Received: 10 May 2022 | Accepted: 23 February 2023

DOI: 10.1111/poms.13997

POMS

ORIGINAL ARTICLE

Coordinated charging station search in stochastic environments: A multiagent approach

Marianne Guillet[1,2] | **Maximilian Schiffer**[1,3]

[1]TUM School of Management, Technical University of Munich, Munich, Germany

[2]TomTom Location Technology Germany GmbH, Berlin, Germany

[3]Munich Data Science Institute, Technical University of Munich, Berlin, Germany

Correspondence
Marianne Guillet, TUM School of Management, Technical University of Munich, 80333 Munich, Germany.
Email: marianne.guillet@tomtom.com

Handling Editor: Vijay Mookerjee

Abstract

Range and charge anxiety remain essential barriers to a faster electric vehicle (EV) market diffusion. To this end, quickly and reliably finding suitable charging stations may foster an EV 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, for example, 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, for example, 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 multiagent 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. Even in long planning horizons, our algorithm reduces the system cost by 25% while increasing each driver's search reliability.

KEYWORDS

electric vehicle charging, information sharing, multiagent systems, stochastic search

1 | INTRODUCTION

Electric vehicles (EVs) can play a crucial role in decarbonizing the transportation sector, given a rapid energy transition and continuous battery technology improvement (Schuller & Stuart, 2018). To sustain the global EV market's growth of the past 2 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

Accepted by Vijay Mookerjee, after two revisions.

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, that is, 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 service platforms that offer services to find available charging stations through local navigation devices or an online Application Programming Interface (API), may centralize driver requests or leverage active and passive driver community input, for example, 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 may increase the search's reliability. However, such approaches so far focus always on a single-agent setting. Accordingly, coordinating drivers in a multiagent 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 its time.

With this paper, we close this research gap by extending stochastic single-agent search algorithms to a stochastic multiagent setting, accounting for information sharing and possible 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 multiagent 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, Goodson, et al. (2021) solve the EV 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 & Wolfson, 2018; Schmoll & Schubert, 2018) or more specific settings, for example, 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 multiagent settings, most works on resource search problems under uncertainty focus on cooperative searches, that is, 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 multitarget search problem. Qin et al. (2020) address the multiagent 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 multiagent charging station search setting: Here, each agent terminates her search when she found at least one non-shareable available resource. Existing work on multiagent 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, Cousineau, et al., 2021) 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. Multiagent 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 threefold. First, we formalize the underlying centralized multiagent decision-making problem. To this end, we define the planning problem as a single–decision maker Markov decision problem (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% on average, and that a static decentralized coordination strategy already achieves a 26% cost decrease if visit intentions are shared, compared to an uncoordinated scenario. 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% average cost decrease when drivers depart within a short time span, but achieves a 26% average 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, focusing on long-term planning horizons, additional analyses show that a centralized coordination strategy achieves a slightly lower cost reduction compared to a short-term planning horizon but nevertheless decreases the system cost by 25% on average compared to an uncoordinated scenario.

1.3 | Organization

The remainder of this paper is as follows. Section 2 introduces our problem setting. In Section 3, we formalize our multiagent 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. Throughout the paper, we refer to its Supporting Information as appendices to keep the writing concise.

2 | PROBLEM SETTING

We focus on a nonadversarial multiagent 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. We consider an online setting in which drivers enter the system sequentially and reveal their current position upon entering. Accordingly, drivers start their search asynchronously; each driver departs from a given location, that is, 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 or 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, that is, 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, that is, 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 rea time with the central platform. In the decentralized

TABLE 1 Problem settings overview.

		Visit intentions	Availability observations	Decision making	User-dependent solutions
Static path planning	D			Decentralized	No
	DI	✓		Decentralized	No
	DO		✓	Decentralized	No
	DIO	✓	✓	Decentralized	No
Dynamic path planning	CIOd	✓	✓	Centralized	Yes
	DOd		✓	Decentralized	Yes

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.

2.1 | Static path 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 decentralized setting (D), no information is shared between agents, while each agent is aware of prior available observations of other agents when computing her search path in the decentralized setting with observation sharing (DO). Both settings D & DO are purely informative, such that agents are unaware of other agents' planned search paths. The decentralized intention-sharing settings (DI & DIO) extend the D & DO settings: Each agent additionally knows previous agents' search paths, that is, their intended station visits and visit times. Agents may use this information selfishly, for example, by visiting other agents' target stations first, or they may use this information collaboratively.

2.2 | Dynamic path 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 (CIOd) in which a central planner, that is, 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 (DOd) of the decentralized observation-sharing setting DO. In this setting, the initial solution is dynamically replanned upon each occupied station visit to account for the latest shared observations. Note that we discard settings COd, DId, CId, and DIOd from our analyses for the following reasons: COd does not exist, as a planner who centrally assigns stations to drivers is aware of all drivers' visit intentions. In a DId or CId setting, 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 a DIOd setting, each driver has nearly as much information as a centralized planner, at the exception of other drivers' search radii or time budget preferences. As both settings are very similar, we limit our analyses to a CIOd setting, which requires less information sharing in practice.

2.3 | Discussion

A few comments on our problem setting are in order. First, we model our coordinated charging station search as a closed system. While this assumption seems to be a limitation from a theoretical perspective, it is appropriate from a practitioner's 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 assume charging station occupancy persistency in our basic problem setting. The validity of this assumption depends on the length of the considered planning horizon. Considering a short planning horizon, for example, between 15 min and 1 h, our persistency assumption remains valid and is in line with real-world observations: Here, typical charging durations exceed the planning horizon. Considering a long planning horizon, our persistency assumption is clearly unrealistic. We describe in Section 4.3 how to relax observation persistency and consider temporarily occupied stations with occupancy probabilities that recover over time. Alternatively, one can roll out several short planning horizons, and assume that information on observed (i.e., excluded) stations outdates

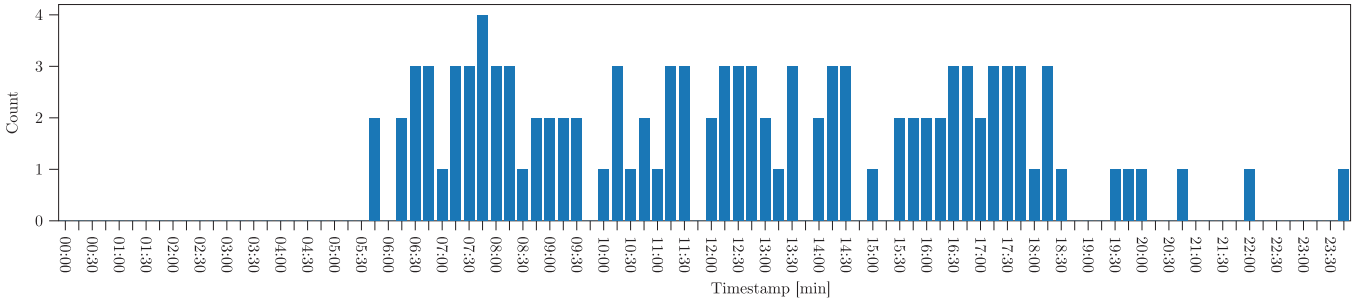


FIGURE 1 Number of ending trips on Tuesday 12.01.22 in the east area of Berlin, Germany. *Note*: The figure shows $\frac{1}{3}$ of all collected anonymized Origin-Destination pairs of all vehicles using TomTom navigation services, that end their trips in the east area of Berlin. [Color figure can be viewed at wileyonlinelibrary.com]

after some time, such that excluded stations can explicitly be considered as candidate stations again.

In our basic problem setting and analyses, we focus on short planning horizons and consequently assume occupancy persistency, which eases the problem's exposition, for two reasons: First, a short planning horizon, possibly used in a rolling-horizon approach, is sufficient to mitigate local bottlenecks, which are spatially and timely limited, and are currently the main problem in practice. Second, from a single-agent perspective, 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). However, we extend our analyses to long planning horizons in Section 6.2.3 to complete our analyses and show the efficacy of our methodology in such a setting.

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.

3 | MDP REPRESENTATION

We refer to the problem setting discussed in Section 2 as the multiagent stochastic charging pole search (MSCPS) problem and model it as a sequential multiagent 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 multiagent search with an omniscient single decision maker. 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 3.3.1.

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 $v \in \mathcal{V}$ can either be a charging station or a driver's departure location. We model the latter as an artificial charging station node with a constant availability probability of zero. 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 to ease the problem exposition, but explain in Section 4.3 how to relax this assumption. Note that when stating the MDP, we assume arrival times to be known for the sake of simplicity. However, our online algorithms do not require such knowledge.

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, that is, 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

$$x = (\vec{x_d}, \mathcal{J}, \mathcal{T}, \mathcal{O}), \tag{1}$$

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 $\vec{x_d} = (x^i)_{\forall i \in \mathcal{J} \bigcup \mathcal{T}}$ being the vector that describes the state of each agent.

Here, we define an agent's state x^i as

$$x^i = (v^i, t^i, s^i), \tag{2}$$

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 be ($s^i = \mathrm{t}$), 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, that is, 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^{f} if the station observed by λ is available, or as x^{t} if λ observes an occupied station or begins her search.

Available station (x^{f}) : 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^{t}): 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 decision stage, its new assigned station is $\acute{v}^\lambda = u$, while its new arrival time is $\acute{t}^\lambda = t^\lambda + t_{v^\lambda, u}$. If the current deciding agent λ takes no decision in the next stage, then the agent's status is set to en route, that is, $\acute{s}^\lambda = \mathrm{r}$. Otherwise, if λ remains the deciding agent in the next stage, that is, $\lambda = \acute{\lambda}$, the agent's status is either set to d, f, or t.

We define a policy $\pi \in \Pi$ as the state-action mapping function, such that $\pi(x) = u \in \mathcal{U}(x)$. To refer to an action u, given for a certain state x by policy π, we write $\pi(x)$.

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 $\acute{x}^{\mathrm{f}}$ for $\acute{\lambda}$ with probability $p_{\acute{v}}$ or an unsuccessful state $\acute{x}^{\mathrm{t}}$ with probability $1 - p_{\acute{v}}$.

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

$$V^\pi(x) = d(x, \pi(x)) + p_{\acute{v}} V^\pi(\acute{x}^{\mathrm{f}}) + (1 - p_{\acute{v}}) V^\pi(\acute{x}^{\mathrm{t}}), \quad (3)$$

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

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

$$V^\pi(x_n) = d(x_n, \pi(x_n)) + \beta^G \delta, \quad (4)$$

with δ being the binary variable that indicates whether at least one agent $i \in \mathcal{T}$ has failed her search. Appendix A in the Supporting Information summarizes the MSCPS problem's and the related MDP's notation.

3.2 | Policy representation

In Section 3.1, we introduced an agent-agnostic policy, that is, a policy that applies 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-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, that is, preserving user-independent search policies. In this special case, coordination accounts for other agents' 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

$$\pi^i(x) = \pi^i(\bar{x}), \ \forall (x, \bar{x}) \in \bar{\mathcal{X}} \times \bar{\mathcal{X}} \text{ st. } v^i = \bar{v}^i = v, \quad (5)$$

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.

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 is aware of the full state x, that is, aware of all other availability realizations observed by all other agents. From a single-agent perspective, the station selection depends on whether and where other agents terminated. In such a setting, an agent i, executing a search policy π, being located at station v with v being occupied, may not always visit the same next station. We let Π^{dep} be the set that contains user-dependent policies $\pi = \{\pi^i\}_{i \in D}$, with π^i being user dependent.

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 $C^i = (v^i_0, \dots, v^i_n)$ with i starting at v^i_0, following the vertices in sequence, and terminating either the search at any first available charging station $v \in C^i$, or unsuccessfully at v^i_n. 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 $C^i = (v^i_0, \dots, v^i_k)$ being the sequence of visited stations, and a being the realized availability at the last visited station v^i_k of C. If $a = 0$, then the agent takes an action u that consists of the single-station selection decision $u = (v), u \in \mathcal{A}$, with $v \in \tilde{\mathcal{V}}, v \notin C^i$, and $\tilde{\mathcal{V}}$ being the set of reachable stations from v^i_k. In this case, the transition function $p^i_t(\acute{x}|x, u)$ 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 $t_{v^i_k, v}$ between v^i_k and v, or to the penalty cost $\bar{\beta}^i$, if the agent already spent her time budget. If $a = 1$, she successfully terminated her search at v^i_k and the agent-specific station usage cost $\gamma^i_{v^i_k}$ 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 $C^i(\pi^i) = (v^i_0, \dots, v^i_n)$, with $\pi^i((v^i_0, \dots, v^i_k), 0) = v^i_{k+1} \ \forall\, k \in [0, \dots, n-1]$.

Agent i aims to find a policy π^i that minimizes her search cost $F^{\pi^i}((v^i_0), 0)$, with F being the single-agent value function, which is defined as

$$\begin{aligned} F^{\pi^i}(x) &= t_{v_k, v} + \sum_{\acute{x} \in \mathcal{S}} p^i_t(\acute{x}|x, \pi^i(x)) F^{\pi^i}(\acute{x}) \\ \Leftrightarrow F^{\pi^i}(C, 0) &= t_{v_k, v} + p^i_t((C', 1)|(C, 0), \pi^i(C, 0)).\gamma^i_v \\ &\quad + p^i_t((C', 0)|(C, 0), \pi^i(C, 0)) F^{\pi^i}(C', 0), \end{aligned} \tag{6}$$

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

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 ($t^i_v \geq t^j_v$) 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 $p^i_v(t)$. We let $\rho^i(t)$ be the probability that agent i found at least one station available among all stations v, from sequence $C^i = (v^i_1, \dots, v^i_n)$, for which the visit time t^i_v is lower than t, and define $\rho^i(t)$ as

$$\rho^i(t) = 1 - \prod_{v \in C^i, t^i_v \leq t} (1 - p^i_v(t)). \tag{7}$$

Then, $p^i_v(t)$ reads

$$p^i_v(t) = p_v \prod_{j \in D, j \neq i, t^j_v \leq t} \rho^j(t^j_v). \tag{8}$$

Note that both definitions are finitely nested as visits are ranked by the agents' arrival times.

We now define the user-dependent transition function as

$$\begin{aligned} p^i_t((C', 1)|(C, 0), (v)) &= p^i_v(t'), \\ p^i_t((C', 0)|(C, 0), (v)) &= 1 - p^i_v(t'), \end{aligned} \tag{9}$$

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

$$p_t((C',1)|(C,0),(v)) = p_v,$$
$$p_t((C',0)|(C,0),(v)) = 1 - p_v. \tag{10}$$

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 as

$$\alpha^i = (1 - \rho^i)\beta^i + A^i(\pi^i), \tag{11}$$

as we explain in more detail in Appendix C in the Supporting Information. 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

$$A^i(\pi^i) = \sum_{k=0}^{n-1} \left[t_{v_k^i, v_{k+1}^i} \bar{\rho}^i(\pi^i, k) \right] + \sum_{k=1}^{n} \gamma^i_{v_k^i} p^i_{v_k^i} \bar{\rho}^i(\pi^i, k-1). \tag{12}$$

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

$$\alpha^\pi = \sum_{i \in \mathcal{D}} \alpha^i + \left(1 - \prod_{i \in \mathcal{D}} \rho^i\right) \beta^G, \tag{13}$$

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. With a slight abuse of notation, we utilize the same notation to describe a single-agent policy that maps from $\mathcal{S}$ to $\mathcal{A}$ and an agent-specific user-independent policy that maps from $\mathcal{X}$ to $\mathcal{U}$, as both policies can be equivalently represented by agent i's search path $C^i(\pi^i) = (v_0^i, \dots, v_n^i)$. 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) and α^π being defined in Equation (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, that is, 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. We first introduce online heuristics for static (Section 4.1) and dynamic (Section 4.2) policy planning with short planning horizons, that is, with persistency station occupancy observations. We then explain how to relax this persistency assumption and to account for time-sensitive occupancy probabilities in Section 4.3, which allows to apply our algorithms to long planning horizons.

We note that the station occupancy–sensitive variant of our algorithms is clearly also applicable to short time horizons and will yield similar results. However, from a practical perspective, the station occupancy–sensitive implementation is computationally more costly, and should only be used if necessary.

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, that is, 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 4.1.5).

4.1.1 | No information sharing (D)

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 noncommunicating 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 10). As shown in Guillet et al. (2022), a multilabel setting algorithm with a heuristic dominance criterion, which we refer to as *l*, can efficiently solve this problem setting.

In general, a multilabel setting algorithm propagates partial policies to find a cost-optimal policy, maintaining a list of all explored and nondominated partial policies in cost-increasing order. A partial policy describes a given search path starting from the initial vertex v_0 and ending at v. We associate each partial policy 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 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 and propagates its 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

$$1 - \rho_v^i(\pi_1) \leq 1 - \rho_v^i(\pi_2)\,, \tag{14}$$

$$A_v^i(\pi_1) \leq A_v^i(\pi_2) \tag{15}$$

are true.

Finally, the labeling procedure returns the—nondominated—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 D in the Supporting Information. We solve each incoming request with this algorithm and refer to its sequential application as hierarchical label setting, denoted with *hl*.

4.1.2 | Occupancy observation sharing (DO)

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 *hl* algorithm for the no information-sharing setting to compute each agent's search path.

4.1.3 | Visit intentions sharing (DI)

In this setting, an agent i knows all station visit intentions of agents who started their search prior to i, that is, 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 9). Given i is the ith 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, that is, the joint set of all single-agent policies for agents in $\{1, \ldots, i-1\}$. Then, the probability of station v to be available to i results from Equation (8), by replacing $\mathcal{D}$ by $\tilde{\mathcal{D}}$

$$p_v(t) = p_v \prod_{j \in \tilde{\mathcal{D}}, j \neq i, t_v^j \leq t_v^i} \rho^j(t), \tag{16}$$

with $\rho^j(t)$ being defined in Equation (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 subsystem that includes herself and already planned policies π_i of other agents. We refer to this collaborative implementation as *hlc*. To optimize the agent's policy with respect to the subsystem cost, we compute n candidate policies for agent i using our multilabel setting algorithm *l*, which queues the evaluated policies by cost-increasing order. Of these candidate policies, we then select the policy that yields the lowest subsystem cost.

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

$$\pi^{i^*} = \underset{\pi^i \in \Gamma^i}{\arg\min}\, \alpha^{\pi_{i-1} \circ \{\pi^i\}}\,. \tag{17}$$

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 can be either shared or approximated.

4.1.4 | Intentions & occupancy observations sharing (DIO)

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 DO, we remove occupied stations from an agent's action space, and similar to DI, 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 (7) and (8).

4.2 | Dynamic policy planning

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

Focusing on a centralized decision maker (CIOd), 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 *hlc* 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 *lro* and note that it combines dynamic and offline planning similar to the work of Ulmer et al. (2019).

4.2.1 | 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 $s^\lambda = \text{d}$ to $s^\lambda = \text{r}$ 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 $\acute{x}^\text{t}$ and $\acute{x}^\text{f}$, 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 K decision epochs are reached. Note that the approximated cost-to-go $V^{\tilde{\pi}}(\acute{x})$ anticipates future system states based on the set of agents in the system in state x, and ignores unrevealed agent requests yet. Given the explicit values of $V^{\tilde{\pi}}(\acute{x}^\text{f})$ and $V^{\tilde{\pi}}(\acute{x}^\text{t})$, the best station $\pi(x) = v$ minimizes the cost-to-go (l.16–l.18) as

$$\pi(x) = \underset{u=(v)\in\mathcal{U}(x)}{\arg\min}\ d(x,u) + p_v V^{\tilde{\pi}}(\acute{x}^\text{f}) + (1-p_v)V^{\tilde{\pi}}(\acute{x}^\text{t}). \tag{18}$$

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

```
1: 𝒥 ← ∅, 𝒯 ← ∅, 𝒪 ← ∅, x ← 0, x ← (𝒥, 𝒯, 𝒪, x)
2: while True do
3:     if newEpochTriggered() then
4:         if 𝒥 ≠ ∅ ∧ s^λ == d then
5:             s^λ ← r, v^λ ← v*, t^λ ← t^λ + t_{v_k,v*}
6:         λ ← decidingAgent(), v^λ ← observedStation()
7:         if a_{v^λ} == 1 then
8:             𝒪.add(v^λ), 𝒯.add(λ), 𝒥.pop(λ), s^λ ← f
9:         else
10:            if λ ∉ 𝒥 then
11:                𝒥.add(λ), s^λ ← d
12:            v* ← 0, Q* ← ∞
13:            for v ∈ 𝒰(x) do
14:                for ẋ^f, ẋ^t ∈ T(x, v) do
15:                    V^f ← greedyC(ẋ^f), V^t ← greedyC(ẋ^t)
16:                    Q ← t_{v_k,v} + (1 − p_v)V^t + p_v V^f
17:                    if Q < Q* then
18:                        Q* ← Q, v* ← v
19:            if v* == 0 then
20:                s^λ ← t
```

FIGURE 2 Online rollout algorithm.

4.2.2 | Dynamic *hlc* algorithm (*lro*)

We now use the dynamic variant of the *hlc* algorithm to solve the CIOd 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, that is, 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

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()`, $v^\lambda \leftarrow$ `observedStat()`
7: Π.pop(π^λ)
8: **if** $a_{v^\lambda} == 1$ **then**
9: $\mathcal{O}$.add(v^λ), $\mathcal{T}$.add(λ), $\mathcal{J}$.pop(λ)
10: $s^\lambda \leftarrow$ f,Π.pop(π^λ)
11: **else**
12: **if** $\lambda \notin \mathcal{J}$ **then**
13: $\mathcal{J}$.add(λ), $s^\lambda \leftarrow$ d
14: Π.pop(π^λ)
15: $\pi^* \leftarrow 0$, $\alpha^* \leftarrow \infty$
16: $T \leftarrow \bar{T}^\lambda - t^\lambda$
17: **for** $\pi \in$ `getBestPaths`$(v^\lambda, T, \Pi, \mathcal{O})$ **do**
18: $\alpha \leftarrow$ `getCost`$(\Pi \circ \pi)$
19: **if** $\alpha < \alpha^*$ **then**
20: $\alpha^* \leftarrow \alpha$, $v^* \leftarrow \pi(x)$, $\pi^* \leftarrow \pi$
21: **if** $v^* == 0$ **then**
22: $s^\lambda \leftarrow$ t
23: **else**
24: Π.add(π^*)

FIGURE 3 Label-based heuristic (*lro*).

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 l that returns the n best candidate single-agent policies for λ. We then select policy π^* that minimizes the subsystem cost $\alpha^{\Pi \circ \pi^*}$ (l.16–l.18) as

$$\pi^* = \underset{\pi \in \Gamma^\lambda}{\arg\min}\, \alpha^{\Pi \circ \pi}, \tag{19}$$

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 currently in the system 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 $(v_0^\lambda, v_1^\lambda, \dots, v_n^\lambda)$, v^* corresponds to the first unvisited station, that is, $v^* := v_1^\lambda$.

In the decentralized decision-making settings with observation sharing (DOd), we compute an agent's search path using the algorithm developed for the static planning setting DO. 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 | Long-term planning horizon

In the following, we discuss how to relax the station occupancy persistency assumption. To this end, we distinguish occupancy observation information that corresponds to (i) stations observed as temporarily occupied (e.g., due to ICEing), (ii) available stations selected for charging by navigated drivers during their search, and (iii) stations being permanently blocked (e.g., due to connector deficiency).

In the first case, we assume that stations observed as occupied start to recover their availability immediately after the occupancy observation, following a time-dependent exponential function (cf. Guillet et al., 2022) defined as

$$p_v^r(\Delta_v^i) = p_v \left(1 - e^{-(\frac{\mu_v}{p_v})(\Delta_v^i)}\right), \tag{20}$$

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 denotes 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 (20). 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 the second case, we assume stations to be occupied during a fixed period of time corresponding to a car charging, and to recover their initial probability afterwards. 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 (20) as

$$p_v^r(\Delta_v^i) = p_v + (1 - p_v)\left(e^{-(\frac{\mu_v}{p_v})(\Delta_v^i)}\right). \tag{21}$$

Note that Equation (21) could also 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 nonselected stations.

In the third case, stations remain occupied, and occupancy observations do not outdate.

We can immediately apply such recovering probabilities in pure observation-sharing settings (DO, DOd) or in the centralized setting (CIOd). In practice, implementing these changes requires timestamped observations for all of the three formerly mentioned cases.

For intention-sharing settings (DI & DIO), we need to account for recovering probabilities when expressing user-dependent probabilities (cf. Equation 8). Here, we recursively express the probability $p_v^i(t)$ that a station v is available to user i at time t given that the last expected station visit may have been realized by driver $i-1$ at time t^{i-1} as

$$\begin{aligned} p_v^i(t) = {} & \rho^{i-1}(t^{i-1}) \cdot p_v^{i-1}(t^{i-1}) \\ & + \bar{\rho}^{i-1}(t^{i-1}) \cdot \bar{p}_v^{i-1}(t^{i-1}) \cdot p_v^r(t - t^{i-1}) \,. \end{aligned} \tag{22}$$

Here, the first term of Equation (22) accounts for the case that station v was available at t^{i-1} but not used by driver $i-1$ due to an already successfully completed search. The second term of the equation accounts for the case that $i-1$ visited v but observed it to be blocked, such that we account for a recovering availability probability afterwards. Note that when recovering probabilities are ignored (as it is the case for short planning horizons), Equation (22) reduces to Equation (8). We note that for DIO, we only account for intentions of drivers still searching at the departure time of the current deciding driver i.

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. 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.

5.1 | 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 $r^s \in \{100, 300, 700\}$ m for a total number of $N \in \{2, \dots, 10\}$ drivers. Additionally, we consider two different driver search radii of $\bar{S} = 1$ km and $\bar{S} = 2$ km. 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.

We analyzed the sensitivity of our algorithms with respect to the penalty term β^G in a preliminary study (see Appendix E.1 in the Supporting Information). 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, which 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 toward 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.

While our main results discussion focuses on the impact of coordination in a short planning horizon with an at-most 15 min departure time window, we further analyze the impact of relaxing our station occupancy persistency assumption by considering a larger departure time window of 180 min.

5.2 | Benchmark algorithms

To evaluate the performance of our online heuristics (*hl*, *hlc*, *ro*, *lro*), which we introduced in Section 4, we introduce two naive benchmarks to compare the quality performances of more advanced coordinated search methods. In D-*gr*, drivers visit the closest station independent of any other drivers, whereas in DO-*gr*, drivers visit the closest station, which has not yet been visited by any of the previously planned drivers.

Furthermore, we evaluate the quality performances of *ro* and *lro* in the centralized setting (CIOd). Since the MDP is too large to be solved optimally, we benchmark both policies against a myopic greedy algorithm CIOd-*gr* and the optimal solution obtained in a deterministic and offline setting, that is, a perfect-information setting (PI-*op*), as suggested in Powell (2009). To analyze the quality loss in the centralized setting due to not anticipating charging requests, we further implement our *ro* algorithm in a stochastic setting (OFF-*ro*) with requests known ahead of time.

In CIOd-*gr*, 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,

$$v^* = \arg\min_{v \in \bar{\mathcal{V}}} t_{v^\lambda, v} + (1 - p_v)\bar{\beta}\,, \tag{23}$$

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

In PI-*op*, we assume the charging demand, that is, all drivers' departure time and location, to be known for the

overall planning horizon. We then compute for each realization $k \in [0, \dots, 100]$ 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 $v \in \mathcal{V}_{\text{avail}}(k)$, which 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).

5.3 | 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

$$\hat{\alpha}^i = \frac{\sum_{k=0}^{100} \hat{t}^k + \hat{\delta}^k \bar{\beta}}{100}, \tag{24}$$

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

$$\hat{\rho}^i = \frac{\sum_{k=0}^{100} \hat{\delta}^k}{100}. \tag{25}$$

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

$$\hat{\alpha} = \sum_{i \in \mathcal{D}} \hat{\alpha}^i + \left(1 - \prod_{i \in \mathcal{D}} \hat{\rho}^i\right) \beta^G. \tag{26}$$

The quantity $\hat{\rho} = \prod_{i \in \mathcal{D}} \hat{\rho}^i$ describes the realized system success rate, that is, 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

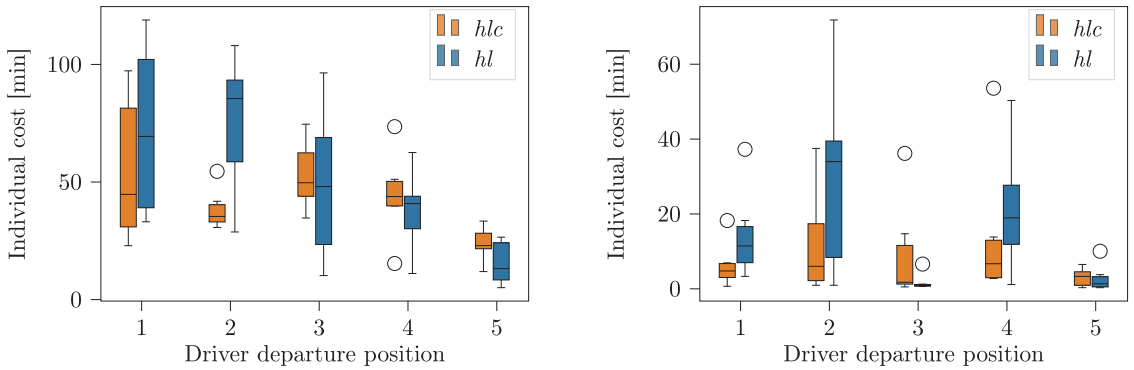
In the following, we first analyze the impact of collaboration in intention-sharing settings, before we focus on the impact of dynamic planning in observation-sharing settings. We then benchmark our algorithms for the centralized setting, and finally draw general conclusions on the performance of all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

6.1.1 | Collaboration in intention-sharing settings

To analyze the benefit of collaboration, we compare the performances of *hl* and *hlc* (see Section 4.1.3) in the DI setting, with respect to individual drivers' costs $\hat{\alpha}$ (see Equation 24) 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. Figure 4a,b shows the distribution of individual drivers' cost depending on their departure order, obtained with the collaborative algorithm (*hlc*), respectively, the noncollaborative algorithm (*hl*) 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, Figure 4c compares the difference between the highest and the lowest value (median, mean, 1st and 3rd quartiles, max, min) obtained for these individual drivers' cost distributions.

As can be seen in Figure 4, solutions obtained with *hlc* outperform the solutions obtained with *hl* on average with respect to both the system cost $\hat{\alpha}$ and the solution's fairness. Figure 4c shows that the departure position has a smaller impact on the results with than without collaboration. The difference between 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 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 *hlc* yields lower realized system cost $\hat{\alpha}$ than *hl* 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 E in the Supporting Information shows a similar trend for $t^s = 1$ min, but decreasing effects for larger departure time horizons of $t^s = 5$ 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*



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

		Δ(mean)	Δ(median)	Δ(max)	Δ(min)	Δ(q1)	Δ(q3)
low-25	hlc	26.7	31.2	63.9	22.7	22.3	53.2
	hl	60.1	72.3	92.4	28.0	50.2	78.0
	Δ_{ref} [%]	-56	-57	-31	-19	-56	-32
high-60	hlc	11.3	4.95	47.1	2.44	2.07	12.8
	hl	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 4 Comparison of *hl* and *hlc* in the DI setting for $t^s = 0$ min. *Note*: Each subplot shows for each driver i the distribution of the realized individual cost $\hat{\alpha}^i$ (see Equation 24) depending on her departure position, over all test instances that correspond to $t^s = 0$ min, $r^s \in \{100, 300, 700\}$ m, $\bar{S} \in \{1000, 2000\}$ m, 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 ith departure position. We compute Δ_{ref} as follows: $\frac{\Delta_{\text{HLH-c}}(m) - \Delta_{\text{HLH}}(m)}{\Delta_{\text{HLH}}(m)}$. [Color figure can be viewed at wileyonlinelibrary.com]

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 replanning solutions in the observation-sharing setting. Table 2 compares the system cost $\hat{\alpha}$ for DO-*hl* and DOd-*hl*, 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, DO even outperforms DOd 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 CIOd-*ro*'s and CIOd-*lro*'s policies to the greedy (CIOd-*gr*) and offline (PI-*op*) benchmark described in Section 5. Here, CIOd-*gr* provides an upper bound to CIOd-*ro* and CIOd-*lro*, which allows to analyze the performance gain by looking ahead (C-*ro* & C-*lro*) rather than myopically deciding (CIOd-*gr*). Contrary, PI-*op* 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. Appendix E in the Supporting Information further analyzes the quality loss due to not anticipating unseen charging requests ahead. Figure 5 shows the detailed distribution of the realized cost $\hat{\alpha}$ for CIOd-*lro*, CIOd-*gr*, and PI-*op*, for a varying number of drivers and search radii $\bar{S}$. Here, we note that we applied two-tailed Wilcoxon signed-rank tests to verify the significance of the average performance comparison and

TABLE 2 System cost comparison between static and dynamic decentralized policy planning.

	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 $\Delta[\%] = \frac{(\hat{\alpha}_{\text{dyn}} - \hat{\alpha}_{\text{stat}})}{\hat{\alpha}_{\text{stat}}}$ of the dynamic setting to the static setting for DO-*hl*. A negative Δ means that the dynamic setting outperforms the static counterpart.

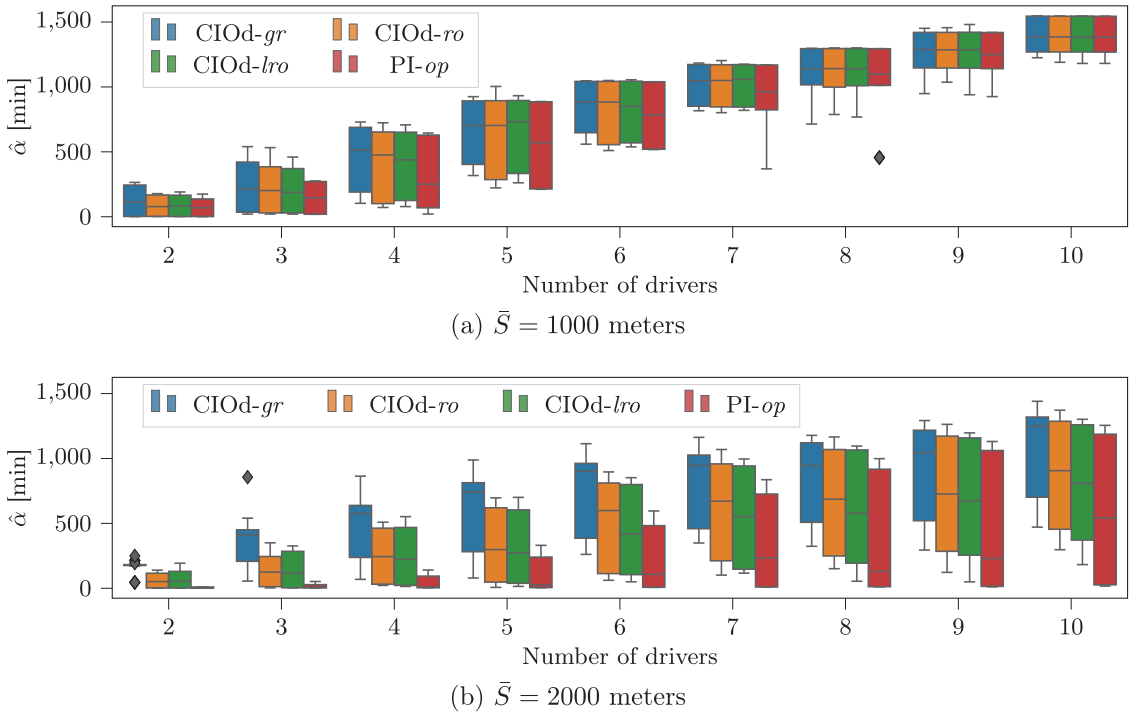


FIGURE 5 Distribution of the realized system cost $\hat{\alpha}$ for CIOd-*ro*, CIOd-*lro*, CIOd-*gr*, and PI-*op* per number of drivers, separated for small and large search areas. [Color figure can be viewed at wileyonlinelibrary.com]

obtained a *p*-value smaller or equal to $4.8 \cdot 10^{-2}$ for any comparison.

As can be seen, both CIOd-*ro* and CIOd-*lro* outperform the myopic policies (CIOd-*gr*) by decreasing cost $\hat{\alpha}$ by 14% (CIOd-*ro*) and 16% (CIOd-*lro*) on average.

The cost reduction depends however on the search radius $\bar{S}$. For small search areas ($\bar{S}$ = 1000 m), both *ro* and *lro* decrease the cost obtained with CIOd-*gr* by 2% on average, while the perfect-information setting PI-*op* yields a 9% cost decrease on average. We observe that the benefits of using CIOd-*ro*, CIOd-*lro*, or even PI-*op*, 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 nonmyopic settings achieve a significantly higher cost reduction of 29% (RO), 34% (*lro*), and 63% (PI-*op*) compared to G. According to *lro* and RO's performances, we decide to use *lro* in the CIOd setting.

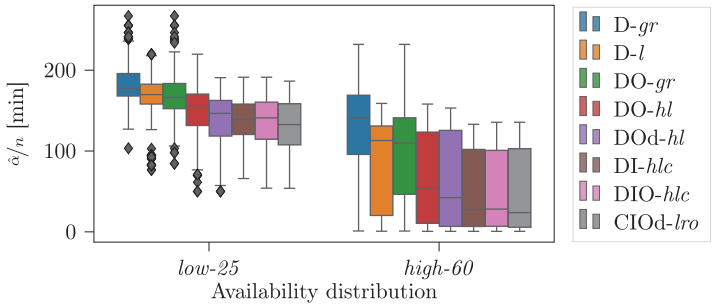


FIGURE 6 Average realized system cost $\hat{\alpha}$ for all information-sharing settings and both selfish settings, aggregated over all instances, per number of drivers. [Color figure can be viewed at wileyonlinelibrary.com]

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

6.1.4 | General performance evaluation

We compare the performances of all decentralized strategies, D-*l*, DI-*hlc*, DO-*hl*, DIO-*hlc*, and DOd-*hl*, the best centralized strategy CIOd-*lro*, and the two naive benchmarks D-*gr* and DO-*gr*. Figure 6 provides a comparison of all settings with respect to the system cost $\hat{\alpha}$, averaged over all test instances and divided by the respective number of drivers for the test instance.

As can bee seen, an advanced search strategy without coordination (D-*l*) already decreases the system cost obtained with a naive search strategy and no information sharing (D-*gr*) by 13% on average. With coordination, the DO-*hl* setting decreases the cost compared to the cost obtained with D-*gr* by 25% in a static setting and by 29% in a dynamic setting (DOd-*hl*), while DI-*hlc* and DIO-*hlc* lead to a 36% reduction, and CIOd-*lro* yields a 38% cost reduction. These average cost reductions decrease when comparing coordinated strategies against a greedy search strategy with observation sharing (DO-*gr*), but remain significant. Compared to DO-*gr*, DO-*hl* yields a 15% cost reduction, DOd-*hl* a 20% cost reduction, both DI-*hlc* and DIO-*hlc* a 28% cost reduction, and finally CIOd-*lro* yields a 30% cost reduction.

In the remainder, we ignore the two naive benchmarks due to inferior performance compared to the other algorithms. Accordingly, we now refer to a strategy, that is, the algorithm used in a given setting, only by the corresponding setting to keep the notation concise, that is, we refer to DI-*hlc* as DI or to D-*l* as D.

In-depth comparisons between these remaining uncoordinated and coordinated settings show that DOd decreases the cost obtained in D by 18% while both DI and DIO decrease it by 26%, and CIOd leads to a 28% decrease. The static observation-sharing setting DO 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 DI and DIO show performances very close to CIOd, which indicates that a decentralized and static setting performs nearly as well as a dynamic and centralized setting, if drivers share intentions.

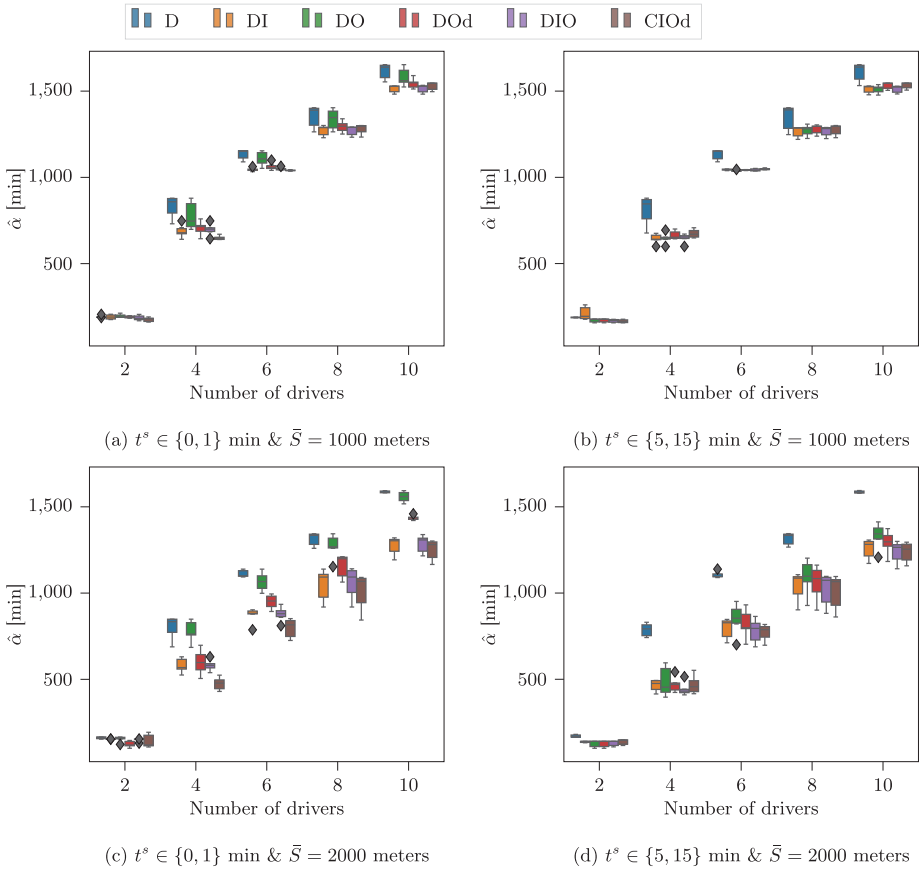
Result 4. *Compared to an uncoordinated setting* (*D*), *decentralized static intention sharing* (*DI, DIO*) *reduces* $\hat{\alpha}$ *on average by* 26%. *A centralized dynamic setting* (*CIOd*) *leads to a slightly higher cost reduction of* 28%.

Figure 7 compares D, DI, DO, DIO, DOd, and CIOd 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 the system cost $\hat{\alpha}$ for coordinated settings significantly decreases on average for a larger search area ($\bar{S} = 2000$ m) compared to a smaller search area ($\bar{S} = 1000$ m) in the best case (CIOd) by 23% in *low*-25 scenarios and by 80% in *high*-60 scenarios.

For short departure time horizons (see Figure 7a,c), we observe that pure intention sharing (DI) slightly outperforms combined observation- and intention sharing (DIO) for static policies, mostly when $\bar{S} = 2000$ m. In this case, DIO tends to provide lower individual cost solutions for the early departing drivers than DI 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, DI 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 (DOd) outperform static policies (DO).

For large departure time horizons (see Figure 7b,d), DIO improves over DI and even outperforms CIOd for smaller search areas. Notably, DO 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 DO slightly outperforms DOd in this case for small search areas ($\bar{S} = 1000$ m). We note that in this case, DO is well suited for a practical implementation as it has lower computational requirements than DIO or CIOd.



(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 7 Comparison of decentralized and centralized decision making in low-availability scenarios. [Color figure can be viewed at wileyonlinelibrary.com]

Figure 8 in Appendix E in the Supporting Information 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 (DOd) 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, that is,* $t^s \in \{0,1\}$ *min. In this case, observation sharing improves the early departing drivers' solution to the detriment of succeeding drivers, which worsens the total system performance.*

Result 6. *For larger departure time horizons, that is,* $t^s \in \{5,15\}$ *min, the decentralized observation-sharing setting (DO) performs similar as the decentralized information-sharing settings (DI, DIO), and as the centralized setting (CIOd).*

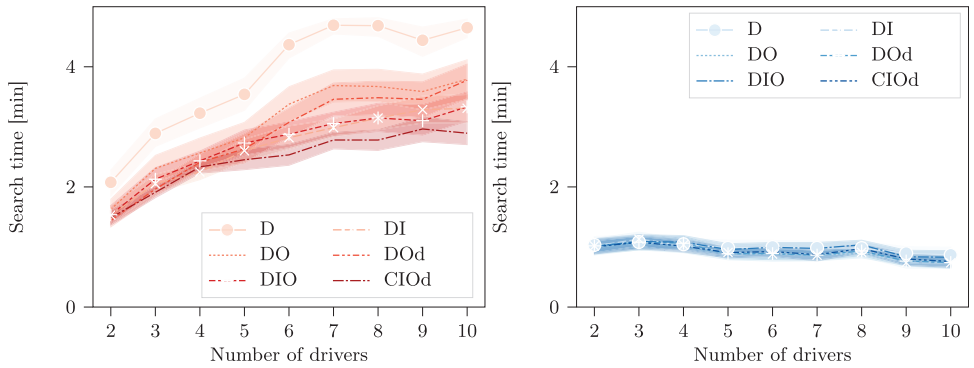
6.2 | Single-driver perspective

In the following, we evaluate how coordination impacts an individual driver's solution, before we analyze the impact of driver heterogeneity. Finally, we analyze the benefits of coordination in a longer planning horizon, while relaxing the persistent charging station occupancy assumption.

6.2.1 | Drivers' benefits of coordination

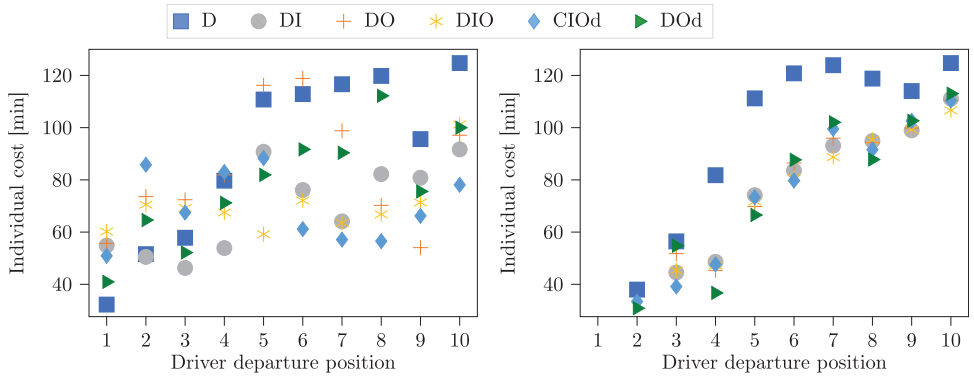
We refer to the uncoordinated solutions (D) as selfish solutions, in which a driver obtains her solution independently, without additional information and in her own best interest. First, we analyze the worst and best realized solutions obtained among all drivers for each test instance. For all decentralized settings (DI, DO, DIO, DOd) as well as for the centralized dynamic setting (CIOd), we compare the results to the uncoordinated setting (D). 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 8 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. 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



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

FIGURE 8 Distribution of the worst and best search times in the low-availability scenario. [Color figure can be viewed at wileyonlinelibrary.com]



(a) $t^s = 1$ min (b) $t^s = 15$ min

FIGURE 9 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). [Color figure can be viewed at wileyonlinelibrary.com]

search times. However, all coordinated settings decrease the maximal individual search times, in particular by 30% in DI and DIO, and by 35% in CIOd. Similarly analyzing performances with respect to individual success rates, 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%. We note that CIOd 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 9 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 10 drivers. Figure 9a shows results for a short departure time horizon ($t^s = 1$ min), while Figure 9b 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.*

Table 3 shows the relative individual search time deviation ($\hat{\Delta}t_{rel}$) and the absolute individual success rate deviation

TABLE 3 Individual search time and reliability improvements with coordination.

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

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

($\hat{\Delta}\rho$) 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% (DOd), 3% (CIOd), 8% (DO, DIO), and 9% (DI) of her search time, while she can increase her success rate by 0.05 (DO), 0.06 (DOd), and 0.09 (DI, DIO, CIOd). In line with the system-perspective evaluation, DI, DIO, and CIOd 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% (CIOd) or 12% (DI, DIO) of their search times on average, and up to 23% (DI, DIO) with a high number of drivers ($N = 9$). The success rate increase is limited to 0.05 on average (DI, DIO, CIOd). For larger search areas ($\bar{S}$ = 2000 m), the time gain decreases. Drivers may only save 5% of their search times on average (DI, DIO) or even slightly increase their search time by 2% (CIOd). In this case, however, the reliability gain significantly increases, with drivers increasing their success rates on average by 0.12 (DI, DIO) or 0.13 (CIOd), and up to 0.30 (CIOd with $N = 10$).

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 10 shows the impact of drivers with heterogeneous search radii in the CIOd setting by analyzing three cases: In the first case (10a), 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 (10b), drivers of both groups have a smaller search radius ($\bar{S}$ = 1000 m) while in the third case (10c), drivers of both groups have a larger search radius ($\bar{S}$ = 2000 m). Accordingly, drivers are heterogeneous in Figure 10a and homogeneous in Figure 10b,c. Analogously, Figure 11 shows the impact of smaller and larger time budgets. In the first case (11a), 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 (11b), all drivers have a smaller time budget, whereas in the third case (11c), 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, that is, have a smaller search radius (see Figure 10a) or a smaller time budget (see Figure 11a), obtain lower search times compared to 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.*

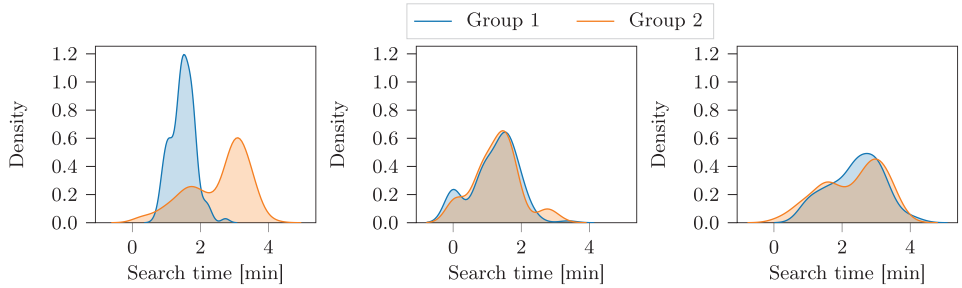


FIGURE 10 Impact of heterogeneous search radii on drivers' search times. [Color figure can be viewed at wileyonlinelibrary.com]

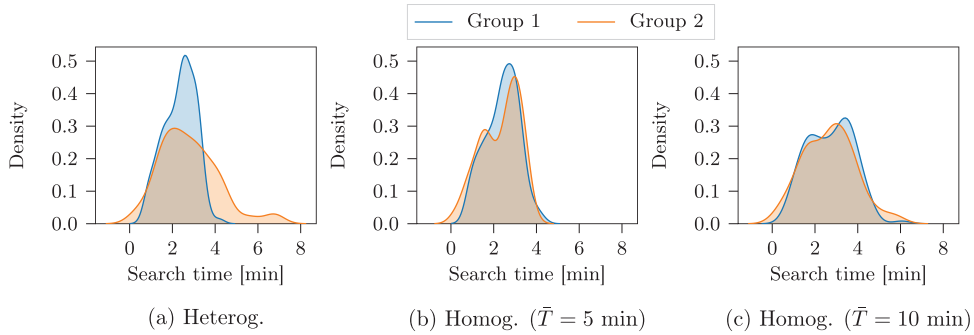


FIGURE 11 Impact of heterogeneous time budgets on drivers' search times. [Color figure can be viewed at wileyonlinelibrary.com]

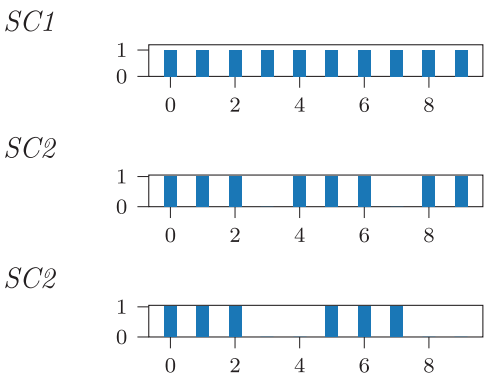


FIGURE 12 Requests distribution scenario. *Note*: The figure shows for each scenario the requests distribution pattern, with each (missing) bar corresponding to a (missing) request. [Color figure can be viewed at wileyonlinelibrary.com]

6.2.3 | Coordination in longer planning horizons

In the following, we analyze the impact of coordination in a longer planning horizon. To keep the discussion concise, we focus our analysis on the best performing strategy, that is, the CIOd-*lro*. We compare three charging demand distribution scenarios, see Figure 12. In *SC*1, drivers are steadily entering the system, with five requests per 15 min time interval. In *SC*2, each fourth request is skipped, whereas in *SC*3, 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 4 Relative cost improvement for CIOd over D.

Horizon	r^s	*SC*1	*SC*2	*SC*3
$t^s = 15$ 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: $\Delta[\%] = \frac{(\hat{\alpha}_{\text{CIOd}} - \hat{\alpha}_{\text{D}})}{\hat{\alpha}_{\text{D}}}$ of the CIOd setting to the D 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.

Table 4 compares the relative cost improvement obtained on average between coordinated (CIOd) and uncoordinated decentralized (D) planning in both short ($t^s = 15$ min) and long ($t^s = 180$ min) planning horizon settings, depending on the radius of the area in which all agents start their search ($r^s \in \{100, 300, 700\}$ m). Figure 13 further compares the average individual drivers cost obtained in the long planning horizon setting with coordination (CIOd) and without (D)

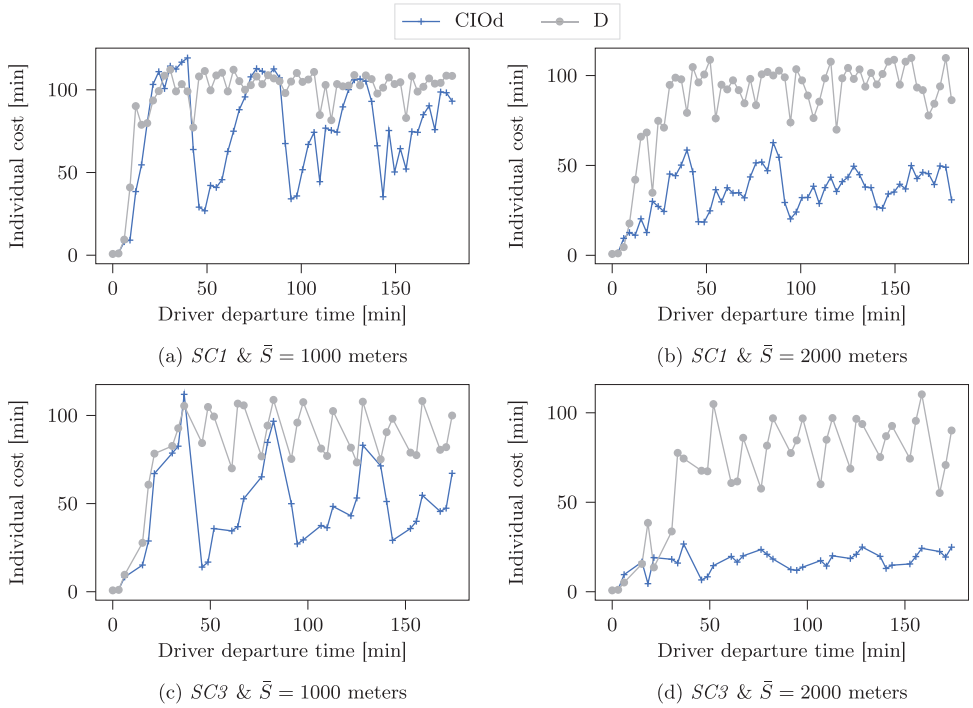


FIGURE 13 Comparison of D and CIOd in high-availability scenarios for a 3-h planning horizon. [Color figure can be viewed at wileyonlinelibrary.com]

depending on the driver's departure times, the search radius ($\bar{S} \in \{1000, 2000\}$ m) and the distribution scenario (*SC*1 or *SC*3).

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 4), our results show that cost savings in the long planning horizon setting decrease for a limited departure area ($r^s \in \{100, 300\}$ m), but increase when drivers are initially better distributed over the search area (i.e., $r^s = 700$ m). 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., *SC*3) 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 13 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 13). 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 individual cost than succeeding drivers. When these stations are freed after $\Delta T^v_{\text{charge}}$ min, 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 9 in Appendix E.5 in the Supporting Information shows complementary analyses for the low-availability scenario. Similar to short planning horizon results, the cost reduction obtained in CIOd compared to D decreases in low-availability scenarios, especially for a small search radius with $\bar{S} = 1000$ m.

7 | CONCLUSION

In this paper, we studied the multiagent 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, that is, drivers sharing their planned visits, observation sharing, that is, 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 compared to an uncoordinated scenario, 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. 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 positive effects of coordination in a 3-h planning horizon, with a centralized coordination strategy achieving 25% cost reduction on average compared to an uncoordinated scenario.

One comment on our study is in order. We assumed a nonadversarial 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 nonadversarial assumption opens a new avenue for further research in this context.

ACKNOWLEDGEMENTS

Open access funding enabled and organized by Projekt DEAL.

ORCID

Maximilian Schiffer https://orcid.org/0000-0003-2682-4975

REFERENCES

Al-Kanj, L., Nascimento, J., & Powell, W. B. (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., Hafner, D., Kellermeier, T., Krogmann, S., Razmjou, A., Krejca, M. S., Rothenberger, R., & Friedrich, T. (2016). Probabilistic routing for on-street parking search. In P. Sankowski & C. Zaroliagis (Eds.), *24th annual European symposium on algorithms (ESA 2016)* (vol. 57, pp. 6:1–6:13). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Bourgault, F., Furukawa, T., & Durrant-Whyte, H. F. (2003). Coordinated decentralized search for a lost target in a Bayesian world. In *Proceedings 2003 IEEE/RSJ international conference on intelligent robots and systems (IROS 2003) (Cat. No.03CH37453)* (vol. 1, pp. 48–53), Las Vegas, NV, USA.

Chung, T. H., & Burdick, J. W. (2008). Multi-agent probabilistic search in a sequential decision-theoretic framework. In *2008 IEEE international conference on robotics and automation* (pp. 146–151). Pasadena, CA, USA.

Dai, W., & Sartoretti, G. (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

Goodson, J. C., Thomas, B. W. T., & Ohlmann, J. W. (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., Hiermann, G., Kröller, A., & Schiffer, M. (2022). Electric vehicle charging station search in stochastic environments. *Transportation Science*, *56*(2), 483–500.

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

Jafari, E., & Boyles, S. D. (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. D., Cousineau, M., Goodson, J. C., & Mendoza, J. E. (2021). Dynamic ridehailing with electric vehicles. *Transportation Science*, *56*(3), 775–794.

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

Lauinger, D., Vuille, F., & Kuhn, D. (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

Myersdorf, D. (2020). *Ultra-fast charging batteries and the cure for EV charge anxiety*. https://www.intelligenttransport.com/transport-articles/112928/charging-anxiety/

Powell, W. B. (2009). What you should know about approximate dynamic programming. *Naval Research Logistics (NRL)*, *56*(3), 239–249.

Qin, K. K., Shao, W., Ren, Y., Chan, J., & Salim, F. D. (2020). Solving multiple travelling officers problem with population-based optimization algorithms. *Neural Computing and Applications*, *32*(16), 12033–12059.

Schmoll, S., & Schubert, M. (2018). Dynamic resource routing using real-time dynamic programming. In *Proceedings of the twenty-seventh international joint conference on artificial intelligence, IJCAI-18* (pp. 4822–4828).

Schuller, A., & Stuart, C. (2018). *From cradle to grave: e-mobility and the energy transition*. https://www.volkswagen-newsroom.com/en/stories/combatting-charging-anxiety-5107

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

Tang, H., Kerber, M., Huang, Q., & Guibas, L. (2013). Locating lucrative passengers for taxicab drivers. In *Proceedings of the 21st ACM SIGSPATIAL international conference on advances in geographic information systems* (pp. 504–507). Association for Computing Machinery.

Ulmer, M., Goodson, J. C., Mattfeld, D., & M, H. (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

Wong, E., Bourgault, F., & Furukawa, T. (2005). Multi-vehicle Bayesian search for multiple lost targets. In *Proceedings of the 2005 IEEE international conference on robotics and automation* (pp. 3169–3174). IEEE.

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

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Guillet, M., & Schiffer, M. (2023). Coordinated charging station search in stochastic environments: A multiagent approach. *Production and Operations Management*, *32,* 2596–2618. https://doi.org/10.1111/poms.13997