Multi-Agent Charging Station Search in Stochastic Environments

M. Guillet[a,b,*], M. Schiffer[a,c]

[a] TUM School of Management, Technical University of Munich, Munich, Germany
marianne.guillet@tum.de, schiffer@tum.de
[b] TomTom Location Technology Germany GmbH, Berlin, Germany
[c] Munich Data Science Institute, Technical University of Munich, Munich, Germany

Extended abstract submitted for presentation at the 11th Triennial Symposium on Transportation Analysis conference (TRISTAN XI)
June 19-25, 2022, Mauritius Island

April 1, 2022

Keywords: electric vehicle, Markov decision process, multi-agent systems, stochastic search

1 Introduction

Electric vehicles (EV) are a promising alternative to conventional cars to decarbonize the transportation sector, but the well-known range anxiety and the newer charge anxiety, caused by unreliable and insufficient public charging infrastructure, are hindering its adoption in the private market. A seamless charging experience may however reduce these anxieties if drivers can easily find and use an available charging station. In practice, existing commercial services help drivers to find available stations based on real-time availability data but struggle with data inaccuracy, e.g., due to conventional vehicles blocking the access to public charging stations. In this context, recent works have studied stochastic search methods that account for charging station availability uncertainty and can be embedded into today's navigation devices (cf. Guillet *et al.* , 2022). 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. So far, however, both practical and theoretical approaches ignore driver coordination enabled by charging requests centralization or sharing of data, e.g., observations of charging stations' availability status or visit intentions between drivers. With this work, we close this research gap by extending stochastic single-agent search algorithms to a stochastic multi-agent setting.

Related work on resources search problems with stochastic availability focus on a single-agent setting (cf. Guo & Wolfson, 2018, Arndt *et al.* , 2016, Schmoll & Schubert, 2018, Guillet *et al.* , 2022), whereas multi-agent settings mostly study cooperative and synchronous resource search problems with unknown resource locations (cf. Bourgault *et al.* , 2003, Wong *et al.* , 2005, Chung & Burdick, 2008, Dai & Sartoretti, 2020). In contrast, we focus on a non-adversarial multi-agent search problem, with stochastic charging station availability, where multiple EV drivers request to find an unoccupied charging station in their vicinity at the earliest possible time. Practically, each driver may either receive a full sequence of stations to visit until an available station is reached (static planning) or may receive the next station to visit (dynamic planning), recommended by her navigation device, that synchronizes with a central navigation service platform in both cases. The solution planning can be (i) centralized within the navigation service platform or (ii) decentralized, i.e., at agent-level. In the latter case, solution planning can happen directly within the local navigation device and devices use the platform only to share information with each other. In both cases, drivers may share their station occupancy observations intermittently

Table 1 – *Problem settings overview*

	information-sharing		settings characteristics			
	visit intentions	availability observations	path planning	decision-making	type	user-dependent solutions
DEC			static	decentralized	selfish	no
DEC-I	✓		static	decentralized	collaborative	no
DEC-O		✓	static	decentralized	informative	no
DEC-IO	✓	✓	static	decentralized	collaborative	no
CEN	✓	✓	dynamic	centralized	collaborative	yes
DEC-O-d		✓	dynamic	decentralized	informative	yes

or in real-time with the central platform. In the decentralized case, they may additionally share the planned charging station visits. To capture these varying characteristics which are of practical relevance, we introduce the following settings as summarized in Table 1, where the DEC setting corresponds to the baseline setting with uncoordinated searches.

The contribution of this work is three-fold. First, we model the underlying decision-making problem as a single decision-maker Markov Decision Process. Second, we present several online solution methods that allow to solve the settings introduced in Table 1 (i.e, with unknown requests). 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.

2 Methodology

Markov Decision Process: In the following, we first consider a centralized representation of the system states and represent the (offline) multi-agent search with an omniscient single decision-maker as a finite-horizon MDP.

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. We represent a system state $x \in \mathcal{X}$ out of state space $\mathcal{X}$ as $x = (\vec{x_d}, \mathcal{J}, \mathcal{T}, \mathcal{O})$, with $\mathcal{J}$ being the set of active agents, $\mathcal{T}$ being the set of successfully terminated agents, $\mathcal{O}$ 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 $x^i = (v^i, t^i, s^i)$ 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 $s^i \in \{\text{'d'}, \text{'f'}, \text{'t'}, \text{'r'}\}$ being the status of the agent: an agent can either (i) be en-route to the station (s^i = 'r'), unaware of v^i's realized availability, (ii) observe v^i to be available, which successfully terminates her search (s^i = 'f'), (iii) observe v^i to be occupied and has enough time to reach a new station (s^i = 'd') or (iv) not (s^i = 't'), which unsuccessfully terminates her search. The observation of v^i in (ii) and (iii) triggers a new decision epoch.

We denote with $u \in \mathcal{U}(x)$, the action taken in state x for agent λ. 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. We refer to state x as x^{S} if the station observed by λ is available: here, λ has successfully terminated its search and $d(x, u)$ corresponds to the cost for using the station. We refer to state x as x^{f} if λ observes an occupied station or begins her search. Here, $d(x, u)$ corresponds to the penalty cost if no station can be reached within λ's remaining time budget, or in the opposite case to the driving time to the newly selected station. We define a policy π as the state-action mapping function, such that $\pi(x) \in \mathcal{U}(x)$.

From state x, the system transitions to the next state $\acute{x}$ upon a single-agent action, with $\acute{\lambda}$ being the new deciding agent. The new state $\acute{x}$ can either be a successful state $\acute{x}^{\mathrm{S}}$ for $\acute{\lambda}$ with probability $p_{\acute{v}}$ (that station $\acute{v}$ is available) or an unsuccessful state $\acute{x}^{\mathrm{f}}$ with probability $1 - p_{\acute{v}}$.

We introduce the policy-specific cost function $V^{\pi}(x)$, that can be expressed as follows

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

with $x \in \{x^{\mathrm{s}}, x^{\mathrm{f}}\}$. Then, our objective is to find a policy π that minimizes the expected cost value $\alpha = V^{\pi}(x_0)$, with x_0 being the initial state.

Algorithmic framework: We develop online algorithms to process sequentially revealed charging requests, i.e, the set of agents $\mathcal{D}$ is initially empty and we update $\mathcal{D}$ each time a new charging request enters the system. Figure 1 describes both decentralized decision-making, i.e., agent-level information-sharing and planning, and centralized decision-making, i.e., system-level planning.

For decentralized decision-making settings, we plan each agent's search path using a modified version of the stochastic search algorithm developed in Guillet *et al.* (2022), denoted with HLH in its basic variant. To avoid the selfish use of shared visit intentions, we introduce the algorithm variant HLH-c, in which agents minimize their search times without compromising other agents' success. For static policy planning (i.e., DEC, DEC-I, DEC-O, DEC-IO), we compute an agent's search path only once, accounting for the latest available information, i.e., the latest shared visit intentions or the latest availability observations, according to the solved setting. For dynamic policy planning (i.e., in DEC-O-d), we re-compute the initially planned search path each time the agent visits an occupied station, using the latest observations shared by the agents.

For a centralized decision-making setting (i.e., CEN), we focus on dynamic policy planning and dynamically solve the large-scale MDP (with unknown requests), by using two different algorithms. The first algorithm is a rollout algorithm (RO) with a one-step decision rule as described in Goodson *et al.* (2017), which explores the MDP solution tree partially, using a base-policy to approximate the value of the policy-specific function. In each state, the algorithm selects the action that yields the lowest approximated cost. The second algorithm bases on a dynamic implementation of our HLH-c algorithm. 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 algorithm as LH-RO and note that it combines dynamic and offline planning similar to the work of Ulmer *et al.* (2019).

3 Numerical results and conclusion

We analyze the benefits of coordination between multiple agent's searches using an extensive case-study for the city of Berlin. We vary the radius of the departure area $r^s \in \{100, 300, 700\}$ meters for a total number of $N \in \{2, ..., 10\}$ drivers, and consider two different driver search

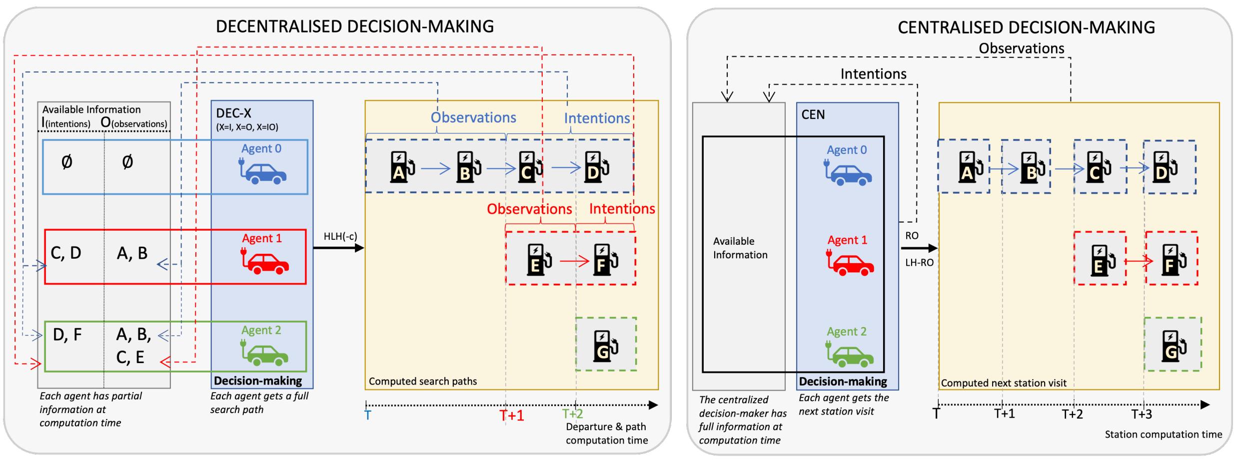

Figure 1 – *Online Algorithmic Framework*

radii of $\bar{S} = 1$ km and $\bar{S} = 2$ km. We equally distribute the drivers' search start time within a varying time horizon $t^s \in \{0, 1, 5, 15\}$ min, and let the search time budget be $\bar{T} = 5$ min for all drivers.

Our results show that coordination increases the system performance while individually benefiting each driver in general. Specifically, a centralized coordination strategy can decrease the system cost by 28%, and a static decentralized coordination strategy already achieves a 26% cost decrease if visit intentions are shared. In a decentralized setting with intention-sharing, observation-sharing does not increase the system's performance further for the analyzed planning horizon, but enforcing drivers' collaboration is required when drivers depart within a short time span. While a decentralized setting with only observation-sharing performs worse than intention-sharing settings, it provides a computationally efficient implementation in practice. When implemented in a dynamic setting, it yields a 10% cost decrease when drivers depart within a short time span, but achieves a 26% cost decrease with larger departure horizons. From a driver-perspective, coordination may save up to 23% of a driver's search time, while increasing her search reliability. Additionally, our results show that a coordinated search dominates uncoordinated searches, with respect to both best and worst solutions that an individual driver may obtain. Figure 2 shows the average relative individual time savings and absolute success rate increase for each coordinated setting compared to the uncoordinated setting DEC.

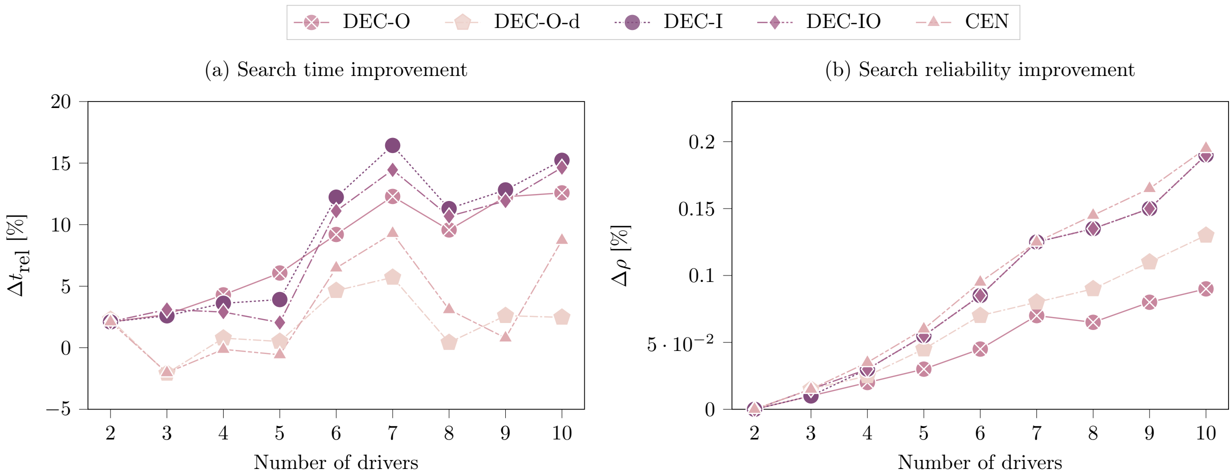

Figure 2 – *Benefits of coordination from a driver-perspective*

Subfigure (a) shows the average search time deviation Δt_{rel}, while Subfigure (b) shows the average success rate deviation $\Delta\rho$. Values are aggregated over all instances and computed as follows: $\Delta t_{rel} = \frac{1}{n}(\sum_{i=0}^{n} \frac{(\hat{t}^i_{setting} - \hat{t}^i_{DEC})}{\hat{t}^i_{DEC}})$ and $\Delta\rho = -\frac{1}{n}(\sum_{i=0}^{n} \hat{\rho}^i_{DEC} - \hat{\rho}^i_{setting})$, with n being the number of drivers considered in the respective instance.

References

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. *Pages 6:1–6:13 of: 24th Annual European Symposium on Algorithms (ESA 2016)*, vol. 57.

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).*

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.*

Dai, W., & Sartoretti, G. 2020. *Multi-Agent Search based on distributed Deep Reinforcement Learning.* Tech. rept. National University of Singapore.

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 (in press)*.

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

Schmoll, S., & Schubert, M. 2018 (7). Dynamic Resource Routing using Real-Time Dynamic Programming. *Pages 4822–4828 of: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*.

Ulmer, M., Goodson, J. C., Mattfeld, D., & M., Hennig. 2019. Offline–Online Approximate Dynamic Programming for Dynamic Vehicle Routing with Stochastic Requests. *Transportation Science*, **53**(1), 185–202.

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*.