V^\pi(x_t) = d_t(x_t, \pi_t(x_t)) + \sum_{x_{t+1}} p(x_{t+1}|x_t, \pi_t(x_t))V^\pi(x_{t+1}) \ . \tag{3.20}$$

The objective of the principal is then to find a policy π that minimizes the expected cost value $V^\pi(x_0)$ over the planning horizon, with x_0 being the initial system state. The realized cost for each individual player corresponds to the summed costs over the entire planning horizon for its drivers, formally

$$v^i(\theta^i, a_{\leq T}) = \sum_{\tau=0}^{T} d^i_\tau(x_\tau, a_\tau) \ . \tag{3.21}$$

Similar to the offline setting in Section 3.1, we assume that a player i may misreport her information θ^k_i and we refer to the reported information as $\hat{\theta}$, used by the planner to decide on the next action $\hat{a}_t$. Then, the immediate cost induced by action $\hat{a}_t$ corresponds to the distance from the *actual* driver location to the next decided station, and depends on the *actual* location of the driver θ^t and the action chosen. Accordingly, $d^i_t((\theta^0, ..., \hat{\theta}^t, \pi), \hat{a}_t) = c(\theta^t, \hat{a}_t)$ when i reports information $\hat{\theta}^t$ with *actual* information θ^t.

Analogous to Parkes et al. (2004), we define the mechanism's prices as the difference between the sum of realized costs of all players but i and the optimal realized cost with perfect information without player i. Each player pays the price at the end of a given planning horizon. Formally, the pricing rule for player i is

$$p^i(\theta, \pi) = -\sum_{j \neq i} \sum_{t=0}^{T} d^j_t((\theta_{<t}, a_{<t-1}), \pi_t(x_t)) + OPT_{\theta_{-i}} \ , \tag{3.22}$$

with $OPT_{\theta_{-i}}$ being the optimal assignment cost that can be obtained under full information without i being in the system, and π being the optimal MDP-policy, such that

$$OPT_{\theta_{-i}} = \sum_{j \neq i} d^j_{<T}(\theta_{-i}, \pi) \ . \tag{3.23}$$

Then, the mechanism $\mathcal{M}_{\text{on}} = (\Theta, p, \pi)$ defined by the players' information Θ, the pricing rule $p = (p_i)_{i \in \mathcal{N}}$ and the assignment decisions policy π constitutes the delayed VCG mechanism. The pricing rule defined in Equation 3.22 makes $\mathcal{M}_{\text{on}}$ Bayesian-Nash Incentive-compatible, i.e.,

$$E_{\tau > t}[v^i(\theta^i, a_{\leq T}) - p^i(\theta, \pi)] \leq E_{\tau > t}[v^i(\theta^i, \hat{a}_{\leq T}) - p^i(\hat{\theta}, \pi)] \ \ \forall \hat{\theta}^t \ \forall t \tag{3.24}$$

with $\hat{a}_{\leq T} = (\pi_0(x_0), ..., \pi_t(\hat{x}_t), ..., \pi_T(\hat{x}_T))$. Here, $\hat{x}_\tau = (\hat{\theta}_{<\tau}, \hat{a}_{\leq \tau - 1})$ with $\hat{\theta}_{<\tau} = (\theta^0, ..., \hat{\theta}^{t-1},$

$...., \hat{\theta}^\tau)\ \forall \tau \in [t, T]$. We assume that $\hat{\theta}^\tau = \theta^\tau$ if player $j \neq i$ reports in epoch τ, i.e., all players but i truthfully report their information. Equation 3.24 ensures that the expected player's payoff cannot be better when the player reports false information $\hat{\theta}^t$, instead of θ^t.

Proposition 5. *Mechanism $\mathcal{M}_{on}$ is Bayesian-Nash incentive-compatible.*

Proof. We refer to Appendix A for a proof of Proposition 5. □

Similar to the VCG mechanism (see Section 3), we show that the delayed VCG mechanism $\mathcal{M}$ can be extended to account for weighted players. We can adapt the immediate cost by weighting players, and introducing the new immediate cost

$$\tilde{d}_t(x_t, a_t) = \sum_{i \in \mathcal{N}} w_i \cdot d_t^i(x_t, a_t) \tag{3.25}$$

with $w_i \geq 1\ \forall i \in \mathcal{N}$. Note that $v^i(\theta^i, a_{\leq T})$ remains unchanged. We accordingly update the pricing rule as

$$\tilde{p}^i(\theta, \pi) = -\sum_{j \neq i} \sum_{t=0}^{T} \frac{w_j}{w_i} d_t^j((\theta_{<t}, a_{<t-1}), \pi_t(x_t)) + \tilde{OPT}_{\theta_{-i}} \tag{3.26}$$

where $\tilde{OPT}_{\theta_{-i}} = \sum_{j \neq i} \frac{w_j}{w_i} d^j_{<T}(\theta_{-i}, \pi)$.

We define $\tilde{\mathcal{M}}_{on} = (\Theta, \tilde{p}, \pi)$ as the weighted delayed VCG mechanism, and show that $\tilde{\mathcal{M}}_{on}$ is still in-expectation incentive-compatible.

Proposition 6. *Mechanism $\tilde{\mathcal{M}}_{on}$ is Bayesian-Nash incentive-compatible.*

Proof. We refer to Appendix A for a proof of Proposition 6. □

3.2.2. Online allocation policy

In an online setting, each decision stage is triggered by a new driver's request, and the principal assigns a charging station to the driver reported by a player based on the online allocation policy. Finally, the mechanism computes prices at the end of the planning horizon, which terminates after a fixed number of considered requests. In the following, we derive two online allocation policies to solve the underlying MDP of the delayed VCG mechanism $\mathcal{M}_{\text{on}}$. The first policy *greedy* allocates the closest available station to the requesting driver. The second policy *data-driven* bases on a data-driven algorithm (cf. Garatti and Campi 2022), and learns the policy parameterization based on historical charging requests. As both policies are suboptimal in practice, the truthfulness of the mechanism cannot be guaranteed in theory. We notice that if one wants to formally ensure truthfulness, one could extend the mechanism similar to the second chance mechanism as introduced in Nisan and Ronen (2007). In this case, players have a chance to report a different information only if it improves their utility, and accordingly the expected social cost.

Greedy policy: The greedy policy is a deterministic policy that assigns a requesting driver to the closest available station, such that $\pi_t(x_t) = \arg\min_{v \in \bar{\mathcal{V}}} t_{\theta^t,v}$, with θ^t being the location of the requesting driver in state x_t, and $\bar{\mathcal{V}}$ being the set of stations that have not been already assigned to preceding drivers.

Data-driven policy: The data-driven policy is a probabilistic policy, i.e., a policy that chooses its action based on a probability distribution. To sample actions, we use a parametric data-driven online algorithm to determine the action taken at each decision stage. Specifically, the algorithm parameters denote with which probability to take a specific action. We learn the parameterization of this algorithm offline, based on a large set of training input sequences, each consisting of charging requests of a fixed length.

We derive the data-driven algorithm A as follows. Formally, we let $\mathcal{I}$ be the set of all possible scenarios I where $I = [\theta^0, ..., \theta^t, ..., \theta^T]$, with $T+1$ being the input sequence length and $\theta^t \in \Theta$ being the location of a driver in position t, such that $\mathcal{D}_I$ corresponds to the set of drivers k contained in I. Our goal is to find an algorithm A that minimizes the competitive ratio α for all possible input sequences $I \in \Delta$, such that

$$\begin{aligned} A^* = \arg\min_A \alpha \\ & ON(I_l, A) \leq \alpha OPT(I_l) \forall I \in \Delta, \\ & \sum_{v \in \mathcal{V} \cup \mathcal{V}_0} x_{kv} = 1 \ \forall k \in \mathcal{D}_I, \ \forall I \in \Delta, \\ & \sum_{k \in \mathcal{D}_I} x_{kv} \leq 1 \ \forall v \in \mathcal{V}, \ \forall I \in \Delta \\ & x_{kv} \in [0,1] \forall k \in \mathcal{D}_I, \ \forall v \in \mathcal{V} \ \forall I \in \Delta \ , \end{aligned} \tag{3.27}$$

where $ON(I, A)$ corresponds to the online solution obtained with A for a given scenario l and $OPT(I)$ corresponds to the offline solution obtained for l. The offline solution $OPT(I)$ corresponds to the minimum cost assignment of the drivers in I to charging stations in $v \in \mathcal{V} \cup \mathcal{V}_0$, as

$$OPT(I) = \min_{x_{kv} \forall k \in \mathcal{D}_I \ \forall v \mathcal{V} \cup \mathcal{V}_0} \sum_{v \in \mathcal{V} \cup \mathcal{V}_0} \sum_{k \in \mathcal{D}_I} x_{kv} \cdot c(k,v) \tag{3.28}$$

$$\sum_{v \in \mathcal{V} \cup \mathcal{V}_0} x_{kv} = 1 \ \forall k \in \mathcal{D}_I \tag{3.29}$$

$$\sum_{k \in \mathcal{D}_I} x_{kv} \leq 1 \ \forall v \in \mathcal{V} \tag{3.30}$$

$$x_{kv} \in \{0,1\} \ \forall k \in \mathcal{D}_I, \ \forall v \in \mathcal{V} \tag{3.31}$$

$$c(k,v) = \begin{cases} \bar{\beta}, & \text{if } v = v_0 \vee \gamma(k,v) \geq \bar{S} \\ t_{k,v}, & \text{else.} \end{cases} \tag{3.32}$$

Here, the objective (3.28) is to find an assignment of drivers to stations that yields minimal cost. Each driver must be assigned to at least one physical or virtual station (3.29), and

at most one driver can be assigned to a physical station (3.30). A driver assignment cost corresponds to the driving time from its current location to the physical station or to a penalty cost if the station is further away than $\bar{S}$ or if the station is virtual (3.32).

To mitigate the computational burden, we chose A to lie on a parametric class and relax the integer constraint on the decision variables x. We then learn the algorithm A^* based on L i.i.d. sampled training input sequences $I_0, ..., I_L$ of the uncertainty set $\mathcal{I} \subset \Delta$. Accordingly, the learning objective becomes

$$\begin{aligned} A^* = &\arg\min_{A} \alpha \\ &ON(I_l, A) \leq \alpha OPT(I_l) \forall I_l \in \mathcal{I},\ l \in [L] \\ &\sum_{v \in \mathcal{V} \cup \mathcal{V}_0} x_{kv} = 1\ \forall k \in \mathcal{D}_{I_l},\ \forall I_l \in \mathcal{I},\ l \in [L] \\ &\sum_{k \in \mathcal{D}_{I_l}} x_{kv} \leq 1\ \forall v \in \mathcal{V},\ \forall I_l \in \mathcal{I},\ l \in [L] \\ &x_{kv} \geq 0 \forall k \in \mathcal{D}_{I_l},\ \forall v \in \mathcal{V}\ . \end{aligned} \tag{3.33}$$

We introduce the parameterization vector of the algorithm A, as $\vec{p}_{\theta,t} \in \mathbf{R}^{|\mathcal{V}|}$ for each (θ, t)-pair, such that $0 \leq p_{\theta,t,v} \leq 1$ represents the probability for a driver in location θ with position index t in the input sequence to be assigned to station v. Thus, we derive the algorithm A with parameterization $\vec{p}^*_{\theta,t}$ for all (θ, t)-pairs, as follows

$$A = \arg\min_{p_{\theta,t,v} \forall \theta \in \Theta, t \in [0,...,T], v \in \mathcal{V}} \alpha \tag{3.34}$$

$$\sum_{t=0}^{T} \sum_{v \in \mathcal{V}} p_{\theta^t,t,v} \cdot c(\theta^t, v) \leq \alpha \cdot OPT(I_l)\ \forall l \in [L]\ \theta_t \in I_l \tag{3.35}$$

$$\sum_{v \in \mathcal{V}} p_{\theta,t,v} = 1\ \forall \theta \in \Theta, \forall t \in [0, ..., T] \tag{3.36}$$

$$\sum_{t=0}^{T} \sum_{\theta \in \Theta} p_{\theta,t,v} \cdot n(\theta, t, \mathcal{I}) \leq 1\ \forall v \in \mathcal{V}\ , \tag{3.37}$$

$$\alpha \in \mathbf{R}^+ \tag{3.38}$$

with $n(\theta, t, \mathcal{I})$ being the likelihood that a driver is located in θ with position index t, estimated based on the occurrence of such combination (θ, t) among all input sequences $I_l,\ l \in [L]$. The Objective (3.34) is to find a parameterization that minimizes the competitive ratio α, i.e., the ratio between the optimal cost and the cost obtained with the parametric algorithm as introduced in (3.27). Constraint (3.35) ensures that the ratio between the expected online solution and the optimal offline solution is lower than the minimized competitive ratio α. Constraint (3.36) enforces that we compute a discrete probability distribution, while Constraint (3.37) ensures that we do not assign (in-expectation) more than one driver to a station.

We apply this data-driven policy in state $x_t = (\theta^0, ..., \theta^t, a_0, ..., a_{t-1})$ as follows. First, we exclude already allocated stations, i.e., $v \in \{a_0, ..., a_{t-1}\}$, from the possible stations

assignment by setting the probability to be selected to 0, such that $p_{\theta,t,v} = 0$, $\forall\theta$ st. $\theta = \theta^t$. Then, we scale the assignment probabilities of the remaining candidate stations to preserve the discrete probability distribution as

$$p'_{\theta,t,v} := \frac{p_{\theta,t,v}}{\sum_{v\in\mathcal{V}} p_{\theta,t,v}\cdot\delta_v} \forall v \in \mathcal{V} \ ,$$

with $\delta_v = 0$ if $v \in \{a_0, ..., a_{t-1}\}$. Finally, we select a station v for the driver in location θ based on the probabilities $p'_{\theta,t,v}$.

In a setting with weighted players, optimizing weights requires to know each player's payoff resulting from a selfish behavior. However, as requests are unseen, we do not know a player's selfish payoff ahead of time and thus cannot optimize weights a-priori. To remedy this issue in practice, one could derive the weights a-posteriori, i.e., after a planning horizon has terminated, and apply them for the next planning horizon. In this case, one needs to additionally ensure that the assignment costs account for the players' weights, as $c(k, v, i) = w_i \cdot c(k, v)$. Consequently, the parameterization needs to be indexed on the players, such that $p_{\theta,t,v,i}$ represents the probability that a driver in location θ at requesting time t and belonging to player i should be assigned to station v.

4. Experimental design

To analyze the impact of coordinating the charging demand at drivers' and platforms' levels, we derive real-world test instances based on the charging station network for the city of Berlin, Germany (cf. Figure 1).

We consider three unweighted navigation platforms, i.e., players. The ratio of available stations and requesting drivers is the main factor impacting our results. We accordingly vary the total number of drivers in the system $N \in [2, .., 40]$, the search radius ($\bar{S} \in \{1000, 2000\}$ meters) and the radius of the circular area in which all drivers depart ($s^r \in \{300, 700, 1100\}$ meters). We set the penalty $\bar{\beta} = 120$ min, such that failing the search for a driver corresponds to a delay of two hours.

To account for the navigation platforms' heterogeneity, we vary the players' imbalance, with respect to the number of drivers managed by each platform as shown in Table 1. We compare three driver distribution scenarios. In the *small* scenario, one player accounts for 20% of the total demand, whereas in the *big* scenario, one single player accounts for 50% of the demand. In both scenarios, the other two players share the remaining demand equally. The *equal* scenario represents a homogeneous player scenario, with equal distribution of drivers between players.

For both offline and online charging demand, we benchmark the *VCG*-based assignment against two naive assignment strategies. In *D-SELF*, each EV driver visits the closest station in its vicinity, whereas in *P-SELF*, navigation platforms compute the cost-minimum assignment of stations to drivers with respect to their own charging demand. We compare all three settings, i.e., *VCG*, *D-SELF*, and *P-SELF*, with respect to the social cost S_g,

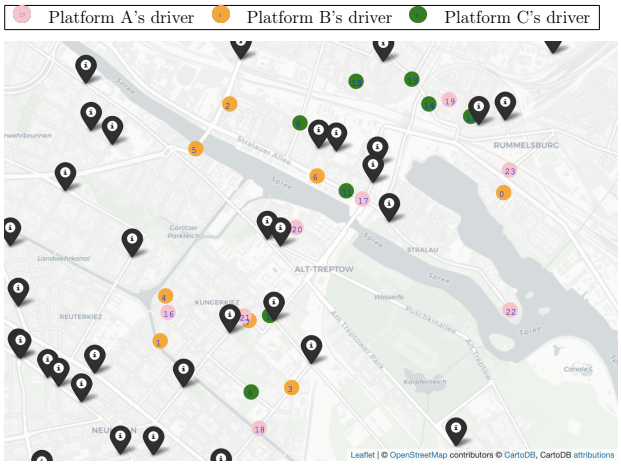


Figure 1: Charging station and driver distribution in a part of Berlin

Table 1: Driver distribution scenarios

scenario	Navigation platform		
	A	B	C
big	25.00%	25.00%	50.00%
equal	33.33%	33.33%	33.33%
small	40.00%	40.00%	20.00%

i.e., the sum of each player's drivers to stations assignment cost, which reflects end-users' satisfaction. To analyze the benefits of *VCG* pricing, we further compare each player's payoff, including *VCG* prices when applicable. To complement the offline analyses, we analyze the impact of optimally weighting players, such that all players benefit from *VCG* when applying the weighted *VCG* pricing scheme. Here, we solve the optimization problem introduced in Equations (3.13)-(3.16), and set the maximal weight value to $W = 10$.

For the online charging demand setting, we assume that for both *D-SELF* and *P-SELF*, there is a latency between the time when the drivers stop at the station and the time the station's availability status is up-to-date for the succeeding drivers. We consider a latency of $\tau = 3$ minutes, i.e., the time for the driver to stop and start charging and for the system to update the availability information. For *P-SELF* the latency only concerns the other player's drivers, as the player knows which stations were assigned to its drivers. We further assume that requests arise every Δ_t minute with $\Delta_t \in \{0.5, 1.5, 2.5\}$ minutes.

For the data-driven parameterization, we limit the computational burden by identifying a set of at maximum 40 possible locations a driver can depart from ($|\theta| \leq 40$). In most instances, candidate departing locations cover all vertices in the corresponding road network, i.e., roads junctions. To evaluate the stochastic data-driven online policy on a test instance with N drivers, we randomly sample $n = 500$ start locations for each driver, and compute the simulated estimates of the social cost and each player's individual payoff.

5. Results

In the following, we first detail our results for an offline charging demand setting in Section 5.1, to obtain an upper bound on the system improvement as well as to understand the interactions between navigation platforms with different driver shares. We further discuss the impact of optimally weighted platforms in this setting in Section 5.2. We then discuss our results for an online charging demand setting in Section 5.3 and compare it to the offline

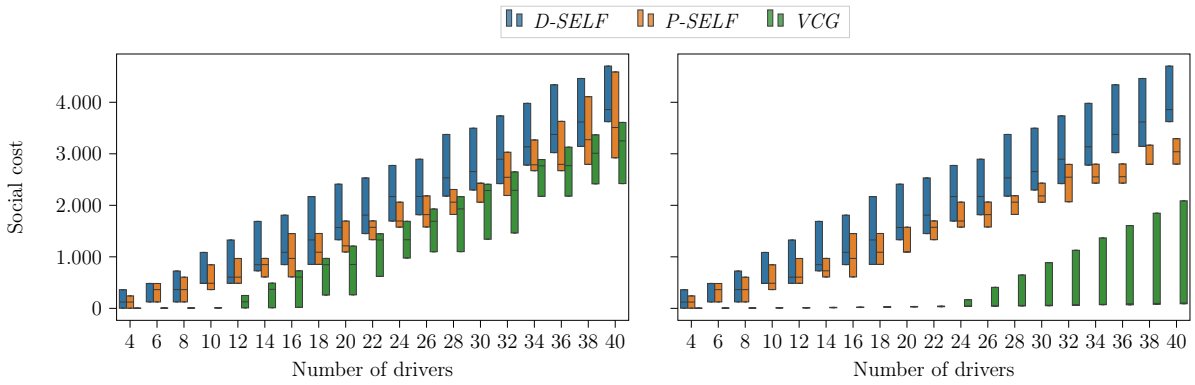
benchmark.

5.1. Offline allocation results

Figure 2 shows the distribution of the social cost for all three strategies (*P-SELF*, *D-SELF*, *VCG*) depending on the number of drivers in the system for both small ($\bar{S} = 1000$m) and large ($\bar{S} = 2000$m) search radii. As can bee seen, selfish navigation service platforms can increase the overall user satisfaction by optimally balancing their own charging demand (*P-SELF*) compared to selfish EV drivers taking greedy visit decisions (*D-SELF*). However, VCG pricing further reduces the social cost significantly. Our results show that the benefit of platform coordination increases when the ratio between the total number of charging stations and the total number of drivers in the system increases. In contrast, when the likelihood of visit conflicts is high, i.e., many drivers depart within small distance or with small search radius, the coordination improvement decreases. However, coordination remains necessary in these cases as selfish platform optimization becomes as bad as individual greedy driver decisions.

Result 1. *Coordinating platforms with VCG pricing decreases the social cost by up to 52% compared to an allocation obtained with selfish platforms.*

Figure 3 shows all three platforms' mean payoffs depending on the number of drivers and for each driver distribution scenario. To compare platforms with a heterogeneous driver share, we normalize each platform's payoff by its number of managed drivers. As can be seen, VCG pricing significantly decreases a platform's payoff on average by 27% (*equal*), respectively 21% (*big*), and 26% (*small*) compared to the payoff obtained for platform-optimized allocations (*P-SELF*). In case of a high charging demand, i.e., for more than 20 drivers departing in a small vicinity and with lower search times, the induced VCG prices increase a platform's payoff compared to an uncoordinated setting. Focusing on the



a) $\bar{S} = 1000$ m b) $\bar{S} = 2000$ m

Figure 2: Benefit of VCG pricing on the social cost

Results are averaged over all values of s^r with $s^r\{300, 700, 110\}$ m.

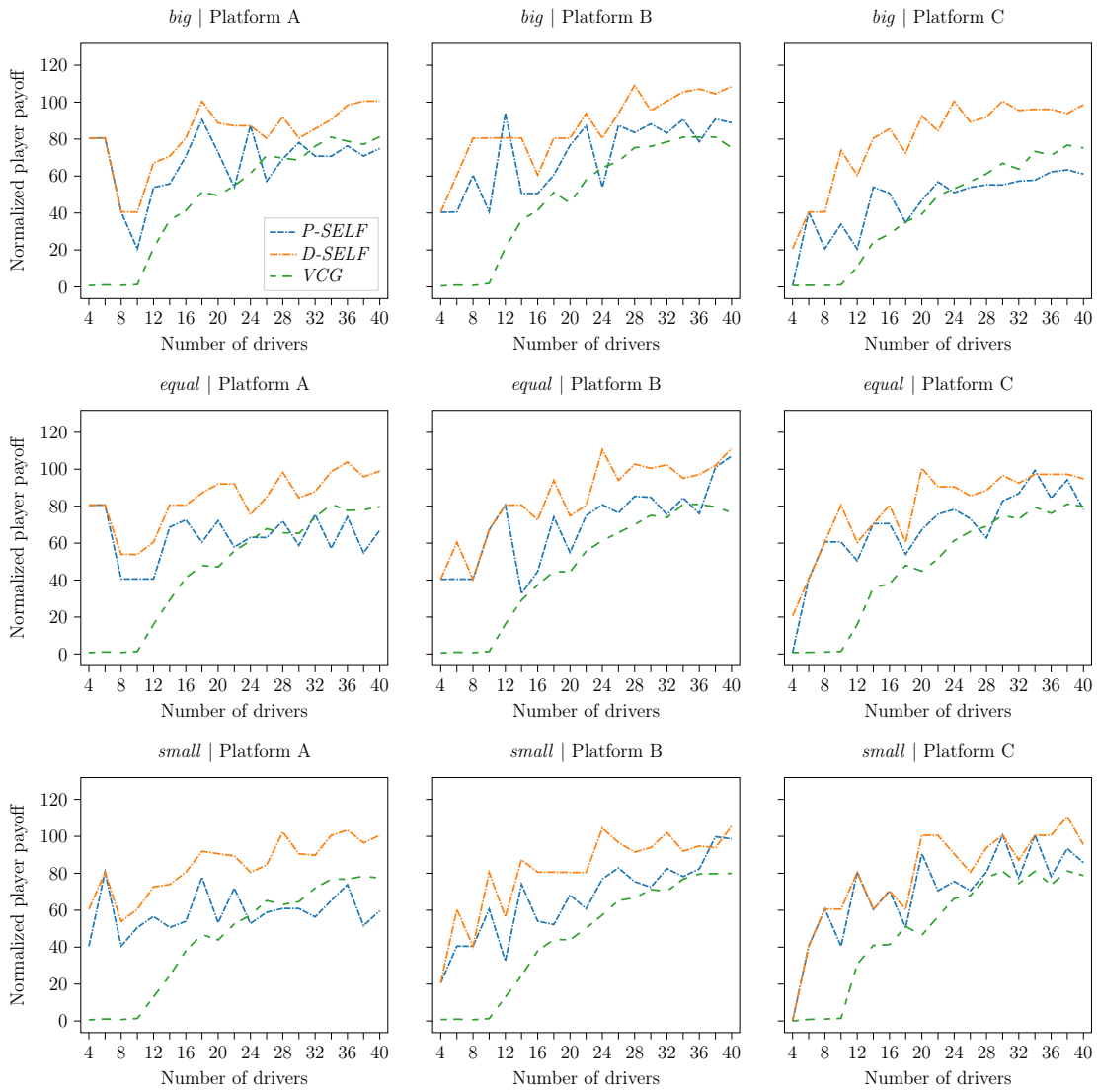


Figure 3: Impact of the distribution scenario on each platform's payoff for *VCG*, *P-SELF*, and *D-SELF*

impact of the driver distribution, our results highlight two effects. Utilizing VCG pricing, a platform's payoff decreases if its share of drivers in the system increases (see e.g., Platform C, in Figure 3). Contrarily, the improvement obtained with *VCG* pricing compared to a selfish assignment is larger for a platform with proportionally less drivers. In some cases, e.g., for $\bar{S} = 1000$ m and a very large number of drivers, *VCG* does not outperform *P-SELF* for a platform that manages the highest share of drivers. If a platform manages most drivers, then the subsystem composed of other platforms' drivers is smaller such that the platform's impact on it is low, which decreases the large platform's price. In this case, the impact of other platforms on the platform with the largest drivers share decreases as well, such that the benefits of coordination do not always compensate the additional VCG prices induced by coordination.

Result 2. *A navigation platform with a low share of drivers (nearly) always benefits from VCG pricing compared to P-SELF. In contrast, a platform with a higher share of drivers is*

sometimes better off not participating in a VCG setting.

Figure 4 compares the mean payoffs obtained for each platform and distribution scenario depending on the number of drivers. In line with Figure 3, results show that a platform with more drivers gets, proportionally to the number of drivers managed, a lower payoff compared to platforms with less drivers. As can be further seen, the impact of a heterogeneous driver share is bigger for a large search radius ($\bar{S} = 2000$ meters) compared to a small search radius ($\bar{S} = 1000$ meters). In summary, if a platform participates in a *VCG* setting, its payoff will inversely decrease with the number of drivers it manages in the system, but the benefit of participating in *VCG* compared to *P-SELF* with a lot of drivers will decrease, too. If a platform has few drivers, it is better off participating in the system.

Result 3. *If participating in a VCG coordinated system, a navigation platform can decrease its cost by increasing its drivers share. However, the relative improvement obtained with VCG compared to P-SELF decreases in this case.*

Figure 5.a shows the average driving time to an available station for any driver in the system, while Figure 5.b shows the average station assignment success rate for a platform for both $\bar{S} = 1000$ m and $\bar{S} = 2000$ m. Analyzing Figures 5.a & 5.b, our results show that coordination through VCG pricing slightly increases the average driven search time needed by its drivers to reach an available station for each platform compared to a selfish or a greedy behavior. However, lower driver search times in selfish settings (*P-SELF&D-SELF*) come

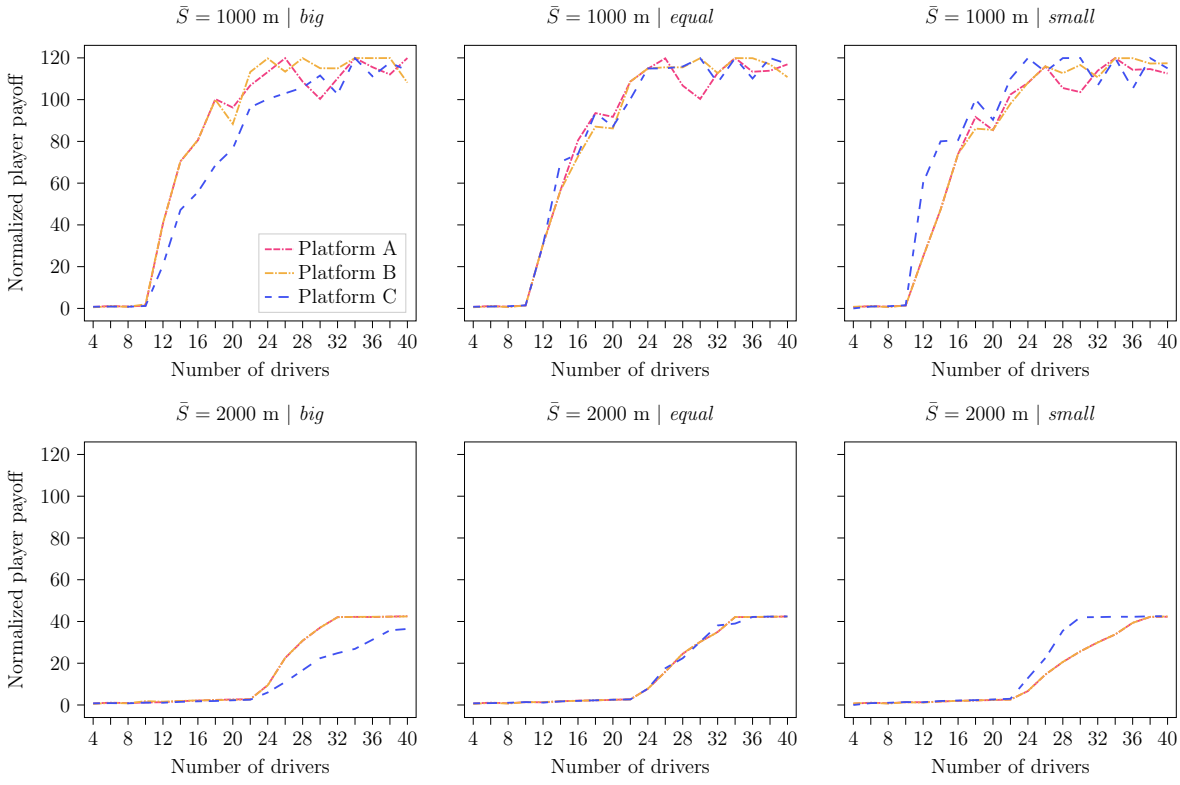
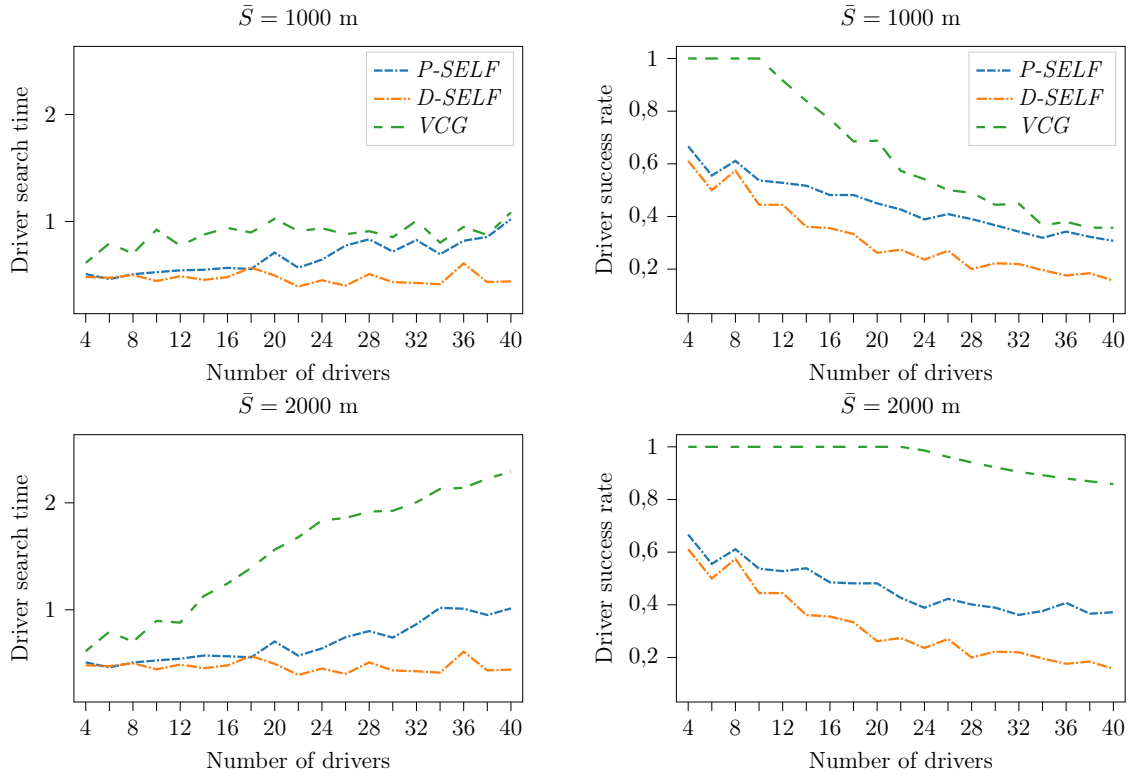


Figure 4: Impact of the distribution scenario on each platform's payoff



(a) Search times (b) Success rates

Figure 5: Impact of VCG pricing on average drivers search times and success rates

at the expense of significantly lower success rates, which highlights the need of coordination to decrease station visits conflicts.

Result 4. *Platform coordination mildly increases the average driving time of drivers in the system by 32 seconds on average, compared to local coordination (P-SELF), and by 47 seconds, compared to uncoordinated drivers (D-SELF). These longer searches significantly reduce the number of station visit conflicts, leading to a success rate increase of 33% compared to P-SELF, and of 49% compared to D-SELF.*

5.2. A-priori weights optimization

In the following, we analyze the impact of optimizing weights prior to applying offline weighted VCG pricing by solving the optimization problem described in Section 3.1.2. Table 2 summarizes the number of instances in which unweighted VCG pricing ("VCG beneficial") benefits all platforms, i.e., each platform obtains lower cost with VCG than without, as well as the number of instances in which VCG pricing benefits all platforms only if players are optimally weighted ("Weighted VCG beneficial"). Furthermore, the table shows the number of instances for which optimal weights could not be derived ("Infeasible weights"), and the number of instances for which the weights optimization problem could not be solved within the computational time limit ("Not solved") of 120 minutes. We differentiate results between low ($\bar{S}$ = 1000 m) and high ($\bar{S}$ = 2000 m) search radius.

Table 2: Number of test instances that improve with weighted VCG over unweighted VCG, respectively are infeasible, or unsolvable

	Number of instances			
$\bar{S}$	VCG beneficial	Weighted VCG beneficial	Infeasible weights	Not solved
1000	23	14	83	51
2000	125	5	24	17
total	148 (43%)	19 (6%)	107 (31%)	68 (20%)

As can bee seen, weighted VCG pricing increases the total number of instances in which all platforms benefit from *VCG* participation from 43% to 49%. There remain 31% of test instances that yield no feasible weights and 20% of instances for which the weights optimization problem could not be solved within the limited computational time. Focusing on test instances that yield infeasible weights, either unweighted *VCG* already benefits all platforms, or the ratio of available stations per driver is too low to distribute all drivers across stations. In the latter case, prices will be high for any weights values for at least one platform, which accordingly results in higher payoff with *VCG* than with *P-SELF*. In a limited number of cases (6% of the total number of test instances), the ratio of stations per driver is low enough such that unweighted *VCG* does not benefit all platforms, but large enough such that a minor reallocation of stations to platforms ensures that all platforms benefit from *VCG* over *P-SELF*. Here, platforms who initially did not benefit from *VCG* have larger weights than other platforms. Moreover, we observed that the range of improvement through weights optimization depends on the search radius, with 60% of the instances benefiting from weighted VCG in a low search radius ($\bar{S} = 1000$ m) setting against only 4% benefiting in a large search radius ($\bar{S} = 2000$ m) setting. In the latter case, initial benefits of VCG pricing are higher for a larger search radius ($\bar{S} = 2000$ m) due to the higher number of available stations per driver, which in turn decreases the improvement potential.

Result 5. *Optimizing platforms' weights increases the number of instances in which all platforms benefit from participating in VCG from 43% to 49%.*

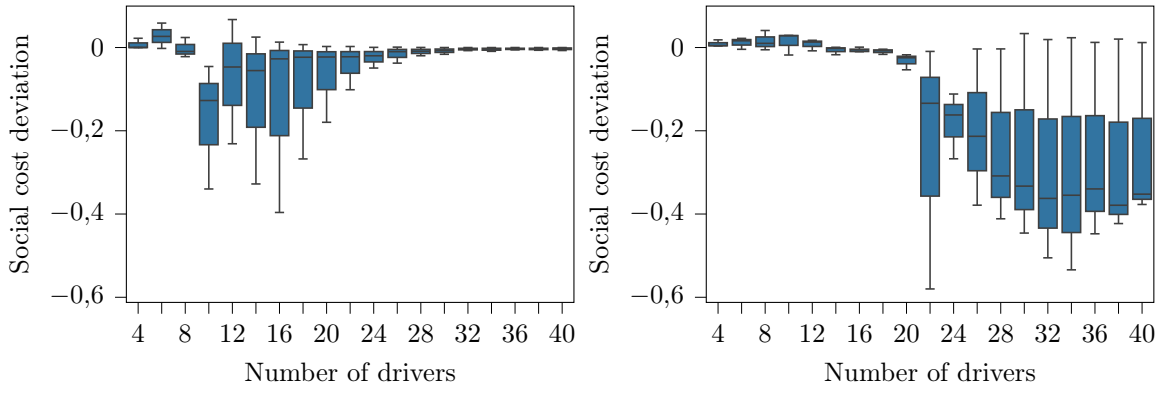
To further detail the impact of weighting players, Figures 10a-10c in Appendix B show the averaged payoff per platform and scenario in both unweighted and weighted cases, depending on the total number of drivers. Platform C appears to benefit the most from the weight optimization in the *small* scenario, when managing only 20% of the total number of drivers. Weighting platforms slightly benefits the system by decreasing a platform's payoff. However, results show little differences between weighted platforms' payoffs and unweighted platforms' payoffs, and no clear trend on the weights values depending on the scenario. Accordingly, we focus on the unweighted platforms setting in the following analyses.

5.3. Online allocation results

The following section analyzes the benefit of *VCG* coordination with online charging requests. Preliminary studies (cf. Appendix B) show a similar effect of driver imbalance on a platform's payoff in an online compared to an offline setting. Accordingly, we focus the results discussion on the balanced platforms scenario *equal* to evaluate the performances of *VCG* and compare the benefits with respect to a perfect-information benchmark (*OFF*).

Figure 6 shows the social cost deviation between the data-driven online policy (*VCG*-dd) and the greedy online policy (*VCG*-greedy), computed as follows $\Delta(\hat{\alpha}) = {}^{(\hat{\alpha}_{\text{dd}} - \hat{\alpha}_{\text{greedy}})}/_{\hat{\alpha}_{\text{greedy}}}$, with $\hat{\alpha}_{\text{dd}}$, respectively $\hat{\alpha}_{\text{greedy}}$, being the realized social cost for the data-driven, respectively greedy, online policy. Figure 7 compares the search cost distribution obtained with *VCG*-dd and *VCG*-greedy online policies against the perfect-information benchmark (*OFF*), per number of drivers, for a small ($\bar{S} = 1000$ meters) and large search radius ($\bar{S} = 2000$ meters).

Our results (cf. Figure 6) show an improvement of the data-driven policy over the greedy allocation policy in an online setting, due to a better anticipation of possible visit conflicts of the data-driven policy. For $\bar{S} = 1000$, the averaged improvement over all values of s^r is 23% in the best case ($N = 10$ drivers), whereas for $\bar{S} = 2000$ it increases to 34% ($N = 22$ drivers). As can bee seen, the benefit of a data-driven policy is highest for a small number of drivers in case of a small search radius ($\bar{S} = 1000$ meters) but highest for a large number of drivers in case of a large search radius ($\bar{S} = 2000$ meters). In the former case, the social outcome for a higher number of drivers is generally bad due to a high number of unavoidable station visit conflicts. Here, the bottleneck significantly limits the improvement potential, such that the two online policies and the perfect-information setting (*OFF*) yield equally bad outcomes (cf. Figure 7). In contrast, the likelihood of station visit conflicts is low for a small number of drivers in the latter case, such that a greedy policy performs already nearly as good as a perfect-information setting allocation (*OFF*). In this case, there will always be enough available stations to allocate to incoming drivers, which will not increase the social cost with penalty costs.



a) $\bar{S} = 1000$ m b) $\bar{S} = 2000$ m

Figure 6: Social cost deviation $\Delta(\hat{\alpha})$ between the *VCG*-dd and *VCG*-greed online policies

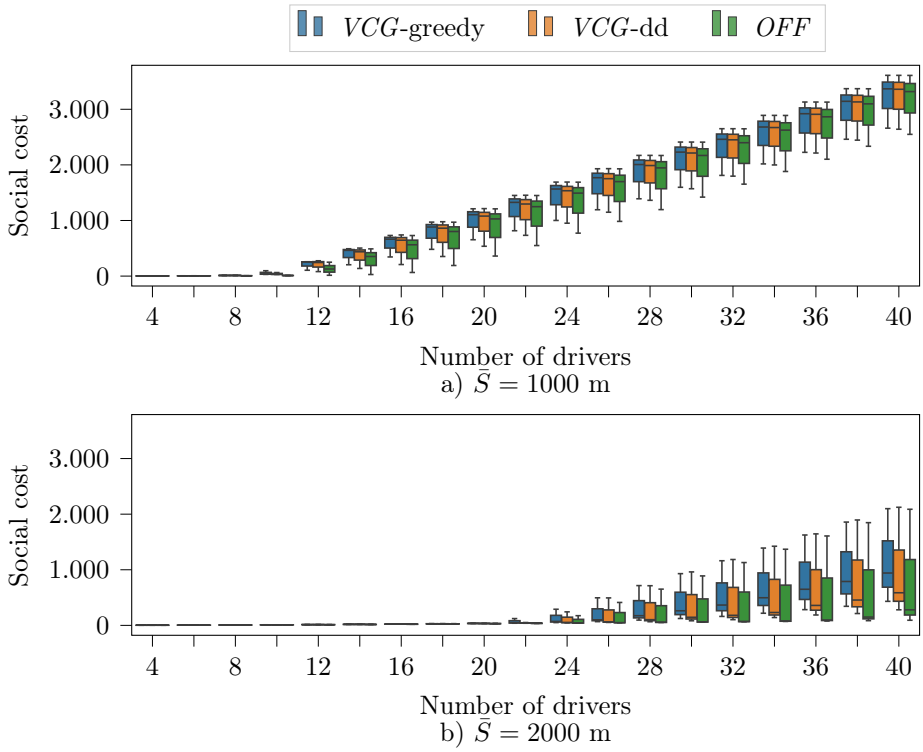


Figure 7: Social cost obtained with the *VCG*-dd and *VCG*-greedy online policies, and the perfect-information benchmark *OFF*

Results are aggregated over all values of s^r with $s^r\{300, 700, 110\}$ m.

Result 6. *A data-driven online assignment policy improves the social outcome when the improvement potential is highest, i.e., for a small number of drivers with small search radius or for a large number of drivers with large search radius.*

Result 7. *A data-driven online assignment policy decreases the social cost on average by 14% given a large search radius, and yields a maximum reduction of 55% for* $s^r = 700$ *meters and* 32 *drivers, compared to a greedy online assignment policy.*

Table 3 shows the deviation of the realized social cost obtained with *P-SELF*, *D-SELF*, and *VCG*-greedy compared to the *VCG*-dd online policy. For *P-SELF* and *D-SELF*, we detail results for a small time span between two driver requests ($\Delta_t = 0.5$ minutes), an average time span ($\Delta_t = 1.5$ minutes) and a large time span ($\Delta_t = 2.5$ minutes). For both *VCG*-greedy and *VCG*-dd online policies, results are independent of Δ_t as the central decision-maker is aware of charging station' availability. As can be seen, both coordinated online assignment policies significantly outperform the two naive benchmarks. In line with Figure 6, results further show the superiority of the online *VCG*-dd policy over the online *VCG*-greedy policy, particularly in cases with widespread drivers, i.e., for a large search radius ($\bar{S} \geq 2000$ m) or for a large departing area ($s^r \geq 700$ m). Our results further show that the benefit of coordinating platforms increases with the requests frequency and are highest for $\Delta_t = 0.5$ minutes. We note that for a lower request frequency, i.e., $\Delta_t = 2.5$ minutes, a platform does not benefit from locally coordinating its charging demand. Here,

Table 3: Social cost deviation for *P-SELF*, *D-SELF*, and *VCG*-greedy compared to *VCG*-dd

S	s^r	*VCG*-greedy	*P-SELF*			*D-SELF*		
			$\Delta_t = 0.5$	$\Delta_t = 1.5$	$\Delta_t = 2.5$	$\Delta_t = 0.5$	$\Delta_t = 1.5$	$\Delta_t = 2.5$
1000	300	-0.17%	15.6%	12.2%	10.7%	34.4%	11.9%	6.20%
	700	0.95%	13.3%	9.23%	7.63%	19.0%	8.81%	5.31%
	1100	4.23%	37.6%	25.0%	20.5%	51.9%	24.3%	15.1%
2000	300	-0.19%	193%	167%	155%	259%	171%	114%
	700	77.5%	994%	778%	670%	1199%	780%	536%
	1100	45.6%	1571%	1185%	1004%	1895%	1195%	811%

The table shows the search cost deviation computed as $\Delta(\hat{\alpha}) = {}^{(\hat{\alpha}_{\text{op}} - \hat{\alpha}_{\text{dd}})}/_{\hat{\alpha}_{\text{dd}}}$, with $\hat{\alpha}_{\text{dd}}$, being the realized social cost for the data-driven online policy and $\hat{\alpha}_{\text{op}}$ being the realized social cost for *P-SELF*, *D-SELF*, and *VCG*-greedy respectively.

the realized social outcome is better when EV drivers greedily decide on their station visits (*D-SELF*) compared to when navigation platforms locally coordinate their charging demand (*P-SELF*).

Result 8. *Online coordination with VCG pricing yields a significant system benefit compared to an uncoordinated setting by decreasing the social cost by 42% when platforms act selfishly (P-SELF), and 44% when drivers act selfishly (D-SELF).*

Result 9. *Local charging demand coordination (P-SELF) can yield a higher social cost than without any coordination at all (D-SELF).*

Finally, Figure 8 compares the realized platform payoff, averaged over all demand realizations and platforms, normalized by the number of navigated drivers per platform for each online setting, against the offline benchmark (*OFF*). These results highlight the trade-off that exists between the additional cost induced by coordination via VCG pricing and the cost reduction due to a system-optimized allocation. As can be seen, VCG pricing mostly benefits a platform when the ratio of stations compared to the number drivers in the system is large enough, as otherwise the price paid by a platform increases due to the platform's negative impact on the system. For a large search radius ($\bar{S} = 2000$ meters), the online *VCG*-dd policy outperforms any naive allocation strategy, independent of the total number of drivers in the system.

Result 10. *VCG pricing significantly outperforms any naive strategy (P-SELF, D-SELF) for larger search areas ($\bar{S} = 2000m$) by decreasing a platform's payoff on average by 58% compared to P-SELF and by 61% compared to D-SELF.*

Result 11. *In some cases, VCG pricing may slightly increase a platform's payoff, when the room for improving the social outcome with coordination is small, e.g, for a large number of drivers in a small search area.*

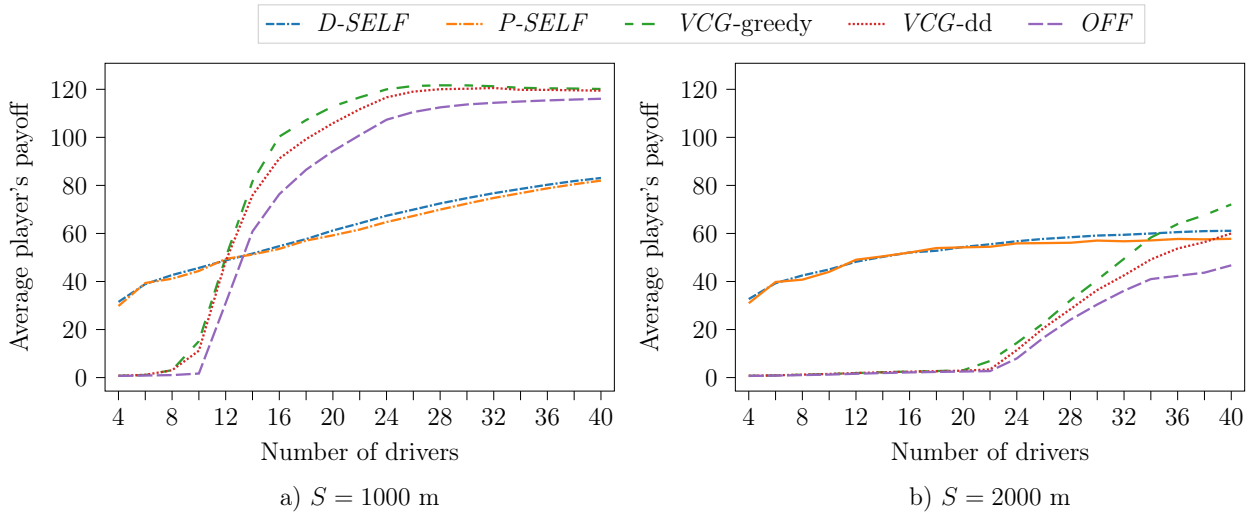


Figure 8: Averaged platform's payoff for *P-SELF*, *D-SELF*, *VCG*-dd, *VCG*-greedy, and *OFF*.

6. Conclusion

In this paper, we analyzed the dynamics between several self-interested navigation service platforms that seek to best allocate charging stations to EV drivers. In this context, we studied the problem of conflicting charging station assignments realized by independent platforms from a game-theoretical perspective and introduced the FCSA game. We showed that the game can neither be represented as a congestion game, nor admits a guaranteed PNE. To steer the system towards a stable and optimal social outcome, we studied the VCG mechanism in both offline and online settings, such that coordinated platforms' assignment decisions benefit the overall system. We further discussed how to extend the mechanism to account for heterogeneously weighted players in both settings. In the online setting, we introduced a data-driven online allocation policy. We analyzed the benefits of implementing our mechanisms by conducting a case study for the city of Berlin, and showed that coordination decreases the social cost by up to 52% in the offline setting and by up to 42% in the online setting. We further showed in an offline setting that optimized players' weights may increase the likelihood that participants benefit from participating in the VCG mechanism. Finally, our results showed that the online data-driven allocation policy performs on average slightly better than a myopic policy, but may significantly improve the social outcome by up to 55% in congested scenarios.

References

Ackermann, H., H. Röglin, B. Vöcking. 2009. Pure nash equilibria in player-specific and weighted congestion games. *Theoretical Computer Science* **410**(17) 1552–1563.

Anshelevich, E., A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, T. Roughgarden. 2008. The price of stability for network design with fair cost allocation. *SIAM Journal on Computing* **38**(4) 1602–1623.

AVER. 2021. Overview of romanian emobility 2020. Url: https://aver.ro/2021/01/05/evmap-by-counties-2020/ (last accessed: 19.04.22).

Ayala, D., O. Wolfson, B. Xu, B. Dasgupta, J. Lin. 2011. Parking slot assignment games. *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*. 299–308.

Clarke, E. H. 1971. Multipart pricing of public goods. *Public Choice* **11** 17–33.

cleantechnica. 2021. 16 countries now over 10% plugin vehicle share, 6 over 20%. Url: https://cleantechnica.com/2021/09/05/16-countries-now-over-10-plugin-vehicle-share-6-over-20/ (last accessed: 04.04.22).

Enlit. 2022. Overcoming the e-mobility chicken or egg dilemma. Url: https://www.enlit.world/low-carbon-transportation/electric-vehicles/overcoming-the-e-mobility-chicken-or-egg-dilemma/ (last accessed: 19.04.22).

Even-Dar, E., A. Kesselman, Y. Mansour. 2003. Convergence time to nash equilibria. *Automata, Languages and Programming*.

Friedman, E. J, D. C Parkes. 2003. Pricing WiFi at Starbucks: issues in online mechanism design. *Proceedings of the 4th ACM Conference on Electronic Commerce*. 240.

Garatti, S., M. Campi. 2022. Risk and complexity in scenario optimization. *Mathematical Programming* **191** 243–279.

Gerding, E. H., V. Robu, S. Stein, David C. Parkes, A. Rogers, N. R. Jennings. 2011. Online mechanism design for electric vehicle charging. *10th International Conference on Autonomous Agents and Multiagent Systems 2011*. 761–768.

Goemans, M.X., L. Li, V.S. Mirrokni, M. Thottan. 2006. Market sharing games applied to content distribution in ad hoc networks. *IEEE Journal on Selected Areas in Communications* **24**(5) 1020–1033.

Groves, T. 1973. Incentives in teams. *Econometrica* **41**(4) 617–631.

Guillet, M., M. Schiffer. 2022. Coordinated Charging Station Search in Stochastic Environments: A Multi-Agent Approach. *https://arxiv.org/pdf/2204.14219.pdf* .

He, F., Y. Yin, Z. Chen, J. Zhou. 2015. Pricing of parking games with atomic players. *Transportation Research Part B: Methodological* **73** 1–12.

Milchtaich, I. 1996. Congestion games with player-specific payoff functions. *Games and Economic Behavior* **13**(1) 111–124.

Monderer, D., L. S. Shapley. 1996. Potential games. *Games and Economic Behavior* **14**(1) 124–143.

Nelder, C., E. Rogers. 2019. Reducing ev charging infrastructure costs. Tech. rep., Rocky Mountain Institute.

Nisan, N., A. Ronen. 2007. Computationally feasible VCG mechanisms. *Journal of Artificial Intelligence Research* **29** 19–47.

Nisan, N., T. Roughgarden, E. Tardos, V. V. Vazirani. 2007. *Algorithmic Game Theory*. Cambridge University Press.

Parkes, D. C., S. Singh, D. Yanovsky. 2004. Approximately efficient online mechanism design. *Advances in Neural Information Processing Systems*.

Parkes, David C, Satinder Singh. 2003. An MDP-based approach to online mechanism design. *Advances in Neural Information Processing Systems*.

Rigas, E. S., E. Gerding, S. Stein, S. D. Ramchurn, N. Bassiliades. 2020. Mechanism design for efficient allocation of electric vehicles to charging stations. *11th Hellenic Conference on Artificial Intelligence*. 10–15.

Roberts, K. 1979. The characterization of implementable choice rules. JJ-Laffont (ed.), ed., *Aggregation and Revelation of Preferences*. Noth Holland Publishing Company.

Rosenthal, R. W. 1973. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory* **2**(1) 65–67.

Secci, S., J. Rougier, A. Pattavina, F. Patrone, G. Maier. 2011. Peering equilibrium multipath routing: A game theory framework for internet peering settlements. *IEEE/ACM Transactions on Networking* **19**(2) 419–432.

Sheikhim, A., S. Bahrami, A. M. Ranjbar, H. Oraee. 2013. Strategic charging method for plugged in hybrid electric vehicles in smart grids, a game theoretic approach. *International Journal of Electrical Power & Energy Systems* **53** 499–506.

Sohet, B., Y. Hayel, O. Beaude, A. Jeandin. 2021. Coupled charging-and-driving incentives design for electric vehicles in urban networks. *IEEE Transactions on Intelligent Transportation Systems* **22**(10) 6342–6352.

Stein, S., M. Ochal, I. A. Moisoiu, E. Gerding, R. Ganti, T. He, T. L. Porta. 2020. Strategyproof reinforcement learning for online resource allocation. *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*. 1296–1304.

Tushar, W., W. Saad, H. V. Poor, D. B. Smith. 2012. Economics of electric vehicle charging: A game theoretic approach. *IEEE Transactions on Smart Grid* **3**(4) 1767–1778.

Vickrey, W. 1961. Counter speculation, auctions, and competitive sealed tenders. *The Journal of finance* **16**(1) 8–37.

Wang, P., H. Guan, P. Liu. 2020. Modeling and solving the optimal allocation-pricing of public parking resources problem in urban-scale network. *Transportation Research Part B: Methodological* **137** 74–98.

Xu, S., M. Cheng, X. Kong, H. Yang, G. Huang. 2016. Private parking slot sharing. *Transportation Research Part B: Methodological* **93** 596–617.

Zou, B., N. Kafle, O. Wolfson, J. Lin. 2015. A mechanism design based approach to solving parking slot assignment in the information era. *Transportation Research Part B: Methodological* **81** 631–653.

A. Proofs

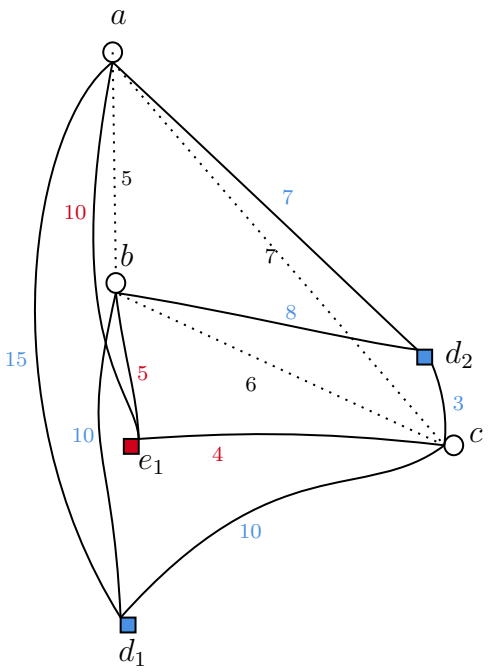
Proof of Proposition 1. We show the non-guaranteed PNE existence by finding a game instance that does not possess any. Figure 9 visualizes such a game instance. The instance comprises three stations a, b, and c for two players p_1 and p_2: the first player has two drivers d_1 and d_2, the second player has only one driver e_1. None of the possible game outcomes corresponds to a Nash equilibrium. If each player individually optimizes the assignment of its drivers to the available station, we obtain the following profile $((b,c),(c))$, with c conflicting. From there, p_2 should re-assign its unique driver to b, which is then conflicting. In this case, p_1 is better off reassigning d_1 to c and d_2 to a, as the total cost (=17) for strategy (c,a) is lower than the total cost (=18) for strategy (a,c). However, p_1 should now re-assign its unique driver to c such that p_1's best response is again its individually optimized assignment solution. The last strategy profile now equals the initial strategy profile, such that the path $P = ((b,c),c),((b,c),b),((c,a),b),((c,a),c),((b,c),c)$ of length 5 is an improvement path. Additionally, starting from any other strategy profile not included in P, one may reach P within at most two best responses moves, such that there exists no sink in the best response dynamics graph and, accordingly, no PNE. □

Proof of Proposition 2. We show the non-existence guarantee of a ρ-PNE with $\rho \leq \bar{\beta} - \Delta_t$, by using an intermediate game G, defined with the same set of strategies than the FCSA game but with new payoff functions. We ignore travel times, such that assigning a station induces a penalty cost only for the non-closest drivers, i.e.,

$$c_i(k,s) = \begin{cases} \bar{\beta}, & \text{if } s_i(k) = v_0 \\ 0, & \text{if } \forall k' \in \mathcal{D}_j \;\; \forall j \in \mathcal{N},\; j \neq i \; (s_j(k') = s_i(k)) \Rightarrow (t_{k',v} \geq t_{k,v}) \\ \bar{\beta} & \text{otherwise} \quad . \end{cases} \tag{A.1}$$

Accordingly, player i's payoff can be expressed as $u_i(s) = k \cdot \bar{\beta}$, with $k \in \mathbf{N}$. Similar to Proposition 1, G has no guaranteed PNE, as we can find an instance of G that admits none.

Figure 9: G1' instance with no PNE



For such instance,

$$\forall s \in \mathcal{S}, \forall i \in \mathcal{N}, \exists s'_i \in \mathcal{S}_i, \text{st. } u_i(s'_i, s_{-i}) < u_i(s) \ . \tag{A.2}$$

When a player can strictly decrease its payoff by switching from strategy s_i to s'_i, then its payoff decreases by at least $\bar{\beta}$, such that $u_i(s) - u_i(s'_i, s_{-i}) \geq \bar{\beta}$, which corresponds to one additional player being successfully assigned to a station.

By assumption, each feasible profile s in G is feasible in the FCSA game and the previous condition implies that for some FCSA game instances, a player can decrease its cost by obtaining at least one more driver successfully assigned to a station. In this case, assume for player i, that l drivers are successfully assigned to a station in s_i. For any alternative strategy improving s'_i, this implies at least $l+1$ successfully assigned drivers. Let us denote with $t_{\max}$ and $t_{\min}$ the highest and lowest driving times between stations and drivers. We let $n_i = |\mathcal{D}_i|$ be the number of drivers belonging to i. Then, we have

$$u_i(s_i, s_{-i}) \geq (n_i - l) \cdot t_{\min} + l \cdot \bar{\beta} \tag{A.3}$$

and

$$u_i(s'_i, s_{-i}) \leq (n_i - l - 1) \cdot t_{\max} + (l-1) \cdot \bar{\beta} \ , \tag{A.4}$$

such that i decreases its cost by switching from s_i to s'_i by at least $\Delta = \bar{\beta} + (t_{\min} - t_{\max})$. Finally, we showed that $\rho \geq \Delta$.

□

Proof of Proposition 3. Assuming that all players tell the truth, the principal computes the

cost-minimal allocation

$$a = \arg\min_{b \in A} \sum_{k \in \mathcal{D}} c(k, b) \ , \tag{A.5}$$

with $c(k, a)$ as defined in Equation (3.3). The payoff of player i corresponds to $\sum_{k \in \mathcal{D}} c(k, a) - h_{-i}$. We let a' be the modified allocation, when player i reassigns its drivers to its stations allocated by the principal. All drivers that do not belong to player i are still assigned to the same station in a'. The payoff of player i in this case corresponds to

$$\sum_{k \in \mathcal{D}} c(k, a') - h_{-i} \ . \tag{A.6}$$

However, as allocation a minimizes the total cost, i.e., the assignment cost of drivers to stations, then

$$\sum_{k \in \mathcal{D}} c(k, a') \geq \sum_{k \in \mathcal{D}} c(k, a) \ , \tag{A.7}$$

such that deviating from the assigned allocation of stations to drivers increases the payoff of a player. Concluding, a player has no benefit from assigning its drivers differently to the stations allocated by the principal. □

Proof of Proposition 4. First, let us assume that a driver cannot deviate from the received assignment of drivers to stations. Let $\hat{\theta}$ be the reported information, and with a slight abuse of notation let $\hat{k}$ be a reported driver information (i.e., its location), while k denotes the actual driver information. Let $\hat{a} = f(\hat{\theta})$. Then a player's payoff corresponds to $\sum_{k \in \mathcal{D}} c(k, \hat{a}) - h_{-i}$, as the assignment cost $c(\hat{k}, \hat{a})$ corresponds to the cost with actual driver information, but with allocation $\hat{a}$ computed based on the driver's reported information by the corresponding player, such that $c(\hat{k}, \hat{a}) = c(k, \hat{a})$. We differentiate two cases:

i) *Correct number of reported drivers:* If a single player correctly reports its number of drivers but misreports its drivers' location, such that $\hat{\theta} \neq \theta$, then

$$\sum_{k \in \mathcal{D}} c(k, \hat{a}) \geq \sum_{k \in \mathcal{D}} c(k, f(\theta)) \ , \tag{A.8}$$

as $f(\theta) = \arg\min_{b \in A} \sum_{k \in \mathcal{D}} c(k, b)$, and the player's payoff increases when not telling the truth.

ii) *Incorrect number of reported drivers:* If a single player lies on the number of reported drivers, then each non-reported driver $\hat{k}$ does not get any assigned station, which yields a penalty cost for each of them, such that $c(k, \hat{a}) = \bar{\beta} \geq c(k, a) \ \forall a \neq \hat{a}$. Then, for each extra reported driver $\hat{k}$ the actual driver assignment cost becomes $c(k, \hat{a}) = 0$. Accordingly, the principal optimizes the allocation of actual drivers on a reduced subset of stations $\mathcal{V}' \subset \mathcal{V}$, i.e., stations that are not assigned to virtual reported drivers, such

that

$$\min_{b\in A,\mathcal{V}'} \sum_{k\in\mathcal{D}} c(k,b) \geq \min_{b\in A,\mathcal{V}} \sum_{k\in\mathcal{D},\mathcal{V}} c(k,a) \ . \tag{A.9}$$

In theses two cases of misreported drivers' numbers, a player's payoff is at least smaller when all drivers are correctly reported. Furthermore, following i), a player's payoff is at least smaller when all players' information are correctly reported.

Second, let us assume that a driver can deviate from the received assignment of drivers to stations. If a player lies, then its payoff is larger than when telling the truth as shown above, such that

$$\sum_{k\in\mathcal{D}} c(k,\hat{a}) + h_{-i} \geq \sum_{k\in\mathcal{D}} c(k,f(\theta)) + h_{-i} \ , \tag{A.10}$$

as $f(\theta) = \arg\min_{b\in A} \sum_{k\in\mathcal{D}} c(k,b)$. As its payoff is not minimized when lying, it can deviate from the allocation $\hat{a}$ by reassigning other drivers to its received stations, such that the new allocation a' decreases its payoff obtained with allocation $\hat{a}$. However, the newly obtained allocation $\hat{a}$ cost is at least as large as the minimum cost allocation obtained when telling the truth, such that a player is better off reporting its true information and not reassigning its drivers to different charging stations.

□

Proof of Proposition 5. We recall the proof exposed in Theorem 1 from Parkes et al. (2004), that transposes the regular VCG incentive-compatibility to an in-expectation incentive-compatibility.

Let t be the last decision epoch before the first epoch, where i misreports information θ^{t+1} (and may further misreport her information), such that $\hat{\theta}^t = \theta^t$. By expliciting payment p_i in Equation 3.24, omitting $OPT_{\theta_{-i}}$ that does not depend on player i, and omitting $\sum_{i\in\mathcal{N}} \sum_{\tau=0}^{\tau-1} d_\tau^t(x_\tau, \pi_\tau(x_\tau))$ that does not depend on realizations later than t, we need to show that

$$\begin{aligned} & E_{\tau>t}[\sum_{i\in\mathcal{N}} \sum_{\tau=t}^{T} d_\tau^i(x_\tau,\pi_\tau(x_\tau))] \leq E_{\tau>t}[\sum_{i\in\mathcal{N}} \sum_{\tau=t}^{T} d_\tau^i(\hat{x}_\tau,\pi_\tau(\hat{x}_\tau))] \ \ \forall\hat{\theta}^t \ \forall t \\ \Leftrightarrow \ & E_{\tau>t}[\sum_{\tau=t}^{T} d_\tau(x_\tau,\pi_\tau(x_\tau))] \leq E_{\tau>t}[\sum_{\tau=t}^{T} d_\tau(\hat{x}_\tau,\pi_\tau(\hat{x}_\tau))] \ \ \forall\hat{\theta}^t \ \forall t \ . \end{aligned} \tag{A.11}$$

The immediate cost for a misreported information depends on the driver's true information, i.e., actual location, and on the decision taken upon the misreported information, such that

$$\sum_{i\in\mathcal{N}} \sum_{\tau=t}^{T} d_\tau^i(\hat{x}_\tau,\pi_\tau(\hat{x}_\tau)) = \sum_{i\in\mathcal{N}} \sum_{\tau=t}^{T} d_\tau^i((\theta_{<\tau},\hat{a}_{\leq\tau-1}),\pi_\tau(\hat{x}_\tau)) \ .$$

The right part of (A.11) corresponds to the expected cost from x_t (as $\hat{x}_t = x_t$ by assumption) following policy $\hat{\pi}$, with $\hat{\pi}(x_\tau) = \pi(\hat{x}_\tau) \ \forall\tau \in [t,...,T]$, i.e., $V^{\hat{\pi}}(x_t)$. The left part

of (A.11) corresponds to the expected cost from x_t following policy π, $V^{\pi}(x_t)$, such that Equation (A.11) can be rewritten as

$$V^{\pi}(x_t) \leq V^{\hat{\pi}}(x_t).$$

Since π is the optimal MDP-policy by assumption, Equation A.11 holds; otherwise this contradicts π's optimality. Finally, we justified that a player has no incentive to reveal false information. □

Proof of Proposition 6. We need to show that the following inequality

$$\begin{aligned} &E_{\tau>t}[v^i(\theta^i, a_{\leq T}) - \tilde{p}^i(\theta, \pi)] \leq E_{\tau>t}[v^i(\theta^i, \hat{a}_{\leq T}) - \tilde{p}^i(\hat{\theta}, \pi)] \;\; \forall \hat{\theta}^t \; \forall t \\ \Leftrightarrow &w_i \cdot E_{\tau>t}[v^i(\theta^i, a_{\leq T}) - \tilde{p}^i(\theta, \pi)] \leq w_i \cdot E_{\tau>t}[v^i(\theta^i, \hat{a}_{\leq T}) - \tilde{p}^i(\hat{\theta}, \pi)] \;\; \forall \hat{\theta}^t \; \forall t \end{aligned} \tag{A.12}$$

holds. By omitting the same terms in (A.12) than in (3.24) (see Proposition 5), we can show that the following inequality holds

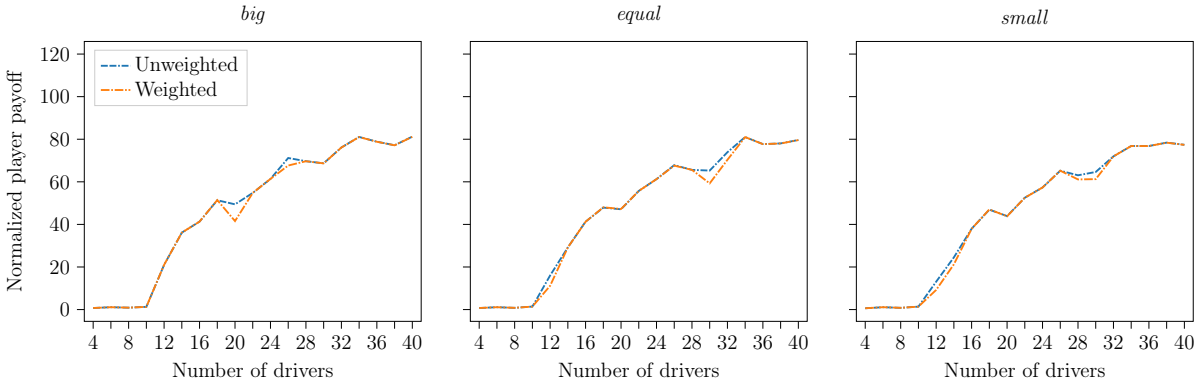
$$\begin{aligned} \Leftrightarrow &E_{\tau>t}[\sum_{i\in\mathcal{N}} \sum_{\tau=t}^{T} w_i \cdot d_{\tau}^i(x_\tau, \pi_\tau(x_\tau))] \leq E_{\tau>t}[\sum_{i\in\mathcal{N}} \sum_{\tau=t}^{T} w_i \cdot d_{\tau}^i(\hat{x}_\tau, \pi_\tau(\hat{x}_\tau))] \;\; \forall \hat{\theta}^t \; \forall t \\ \Leftrightarrow &E_{\tau>t}[\sum_{\tau=t}^{T} \tilde{d}_{\tau}(x_\tau, \pi_\tau(x_\tau))] \leq E_{\tau>t}[\sum_{\tau=t}^{T} \tilde{d}_{\tau}(\hat{x}_\tau, \pi_\tau(\hat{x}_\tau))] \;\; \forall \hat{\theta}^t \; \forall t \; . \end{aligned} \tag{A.13}$$

Using the same argument as previously, with π being in this case the optimal policy solving the MDP with updated immediate cost $\tilde{d}$, we justified inequality (A.13). Accordingly, a player has no incentive to reveal false information when players are weighted, if the payment rule is weighted too. □

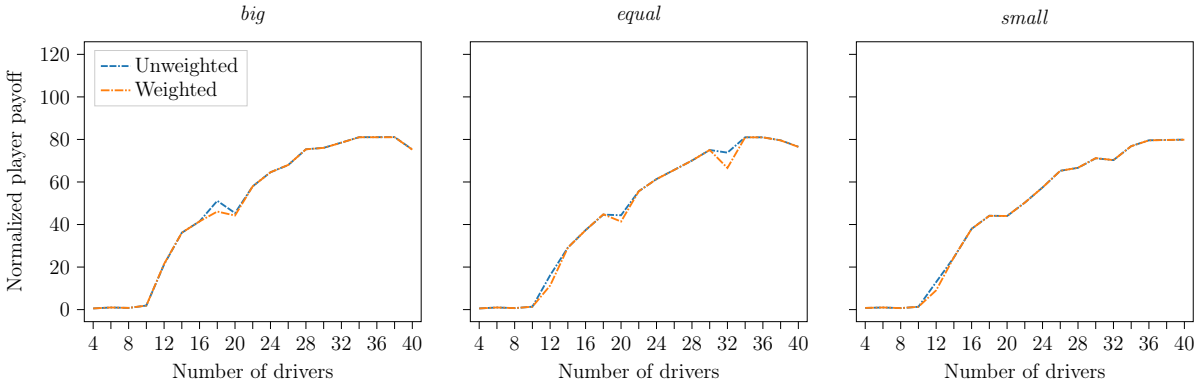
B. Additional Numerical results

Weights optimization: Figures 10a-10c show the averaged payoff per platform per scenario, depending on the total number of drivers. Payoffs are shown for both weighted and unweighted *VCG*. For instances that could not solve the weights optimization, unweighted payoffs are shown.

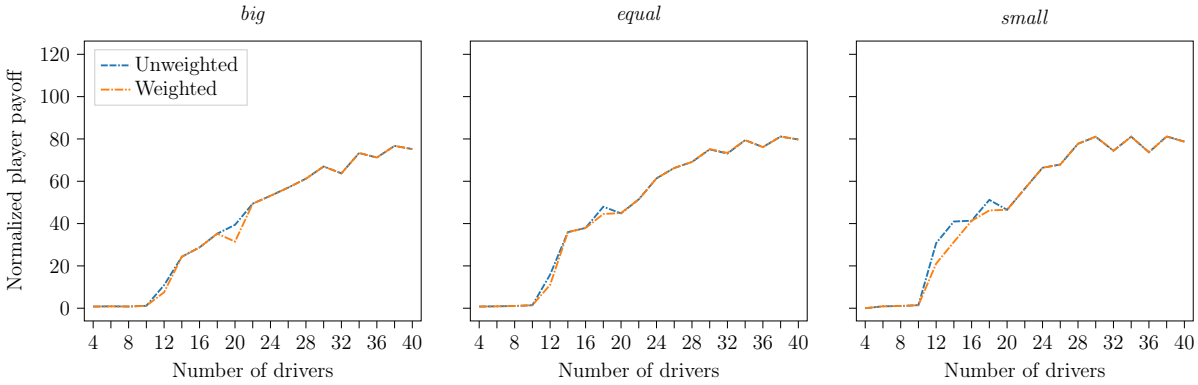
Online averaged platform payoffs: Similar to Figure 4, Figure 11 compares the normalized payoffs obtained for each platform depending on the number of drivers, for each distribution scenario in an online setting. Analogous to offline results, online results show that a platform with a lower share of drivers (platform B in *small*) obtains a higher payoff than a platform with a higher share of managed drivers. Compared to the offline setting, platforms' payoffs are slightly higher due to the approximation induced by the online allocation policy compared to a perfect-information allocation (cf. Figure 4).



(a) Platform A



(b) Platform B



(c) Platform C

Figure 10: Impact of weighted *VCG* on each platform's payoff in an online setting

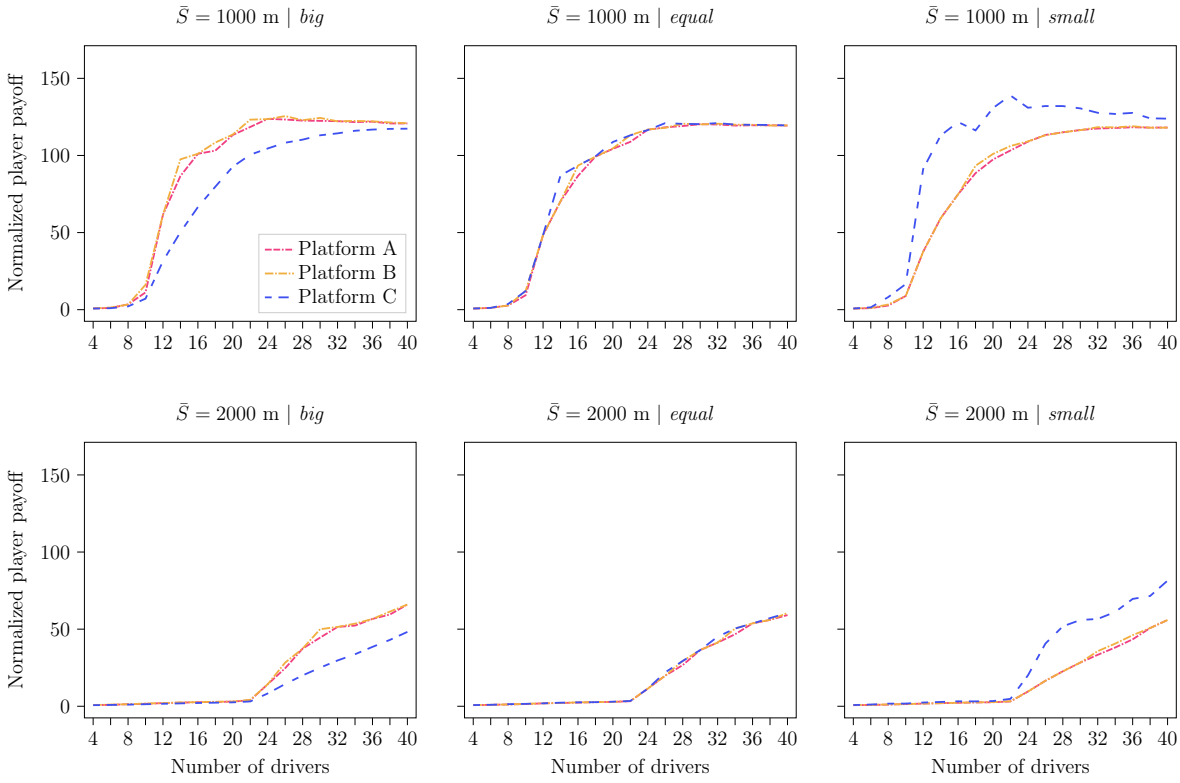


Figure 11: Impact of the distribution scenario on each platform's payoff in an online setting

6 Conclusion

1 Main findings of the thesis

Electric vehicles (EVs) are an essential lever for decarbonizing the transportation sector but challenges to seamlessly recharging a vehicle hinder its adoption in the private market. To this end, at-home charging solutions offer a trusted charging experience but are insufficient for one-way or long-distance trips, and must complement but cannot replace reliable public charging options. However, unreliable public charging station availability and congestion risks prevent EV users from trusting the public charging infrastructure. Therefore, operational solutions that mitigate uncertainties in the public charging experiences are needed to foster EV adoption. Against this background, this thesis contributes novel models and solution methods, that aim to reliably help EV drivers find a free charging station in the presence of uncertainties. The first two methodological chapters focus on a driver-centric approach, while the third methodological chapter approaches the problem from a system-centric angle.

Chapter 3 derives a comprehensive algorithmic solution framework to solve the stochastic charge pole search (SCPS) problem under uncertainty, in which a driver aims to find a cost minimal sequence of stations to visit. Several variants of the problem are analyzed to account for charging station heterogeneity and the possibility of waiting at an occupied charging station. A labeling algorithm is developed and exactly solves the problem by decomposing the Markovian policy cost to derive an efficient dominance criterion. A simplified dominance criterion provides a high-quality heuristic. Furthermore, a rollout algorithm is developed to solve large-sized instances. Results show that implementing a combination of both heuristics best addresses the trade-off between solution quality and computational performances. Simulation-based results that use real-world data for the city of Berlin further show that advanced search strategies compared to greedy search strategies can save significant time for the EV driver, by up to 44%, while increasing the likelihood of successfully terminating the search within a limited time span.

Chapter 4 extends the SCPS problem to a multi-agent setting and formalizes the multi-agent SCPS (MASCPS) problem as a Markov decision process (MDP). Online algorithms are introduced to synchronize multiple agents' search paths through joint search path planning or information-sharing. Extensive numerical experiments compare several practical settings: search path planning can be decentralized or centralized, and static or dynamic. In the static decentralized case, agents can share either en-routes observations of occupied stations, their intended charging station visit, or both. The numerical results show that coordinated search strategies significantly outperform uncoordinated search strategies. On average, a centralized decision-making approach decreases the system cost by 28% compared to agents performing uncoordinated stochastic searches, while a de-

centralized decision-making setting with information-sharing already achieves a 26% cost decrease. User-centric analyses further show that coordination improves the worst possible uncoordinated search solution and preserves the best possible uncoordinated search solution quality.

Chapter 5 aims to mitigate charging station utilization congestion caused by large uncoordinated charging demand, and analyzes the optimal allocation of drivers to available stations by navigation service providers in competitive environments. To this end, the fleet charging station allocation (FCSA) game is introduced and shown not to admit a guaranteed existence of a pure Nash equilibrium. Then, a mechanism design approach is introduced to coordinate the allocation of stations to drivers across self-interested navigation service platforms. Specifically, the VCG mechanism is implemented in both offline and online settings, and a data-driven approach is applied to solve the resulting online station allocation problem. Numerical results show that platforms coordination significantly improves the system efficiency, reducing the social cost by up to 52% in an offline setting and up to 42% in an online setting.

In conclusion, this thesis contributes new models, methods, and managerial insights on relevant real-world problems closely related to the adoption of EVs. First, the SCPS problem is formalized as a single-agent finite-horizon MDP, and then extended to the MASCPS problem. The single-agent model accounts for charging station heterogeneity and waiting decisions at occupied charging stations. In the multi-agent case, we define several variants of the problem depending on the level of information-sharing between agents and the degree of centralization. The FCSA is formalized as a new game, and a mechanism design approach based on off and online VCG is applied to this setting to coordinate the self-interested participants. Second, a label-setting algorithm with a novel dominance criterion is developed to solve the MDP-based formulation of the SCPS problem. A heuristic that uses a relevant subset of the dominance checks yields close to the optimal solutions, and is a central component of the online algorithmic framework developed to solve the multi-agent setting. Third, short-term and long-term managerial insights are derived. When helping drivers to recharge their vehicle using existing public charging infrastructure, results show the benefit of accounting for charging station availability uncertainty, and coordinating EV drivers' requests coordination. While an EV driver can already save up to 44% of her driving time without search coordination, our results show that the averaged system performance cost further decreases by up to 28% with coordination. Focusing on a longer-term perspective, centrally coordinating the allocation of stations to drivers, which is otherwise realized by competing self-interested navigation service platforms, benefits EV users. Specifically, system-level coordination may decrease

the social cost by 42% in a real-world setting, compared to a scenario without charging station visits coordination between navigation service platforms.

2 Limitations of the thesis and future research directions

Although this thesis provides a comprehensive algorithmic framework that addresses charge anxiety faced by EV drivers, a few comments on the models and corresponding assumptions are in order. First, Chapter 3 and Chapter 4 assume that station availability probabilities depend on drivers' observations, and both chapters model time-dependent recovering availability probabilities. However, these two chapters also assume constant, initially given availability probabilities. It is reasonable in practice to have marginal probability variation in a relatively short time span, i.e., at most in a couple of hours, but analyzing longer planning horizons may require taking into account probability variations. Second, in Chapter 4, future charging demand is represented as unpredictable exogenous information, and the modeled uncertainty remains limited to the availability of visited charging stations. Uncertainty about future charging requests can be addressed by a model-free (deep) reinforcement learning approach. Third, for both the MASCPS problem and the FCSA game settings, respectively introduced in Chapter 4 and Chapter 5, drivers' behaviors are considered to be deterministic. While this assumption is necessary to provide an initial rigorous and methodological analysis, it may be relaxed in subsequent works. Here, considering drivers as self-interested players may yield further interesting insights.

In the following, we identify possible avenues for further research, related to modeling or methodological contributions. Focusing on modeling contributions, the MASCPS problem (Chapter 4) can be extended to account for the robustness and fairness of an individual driver solution but also for adversarial driver behaviors. Furthermore, the SCPS and MASCPS problems focus on charging-on-the-spot use-cases, e.g., charging at destination, but availability uncertainty may also arise during long-distance trips, which requires considering en-route possible detours due to unoccupied charging stations. Both SCPS and MASCPS problems may extend to long-distance EV routing, with similar analyses on the benefit of information-sharing and request centralization in the multi-agent case. Future work may account for increased real-time availability information accuracy, e.g., correct real-time information of unavailable charging stations. In this case, dynamic re-planning approaches must be carefully designed to account for updated real-time occupancy. Finally, the FCSA game can be extended in two manners. First, the mechanisms'

principal may be profit-driven, e.g., an inter-charge operator that aims to maximize its revenues from station booking payments. Second, single drivers may also be considered strategic agents, although non-strategic drivers may remain relevant in the context of future autonomous mobility-on-demand systems.

Focusing on methodological contributions, the effective labeling-based solution procedure derived in Chapter 2 to solve the SCPS problem may be further researched and applied to MDPs with similar structures. An immediate application extension can be the 2-agents SCPS problem with joint static path planning. In this case, nodes of the underlying graph on which labels expand represent the joint agent positions.