Online Learning of Energy Consumption for Navigation of Electric Vehicles

Citation for the original published paper (version of record):
Åkerblom, N., Chen, Y., Haghir Chehreghani, M. (2023). Online Learning of Energy Consumption for Navigation of Electric Vehicles. Artificial Intelligence, 317.
http://dx.doi.org/10.1016/j.artint.2023.103879

(article starts on next page)

Artificial Intelligence 317 (2023) 103879

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint

Online learning of energy consumption for navigation of electric vehicles [☆]

Niklas Åkerblom [a,b,*], Yuxin Chen [c], Morteza Haghir Chehreghani [b]

[a] *Volvo Car Corporation, Gothenburg, Sweden*
[b] *Department of Computer Science and Engineering, Chalmers University of Technology, Gothenburg, Sweden*
[c] *Department of Computer Science, University of Chicago, Chicago, USA*

ARTICLE INFO

Article history:
Received 18 March 2022
Received in revised form 6 December 2022
Accepted 1 February 2023
Available online 3 February 2023

Keywords:
Energy efficient navigation
Online learning
Multi-armed bandits
Thompson Sampling

ABSTRACT

Energy efficient navigation constitutes an important challenge in electric vehicles, due to their limited battery capacity. We employ a Bayesian approach to model the energy consumption at road segments for efficient navigation. In order to learn the model parameters, we develop an online learning framework and investigate several exploration strategies such as Thompson Sampling and Upper Confidence Bound. We then extend our online learning framework to the multi-agent setting, where multiple vehicles adaptively navigate and learn the parameters of the energy model. We analyze Thompson Sampling and establish rigorous regret bounds on its performance in the single-agent and multi-agent settings, through an analysis of the algorithm under batched feedback. Finally, we demonstrate the performance of our methods via experiments on several real-world city road networks.

1. Introduction

Today, electric vehicles experience a fast-growing role in many different transport systems. However, the applicability of electric vehicles is often constrained by the limited capacity of their batteries. Due to the historically high cost of batteries, the range of electric vehicles has generally been much shorter than that of conventional vehicles. This has led to the fear of being stranded when the battery is depleted, an effect known as "range anxiety". Such concerns could be alleviated by improving the navigation algorithms and route planning methods for these systems. Therefore, in this paper we aim at developing principled methods for energy efficient navigation of electric vehicles.

Several works employ variants of shortest path algorithms for the purpose of finding the routes that minimize the energy consumption. Some of them (e.g., [2,3]) focus on computational efficiency in searching for feasible paths where the constraints induced by limited battery capacity are satisfied. Both [2] and [3] use energy consumption as edge weights for the shortest path problem. They also consider recuperation of energy modeled as negative edge weights, since they identify that negative cycles cannot occur due to the law of conservation of energy. In [3], a consistent heuristic function for energy consumption is used with a modified version of A*-search [4] to capture battery constraints at query-time. In [5], instead

[☆] This paper is an extended version of [1].

[*] Corresponding author at: Department of Computer Science and Engineering, Chalmers University of Technology, Gothenburg, Sweden.
E-mail address: niklas.akerblom@chalmers.se (N. Åkerblom).

https://doi.org/10.1016/j.artint.2023.103879

of using fixed scalar energy consumption edge weights, the authors use piecewise linear functions to represent the energy demand, as well as lower and upper limits on battery capacity.

This task has also been developed beyond the shortest path problems in the context of the well-known vehicle routing problem (VRP). In [6], VRP is applied to electrified commercial vehicles in a two-stage approach (i.e., an electric vehicle routing problem, EVRP), where the first stage consists of finding the paths between customers with the lowest energy consumption and at the second stage the EVRP including optional public charging station nodes is solved. The same authors later extend their methods in [7], using Bayesian methods to learn the energy consumption of individual road segments while solving the EVRP.

The aforementioned methods either assume that the necessary information for computing the optimal path is available, or do not provide enough exploration to acquire it. Thereby, we focus on developing an *online* framework to learn (explore) the parameters of the energy model adaptively alongside solving the navigation (optimization) problem instances. We adopt a Bayesian approach to model the energy consumption for each road segment. The goal is to learn the parameters of such an energy model to be used for efficient navigation. Therefore, we develop an online learning framework to investigate and analyze several exploration strategies for learning the unknown parameters.

Thompson Sampling (TS) [8], also called *posterior sampling* and *probability matching*, is a model-based exploration method for an optimal trade-off between exploration and exploitation. Several experimental [9–11] and theoretical studies [12–15] have shown the effectiveness of Thompson Sampling in different settings. [11] develops an online framework to explore the parameters of a decision model via Thompson Sampling in the application of interactive troubleshooting. In [16], Thompson Sampling is used for combinatorial semi-bandit problems, including the shortest path problem with Bernoulli-distributed edge costs, and distribution-dependent regret bounds are derived.

Upper Confidence Bound (UCB) [17] is another approach widely used for exploration-exploitation trade-off. A variant of UCB for combinatorial semi-bandits is introduced and analyzed in [18]. A Bayesian version of the Upper Confidence Bound method is introduced in [19] and later analyzed in terms of regret bounds in [20]. An alternative Bayesian approach is proposed in [21], which the authors call the Upper Credible Limit algorithm.

In this work, beyond the novel online learning framework for energy efficient navigation, we further extend our methods to the batched feedback and multi-agent settings. In the former, feedback from the environment is delayed and received in batches, while in the latter, multiple vehicles adaptively navigate and learn the parameters of the energy model. We then extensively analyze the proposed methods and evaluate them on several synthetic navigation tasks, as well as on real-world settings using SUMO-simulated traffic data from three different cities: Luxembourg [22], Monaco [23] and Turin [24].

1.1. Related work

The general problem considered in this paper is finding paths of minimum expected cost through graphs with unknown edge weight distributions. This problem has been studied using the framework of stochastic multi-armed bandits for at least a decade, where [25] and [26] are prominent examples of early work addressing the problem. The authors of [25] introduce a stochastic combinatorial bandit framework, where it is assumed that the weight of each edge in a traveled path is revealed afterwards (i.e., semi-bandit feedback). They propose a method called *Learning with Linear Rewards* based on the celebrated principle of optimism in the face of uncertainty, where paths are selected using an exploration bonus added to the estimated mean of each edge. Other works, also based on the utilizing this principle for combinatorial bandits and online shortest path problems, are [18] and [27].

Semi-bandit feedback is a natural assumption in our setting, since it is straightforward to record the actual energy consumed by a vehicle for each edge traversed. However, there are several methods for stochastic combinatorial bandits that do not need, nor utilize, this assumption. An example of such a method is [26], in which the authors leverage path interdependencies using barycentric spanners. Other examples include any algorithm for the linear stochastic bandit model (see e.g., [28,29]), of which combinatorial bandits with linear rewards is a special case.

Thompson Sampling has been analyzed and evaluated with promising results for combinatorial bandit problems in general (e.g., [30,31,16]), and the online shortest path problem is a commonly suggested application. The authors of [32] propose a framework for analyzing the Bayesian regret of Thompson Sampling, and apply it to several different problem settings. Their technique for converting regret bounds of UCB algorithms into bounds for Thompson Sampling is utilized in our work to derive bounds for batched feedback and multi-agent problem settings.

Bandit problems with delayed or batched feedback have been of intense interest to the research community, due to the wide applicability in real-world settings. Thompson Sampling has been empirically shown to achieve good results when reward observations are delayed [10]. The authors of [33] propose a *black box* algorithm which may convert any stochastic bandit algorithm into an algorithm handling delayed feedback. The converted algorithms retain the regret bounds of the original algorithms, except for an additive term which is constant in the horizon and linear in the maximum delay. In [34], a lower regret bound for the two-armed bandit problem with batched feedback is derived, again exhibiting a linear dependence with respect to the batch size.

We take inspiration from the frequentist analysis of batched linear contextual UCB presented in [35] and extended to the generalized linear setting in [36], utilizing a similar technique in our analysis to decompose the Bayesian regret over the batches. Another extension of [35] is [37], which presents a greedy LASSO-based algorithm for a *high-dimensional* batched linear contextual bandit setting, where the dimension of the context is assumed to be much higher than the time horizon.

To provide an upper bound for the frequentist regret, they assume that the context is stochastic with enough variance to induce sufficient exploration. This assumption does not hold for the non-contextual setting studied in our work.

Finally, regarding incremental learning of energy consumption in graphs, the authors of [7] use a Bayesian approach, similar to the one in this work, to learn the edge-specific distributions of electric vehicle energy consumption in a road network. They utilize the posterior distributions to formulate and solve an EVRP for commercial vehicles, where the paths between customers, charging stations and depots are selected using learned parameters and information from the environment. Since exploration is not the focus of their work, their method of calculating the shortest paths most closely corresponds to the greedy baseline used for the experiments in this work.

1.2. Our contributions

First and foremost, we propose a novel online learning framework for energy efficient navigation of electric vehicles, in a setting where the vehicle energy consumption of road segments is assumed to be stochastic and the corresponding distributions are unknown *a priori*. We utilize a physical model of vehicle energy consumption to assign the edge-specific parameters of prior distributions for Bayesian bandit algorithms, such as Thompson Sampling and BayesUCB, in order to intelligently guide necessary exploration towards reasonable paths. The multi-armed bandit problem can be seen as a resource allocation problem, and as such, bandit algorithms are most useful where there is a limited number of agents available for data collection.

While travel time in a road network is both stochastic and a common edge weight in shortest path problems, there is an abundance of travel time data available from various sources, e.g., from cellular devices. For vehicle energy consumption, however, there are factors limiting the number of agents. As energy consumption depends heavily on the specific vehicle type used, internal vehicle sensors are required for data collection. Furthermore, energy consumption also depends on the characteristics of the road traveled, like slopes, curvature, bumps, etc. Hence, it is a problem setting highly suited for Bayesian bandit algorithms.

While several works on Bayesian combinatorial bandit algorithms have been empirically evaluated using uninformative priors, it is less common with experiments where informative priors are used to explore combinatorial arm sets more efficiently. We not only utilize informative priors in our experiments, but also study the exploration of the road network through visual inspection of geospatial plots. Furthermore, we experimentally evaluate the robustness of the proposed framework to prior misspecification. We perform experiments for the road networks of multiple cities, using realistic traffic environment data.

As far as we are aware, there are no previous works analyzing the Bayesian regret of batched combinatorial Thompson Sampling. Furthermore, we also extend our analysis to the synchronous multi-agent setting. While there is prior work for batched linear contextual bandits (e.g., UCB in [35] and [36]), a combinatorial bandit problem is only a special case of the linear bandit problem for linear reward functions. Our technique, however, is feasible to extend for non-linear reward functions, such as in [38], where combinatorial Thompson Sampling is used to address the problem of finding paths which minimize their maximum edge weights.

Finally, this is the first work extending and evaluating the BayesUCB algorithm [19] to the online shortest path problem, empirically demonstrating good performance of the algorithm in this problem setting.

2. Energy consumption model

In this section, we start by describing how we model the road network and the different factors affecting the energy consumption of a vehicle traversing a specific road segment. We then outline two different Bayesian approaches to extend the deterministic energy consumption model to a probabilistic setting.

2.1. Setup of the energy consumption model

We model the road network by a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \boldsymbol{w})$ where each vertex $u \in \mathcal{V}$ represents an intersection of the road segments, and $\mathcal{E}$ indicates the set of directed edges. Each edge $e = (u_1, u_2) \in \mathcal{E}$ is an ordered pair of vertices $u_1, u_2 \in \mathcal{V}$ such that $u_1 \neq u_2$ and it represents the road segment between the intersections associated with u_1 and u_2. In the cases where bidirectional travel is allowed on a road segment represented by $(u_1, u_2) \in \mathcal{E}$, we add an edge $(u_2, u_1) \in \mathcal{E}$ in the opposite direction. A directed *path* is a sequence of vertices $\langle u_1, u_2, \ldots, u_n \rangle$, where $u_h \in \mathcal{V}$ for $h = 1, \ldots, n$ and $(u_h, u_{h+1}) \in \mathcal{E}$ for $h = 1, \ldots, n-1$. Hence, a path $\boldsymbol{p}$ can also be viewed as a sequence of edges. If $\boldsymbol{p}$ starts and ends with the same vertex, $\boldsymbol{p}$ is called a cycle. Note that, in this work, different paths may have different numbers of vertices.

We associate a weight vector $\boldsymbol{w}$ to the graph, where each element w_e represents the total energy consumed by a vehicle traversing edge $e \in \mathcal{E}$. We extend the notation so that the total weight of a path $\boldsymbol{p}$ is denoted $w_{\boldsymbol{p}} := \sum_{e \in \boldsymbol{p}} w_e$. For each edge e, we also define other edge attributes associated with road segments, such as the average speed v_e, the length l_e, and the inclination α_e.

In our setting, the amount of energy consumed at different road segments is stochastic and *a priori* unknown. We adopt a Bayesian approach to model the energy consumption at each road segment $e \in \mathcal{E}$, i.e., the edge weights. Such a choice provides a principled way to induce prior knowledge. Furthermore, as we will see, this approach fits well with the online learning and exploration of the parameters of the energy model.

We first consider a deterministic model of vehicle energy consumption E_e for an edge e, which will be used later as the prior. Similar to e.g., [39,7], our model is based on longitudinal vehicle dynamics and Newton's second law of motion. For convenience, we assume that vehicles drive with constant speed along individual edges so that we can disregard the longitudinal acceleration term. However, this assumption is only used for the prior. We then have the following equation for the approximated energy consumption (in watt-hours):

$$E_e := \frac{mgl_e \sin(\alpha_e) + mgC_r l_e \cos(\alpha_e) + 0.5C_d A\rho l_e v_e^2}{3600\eta}. \tag{2.1}$$

In Eq. (2.1) the vehicle mass m, the rolling resistance coefficient C_r, the front surface area A and the air drag coefficient C_d are vehicle-specific parameters. Whereas, the road segment length l, speed v and inclination angle α are location (edge) dependent. In principle, C_r could also be considered edge-specific (since it also depends on the surface of the road), but in this work, we assume that it is the same for all edges. We treat the gravitational acceleration g and air density ρ as constants. The powertrain efficiency η is vehicle specific and can be approximated by a constant $\eta = 1$ for an ideal vehicle with no battery-to-wheel energy losses.

Actual energy consumption can be either positive (traction) or negative (regenerative braking). If the energy consumption is modeled accurately and used as w_e in a graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \boldsymbol{w})$, the law of conservation of energy guarantees that there exists no cycle $\boldsymbol{c}$ in $\mathcal{G}$ where $w_{\boldsymbol{c}} < 0$. However, since we are modeling and estimating the expected energy consumption of each individual road segment independently (to ensure that the problem is tractable), this guarantee does not necessarily hold in our case.

While modeling energy recuperation is desirable from an accuracy perspective, it introduces some difficulties. In terms of computational complexity, Dijkstra's algorithm [40] does not allow negative edge weights and the Bellman-Ford algorithm [41–43] is slower by an order of magnitude. There are methods to overcome this (e.g., [44]), but they still assume that there are no negative edge weight cycles in the network. Hence, we choose to only consider positive edge weights when solving the energy efficient (shortest path) problem, which enables us to use Dijkstra's algorithm in this work. This approximation still achieves meaningful results, since even with recuperation discarded, edges with high energy consumption are avoided. So while the powertrain efficiency η has a higher value when the energy consumption is negative than when it is positive, we believe using a constant is a justified simplification as we only consider positive edge-level energy consumption in the optimization stage. However, we emphasize that our generic online learning framework is independent of such approximations, and can be employed with any senseful energy model and shortest path algorithm.

2.2. Rectified Gaussian model of energy consumption

Motivated by [45], as the first attempt at a probabilistic model of energy consumption, we assume the *stochastic* energy consumption $\tilde{E}_e$ of a road segment represented by an edge e follows a Gaussian distribution, given a certain small range of inclination, vehicle speed and acceleration. We also assume that $\tilde{E}_e$ is independent from $\tilde{E}_{e'}$ for all $e' \in \mathcal{E}$ where $e' \neq e$ and that we may observe negative energy consumption. In other words, we assume that we may *observe* the energy recuperation of the vehicle, even though we only use estimates of the non-negative energy consumption when solving the shortest path problem (as stated in Section 2.1). The likelihood function (where, for later convenience, $\tilde{E}_e$ is negated so that θ_e^* indicates a mean *reward*) is then

$$P(\tilde{E}_e \mid \theta_e^*, \sigma_e^2) := \mathcal{N}(-\tilde{E}_e \mid \theta_e^*, \sigma_e^2).$$

Here, for clarity, we assume the noise variance σ_e^2 is given. We can then follow a Bayesian approach, and use a Gaussian conjugate prior over the mean energy consumption:

$$P(\theta_e^* \mid \mu_{e,0}, \varsigma_{e,0}^2) := \mathcal{N}(\theta_e^* \mid \mu_{e,0}, \varsigma_{e,0}^2),$$

where we choose $\mu_{e,0} \leftarrow -E_e$ and $\varsigma_{e,0}^2 \leftarrow (\vartheta \mu_{e,0})^2$ for some constant $\vartheta > 0$. Due to the conjugacy properties, we have closed-form expressions for updating the posterior distributions with new observations of $\tilde{E}_e$. For any path $\boldsymbol{p}$ in $\mathcal{G}$, we have $\mathbb{E}\left[\sum_{e\in\boldsymbol{p}} \tilde{E}_e\right] = \sum_{e\in\boldsymbol{p}} \mathbb{E}[\tilde{E}_e]$, which means we can find the path with the lowest expected energy demand if we set $w_e \leftarrow \mathbb{E}[\tilde{E}_e]$ and solve the shortest path problem over $\mathcal{G}(\mathcal{V}, \mathcal{E}, \boldsymbol{w})$. When the expected energy consumption is estimated instead of being known, to deal with the risk of $w_e < 0$ (i.e., negative weights), we instead set $w_e \leftarrow \mathbb{E}[z_e]$ where z_e is distributed according to the rectified Gaussian distribution $\mathcal{N}^R(-\theta_e^*, \sigma_e^2)$, which is defined so that $z_e := \max(0, \tilde{E}_e)$ and $\tilde{E}_e \sim \mathcal{N}(-\theta_e^*, \sigma_e^2)$. The expected value of z_e is then $\mathbb{E}[z_e] = -(\theta_e^* \cdot (1 - \Phi(-\theta_e^*/\sigma_e)) + \sigma_e \cdot \phi(-\theta_e^*/\sigma_e))$, where Φ and ϕ are the standard Gaussian CDF and PDF respectively. Thus, since we observe both negative and positive energy consumption, we may utilize the conjugacy properties of the Gaussian likelihood and prior distribution to efficiently update and sample from the posterior distribution over the (non-negative) rectified Gaussian mean.

2.3. Log-Gaussian model of energy consumption

Alternatively, instead of assuming a rectified Gaussian distribution for the energy consumption of each edge, we model the non-negative edge weights by (conjugate) Log-Gaussian likelihood and prior distributions. By definition, if we have a Log-Gaussian random variable $Z \sim \mathcal{LN}(\mu, \sigma^2)$, then the logarithm of Z is a Gaussian random variable $(\log Z) \sim \mathcal{N}(\mu, \sigma^2)$. Therefore, we have the expected value $\mathbb{E}[Z] = \exp\{\mu + 0.5\sigma^2\}$ and the variance $\mathbf{Var}[Z] = (\exp\{\sigma^2\} - 1) \cdot \exp\{2\mu + \sigma^2\}$. We can then define the likelihood function as

$$P\left(\tilde{E}_e \middle| \theta_e^*, \sigma_e^2\right) := \mathcal{LN}\left(\tilde{E}_e \middle| \log(-\theta_e^*) - \frac{1}{2}\log\left(1 + \frac{\sigma_e^2}{\psi_e^2}\right),\ \log\left(1 + \frac{\sigma_e^2}{\psi_e^2}\right)\right), \tag{2.2}$$

such that we match the moments of the rectified Gaussian model as well as possible, with $\mathbb{E}[\tilde{E}_e] = -\theta_e^*$ and $\mathbf{Var}[\tilde{E}_e] = \sigma_e^2 \cdot \frac{(\theta_e^*)^2}{\psi_e^2}$. We also choose the prior hyper-parameters such that $\mathbb{E}[\theta_e^*] = \mu_{e,0}$ and $\mathbf{Var}[\theta_e^*] = \varsigma_{e,0}^2$, and also let $\psi_e = \mu_{e,0}$, where $\mu_{e,0}$ and $\varsigma_{e,0}$ are calculated in the same way as for the Gaussian prior (except that $\mu_{e,0}$ is restricted to be negative) in order to make fair comparisons between the Log-Gaussian and rectified Gaussian results. The resulting prior distribution is

$$P\left(\theta_e^* \middle| \mu_{e,0}, \varsigma_{e,0}^2\right) := \mathcal{LN}\left(-\theta_e^* \middle| \log(-\mu_{e,0}) - \frac{1}{2}\log\left(1 + \frac{\varsigma_{e,0}^2}{\mu_{e,0}^2}\right),\ \log\left(1 + \frac{\varsigma_{e,0}^2}{\mu_{e,0}^2}\right)\right). \tag{2.3}$$

We emphasize that the specific parameterization that we use for the Log-Gaussian model in Eq. (2.2) allows for closed form posterior updates with the prior distribution in Eq. (2.3). Since $-\theta_e$ is drawn from a Log-Gaussian prior distribution, then the value (i.e., a linear function of $\log(-\theta_e)$) of the first parameter of the Log-Gaussian likelihood described in Eq. (2.2) is Gaussian (i.e., the conjugate prior distribution of the first parameter using the standard parameterization). For more details on Bayesian updates with this Log-Gaussian parameterization, see e.g., [46]. We summarize the notation used in the preceding sections and the rest of the paper in Table A.1 of Appendix A.

3. Online learning and exploration of the energy model

We develop an *online learning* framework to explore the parameters of the energy model adaptively alongside sequentially solving the navigation (optimization) problem at different time steps. Here, a *time step* (or round) refers to each time we select and traverse a path. At the beginning, the exact energy consumption of the road segments and the parameters of the respective model are unknown. Thus, we start with an approximate and possibly inaccurate estimate of the parameters. We use the current estimates to solve the current navigation task. We then update the model parameters according to the observed energy consumption at different road segments (edges) of the navigated path, and use the new parameters to solve the next task.

Algorithm 1 describes these steps, where the vectors $\boldsymbol{\mu}_{t-1}$ and $\boldsymbol{\varsigma}_{t-1}$ refer to the current posterior parameters of the energy model for all the edges at the current time t, which are used to obtain the current edge weight vector $\boldsymbol{w}_t$. Whenever we refer to an element of a vector indexed by a time step t, we always let the *rightmost* index be t, e.g., $w_{e,t}$ in the vector $\boldsymbol{w}_t$. We solve the optimization problem using $\boldsymbol{w}_t$ to determine the optimal action (or *arm* in the nomenclature of multi-armed bandit problems) $\boldsymbol{a}_t$, which in this context is a path through a graph. The action $\boldsymbol{a}_t$ is applied and a reward $r_t(\boldsymbol{a}_t)$ is observed, consisting of the actual measured energy consumption for each of the passed edges. We assume that the energy consumption distribution of each edge is fixed over time, and therefore, we exclude the subscript t of the reward where it is not needed, such as for the expected reward $\mathbb{E}[r(\boldsymbol{a}_t)]$. Since we want to minimize energy consumption, we regard it as a negative reward when we update the parameters (shown for example for the rectified Gaussian model in Algorithm 2). T indicates the total number of time steps, sometimes called the horizon.

To measure the effectiveness of our online learning algorithm, we consider its regret, which is the difference in the total expected reward between always playing the optimal action and playing actions according to the algorithm. Formally, the instant regret at time t (or alternatively the *gap* of the action selected at time t) is defined as $\Delta_t := \mathbb{E}[r(\boldsymbol{a}^*)] - \mathbb{E}[r(\boldsymbol{a}_t)]$ where $\boldsymbol{a}^* := \arg\max_{\boldsymbol{a}} \mathbb{E}[r(\boldsymbol{a})]$ is the action which maximizes the expected reward, and the cumulative regret is defined as $\text{Regret}(T) := \sum_{t=1}^{T} \Delta_t$. Since our framework uses a Bayesian approach, we also consider *Bayesian regret*, which is the expected value of the regret over problem instances sampled from the prior distribution, so that $\text{BayesRegret}(T) := \mathbb{E}[\text{Regret}(T)]$.

3.1. Shortest path problem as multi-armed bandit

A combinatorial bandit [47,25] is a multi-armed bandit problem where an agent is only allowed to pull sets of arms instead of an individual arm. However, there may be restrictions on the feasible combinations of the arms. We consider the combinatorial semi-bandit case where the rewards are observed for each individual arm pulled by an agent during a round.

Algorithm 1 Online learning for energy efficient navigation.

```
Require: $\boldsymbol{\mu}_0, \boldsymbol{\varsigma}_0$
for $t \leftarrow 1, \ldots, T$ do
    $\boldsymbol{w}_t \leftarrow$ GetEdgeWeights$(t, \boldsymbol{\mu}_{t-1}, \boldsymbol{\varsigma}_{t-1})$
    $\boldsymbol{a}_t \leftarrow$ SolveOptimizationToFindAction$(\boldsymbol{w}_t)$
    $\boldsymbol{r}_t \leftarrow$ ApplyActionAndObserveReward$(\boldsymbol{a}_t)$
    $\boldsymbol{\mu}_t, \boldsymbol{\varsigma}_t \leftarrow$ UpdateParameters$(\boldsymbol{a}_t, \boldsymbol{r}_t, \boldsymbol{\mu}_{t-1}, \boldsymbol{\varsigma}_{t-1})$
```

Algorithm 2 Gaussian parameter update of the energy model.

```
procedure UpdateParameters$(\boldsymbol{a}_t, \boldsymbol{r}_t, \boldsymbol{\mu}_{t-1}, \boldsymbol{\varsigma}_{t-1})$
    for each edge $e \in \boldsymbol{a}_t$ do
        $\varsigma^2_{e,t} \leftarrow \left(\frac{1}{\varsigma^2_{e,t-1}} + \frac{1}{\sigma^2_e}\right)^{-1}$
        $\mu_{e,t} \leftarrow \varsigma^2_{e,t}\left(\frac{\mu_{e,t-1}}{\varsigma^2_{e,t-1}} + \frac{r_t(e)}{\sigma^2_e}\right)$
    return $\boldsymbol{\mu}_t, \boldsymbol{\varsigma}_t$
```

A number of different combinatorial problems can cast to multi-armed bandits in this way, among them the online shortest path problem [25–27] is the focus of this work. An efficient algorithm for the deterministic problem (e.g., [40]) can be used as an oracle [16] to provide feasible sets of arms to the agent, as well as to maximize the expected reward.

We connect this to the optimization problem in Algorithm 1, where we want to find an arm $\boldsymbol{a}_t$. At time t, let $\mathcal{G}(\mathcal{V}, \mathcal{E}, \boldsymbol{w}_t)$ be a directed graph with weight vector $\boldsymbol{w}_t$ and sets of vertices $\mathcal{V}$ and edges $\mathcal{E}$. Given a source vertex $u_1 \in \mathcal{V}$ and a target vertex $u_n \in \mathcal{V}$, let $\mathcal{P}$ be the set of all paths $\boldsymbol{p}$ in $\mathcal{G}$ such that $\boldsymbol{p} = \langle u_1, \ldots, u_n \rangle$. Assuming non-negative edge costs $w_{e,t}$ for each edge $e \in \mathcal{E}$, the problem of finding the shortest path (arm $\boldsymbol{a}_t$) from u_1 to u_n can be formulated as

$$\boldsymbol{a}_t = \arg\min_{\boldsymbol{p} \in \mathcal{P}} \sum_{e \in \boldsymbol{p}} w_{e,t}.$$

For the analysis, we introduce some formal definitions for this stochastic combinatorial semi-bandit problem. There is a set of *base arms* $\mathcal{A}$, which corresponds to $\mathcal{E}$ in the considered graph. The set of arms selected at time t is called the *super-arm* $\boldsymbol{a}_t \subseteq \mathcal{A}$. The set of feasible super-arms $\mathcal{I}$ such that $\boldsymbol{a}_t \in \mathcal{I}$, is equal to the set of paths $\mathcal{P}$. We further define the expected reward of super-arm $\boldsymbol{a}$ with respect to a particular mean reward vector (for all base arms) $\boldsymbol{\theta}$ as $f_{\boldsymbol{\theta}}(\boldsymbol{a}) := \sum_{i \in \boldsymbol{a}} \theta_i$. Hence, according to the previously introduced definition of regret, we have that $\text{Regret}(T) = \sum_{t=1}^{T} (f_{\boldsymbol{\theta}^*}(\boldsymbol{a}^*) - f_{\boldsymbol{\theta}^*}(\boldsymbol{a}_t))$.

3.2. Thompson Sampling

In our Bayesian setup, a greedy strategy chooses the arm which maximizes the expected reward according to the current estimate of the mean rewards. Since the greedy method does not actively explore the environment, there are other methods which perform better in terms of minimizing cumulative regret. One commonly used method is ϵ-greedy, where a (uniformly) random arm is taken with probability ϵ and the greedy strategy is used otherwise. While, in principle, it could possible to select paths uniformly at random for the exploration time steps, the size of the set of all paths $\mathcal{P}$ (corresponding to the set of feasible super-arms $\mathcal{I}$) can be exponential with respect to the number edges in the graph. This might even include paths similar to random walks through the graph, which would almost certainly be very inefficient in terms of accumulated edge costs. Hence, this method is not well suited to the shortest path problem. A modification of ϵ-greedy (based on Algorithm 1 in the supplementary material of [18]), where only a single edge (and the shortest path through it) is sampled, is introduced in Algorithm 7. However, for large graphs this might still lead to unreasonable exploration paths (e.g., a path between New York City and Boston through a randomly selected detour around Los Angeles).

An alternative method for exploration is Thompson Sampling (TS). In contrast to the greedy method, with TS (like in ϵ-greedy), arms are randomly sampled. However, where arms are sampled uniformly at random with ϵ-greedy, the TS agent samples from the model, i.e., during each time step, it selects an arm which has a high probability of being optimal by sampling mean rewards from the posterior distribution and choosing an arm which maximizes them. In other words, the method utilizes the prior beliefs about the parameter values to guide exploration towards reasonable arms.

Thompson Sampling for the energy consumption shortest path problem is outlined in Algorithm 3, where it can be used in Algorithm 1 to obtain the edge weights in the network (only shown for the rectified Gaussian model).

3.2.1. Regret analysis

In the following section, we provide an analysis on the cumulative regret of Thompson Sampling for the shortest path navigation problem. While better upper bounds on Bayesian regret for combinatorial TS is possible (e.g., using our proof for the batched combinatorial setting in Theorem 3 with batch size 1, we obtain a Bayesian regret upper bound of $\tilde{\mathcal{O}}\left(|\mathcal{E}|\sqrt{T}\right)$), this result may give some insight on the relationship between reinforcement learning problems and combinatorial bandit problems.

Algorithm 3 Thompson Sampling.

```
procedure GetEdgeWeights($t, \boldsymbol{\mu}_{t-1}, \boldsymbol{\varsigma}_{t-1}$)
    for each edge $e \in \mathcal{E}$ do
        $\tilde{\theta}_e \leftarrow$ Sample from posterior $\mathcal{N}(\mu_{e,t-1}, \varsigma^2_{e,t-1})$
        $w_{e,t} \leftarrow \mathbb{E}[z_e]$ where $z_e \sim \mathcal{N}^R(-\tilde{\theta}_e, \sigma_e^2)$
    return $\boldsymbol{w}_t$
```

Proposition 1. *The Bayesian regret of Algorithm 1 is upper bounded by*

$$\mathit{BayesRegret}(T) \le \tilde{\mathcal{O}}\left(|\mathcal{V}|^2\sqrt{|\mathcal{E}|\,T}\right).$$

We arrive at this result by relating the problem to recent results in reinforcement learning literature [15]. We view the online shortest path problem as an episodic reinforcement learning problem on an unknown finite time horizon Markov decision process (MDP). Here, each vertex $u \in \mathcal{V}$ corresponds to a state, each edge $e \in \mathcal{E}$ corresponds to an action, and the reward distributions for each action are the same as in the bandit problem. Like in the bandit problem formulation, the rewards of different states are assumed to be independent. Furthermore, given a state and an (allowed) action, transitions are deterministic, such that the next state is the end vertex of the edge corresponding to the action. Each episode starts in the source vertex state and ends when the target vertex state is reached. In other words, each episode corresponds a time step (and path selection) in the bandit problem formulation.

Applying posterior sampling for reinforcement learning (PSRL) like in [48], to this problem, using identical priors over reward distribution parameters as in the bandit problem, is equivalent to using TS on the combinatorial semi-bandit problem. At the start of each episode, PSRL samples an MDP from the current prior / posterior distribution over MDPs (here, a distribution over reward distributions, since the transitions are deterministic and known).

The policy used during this episode by PSRL is then the optimal policy with respect to the sampled MDP. In this problem, since the rewards are the negative edge weights of the graph, the shortest path between the source and target vertices will be selected.

Since the posterior parameters involved in PSRL are updated in the same way as in the bandit problem, identical observations and samples will lead to identical posterior updates. Hence, they are equivalent, and a regret bound for one will apply to the other. From [15], with τ being the episode length and T from the bandit problem corresponding to the number of episodes in the RL problem, we have

$$\begin{aligned}\text{BayesRegret}(T) &\le \tilde{\mathcal{O}}\left(\tau\sqrt{|\mathcal{V}|\,|\mathcal{E}|\,\tau T}\right)\\ &\le \tilde{\mathcal{O}}\left(|\mathcal{V}|^2\sqrt{|\mathcal{E}|\,T}\right).\end{aligned}$$

We also note that Conjecture 1 of [15] would improve this result so that

$$\text{BayesRegret}(T) \le \tilde{\mathcal{O}}\left(|\mathcal{V}|\sqrt{|\mathcal{V}|\,|\mathcal{E}|\,T}\right).$$

We note that the combinatorial semi-bandit problem formulation of Section 3.1 can be seen as a simpler special case of the reinforcement learning problem with less complexities to learn (e.g., less parameters to estimate, no state transitions modeling, etc.). In particular, whereas the traffic environment is affected by the paths that we choose, any state changes caused by an agent do not typically affect it immediately, since an edge is likely not traversed more than once during a single episode (path). If we want to adapt to the observed immediate rewards of different base arms while driving on a selected path, this could be modeled as a reinforcement learning problem, e.g., like the (#P-hard) *stochastic shortest path problem with recourse* [49], which may (in principle) then be addressed by PSRL. In general, however, choosing the less complex (though still meaningful) bandit problem formulation enables us to use powerful methods with proven strong performance guarantees.

3.3. Bayesian Upper Confidence Bound

Another class of algorithms demonstrated to work well in the context of multi-armed bandits is the collection of the methods developed around the Upper Confidence Bound (UCB). Informally, these methods are designed based on the principle of optimism in the face of uncertainty. The algorithms achieve efficient exploration by choosing the arm with the highest empirical mean reward added to an exploration term (the confidence width). Hence, the arms chosen are those with a plausible possibility of being optimal.

In [18] a combinatorial version of UCB (CUCB) is shown to achieve sub-linear regret for combinatorial semi-bandits. However, using a Bayesian approach is beneficial in this problem since it allows us to employ the theoretical knowledge on the energy consumption in a prior. Hence, we consider BayesUCB [19] and adapt it to the combinatorial semi-bandit setting. Similar to [19], we denote the quantile function for a distribution λ as $Q(\beta, \lambda)$, defined such that for a random variable distributed according to λ (s.t. $X \sim \lambda$), we have $\Pr(X \le Q(\beta, \lambda)) = \beta$. The idea of that work is to use upper quantiles of the

Algorithm 4 BayesUCB.

```
procedure GetEdgeWeights($t, \boldsymbol{\mu}_{t-1}, \boldsymbol{\varsigma}_{t-1}$)
    for each edge $e \in \mathcal{E}$ do
        $-\tilde{\theta}_e \leftarrow Q\left(\frac{1}{t}, \mathcal{N}(-\mu_{e,t-1}, \varsigma^2_{e,t-1})\right)$
        $w_{e,t} \leftarrow \mathbb{E}[z_e]$ where $z_e \sim \mathcal{N}^R(-\tilde{\theta}_e, \sigma_e^2)$
    return $\boldsymbol{w}_t$
```

posterior distributions of the expected arm rewards to select arms. If λ denotes the posterior distribution of a base arm and t is the current time step, the Bayesian Upper Confidence Bound (BayesUCB) for that base arm is $Q(1 - 1/t, \lambda)$.

This method is outlined in Algorithm 4 for the rectified Gaussian model. Here, since the goal is to minimize the energy consumption which can be considered as the negative of the reward, we use the *lower* quantile $Q(1/t, \lambda)$.

4. Multi-agent learning and exploration

The online learning may speed up via having multiple agents exploring simultaneously and sharing information on the observed rewards with each other. In our particular application, this corresponds to a fleet of vehicles of similar type sharing information about energy consumption across the fleet. Such a setting can be very important for road planning, electric vehicle industries, vehicle fleet operators and city principals.

The communication between the agents for the sake of sharing the observed rewards can be synchronous or asynchronous. In this paper, we consider the synchronous setting, where the vehicles drive concurrently in each time step and share their accumulated knowledge with the fleet before the next iteration starts. At each time step, each individual vehicle independently selects a path to explore/exploit according to the online learning strategies provided in Section 3. Here, we assume that all vehicles start their paths with the same source vertex and end them at the same target vertex, though even without this assumption, vehicles would benefit from information sharing as long as there is some overlap between selected paths. The vehicles share information synchronously, when all agents have finished their trips for a certain time step. During each time step, the agents are allowed to select paths which are overlapping (with shared edges), but we do not model any physical interactions between vehicles (e.g., how increased traffic intensity on those road segments affects energy consumption). However, this could be an interesting topic for future work.

Below, we provide two different regret bounds for TS-based multi-agent learning under the synchronous setting. Both are based on the idea of viewing the synchronous multi-agent problem as a single-agent problem with delayed feedback received in batches. Specifically, the *delay* corresponds to the number of vehicles in the fleet, since we wait for all of them to finish traversing their selected paths until we update the posterior distributions and start the next time step.

4.1. Thompson Sampling with queued delayed feedback

The first approach is based on the method of [33], which converts any algorithm for non-delayed stochastic bandit problems to an algorithm which handles delayed feedback, with a term constant in T added to the regret. This method and other similar queue-based methods have previously been used to adapt (and analyze) existing bandit algorithms for various problem settings with delayed feedback (see e.g., [50,51]). The approach of [33] is to wrap the original algorithm in an outer algorithm, which they call Queued Partial Monitoring with Delays (QPM-D). In essence, the inner algorithm functions as in the non-delayed case, unless the feedback of a selected arm is delayed and not available yet. In that case, the outer algorithm takes over and repeatedly plays the selected arm until feedback is received. Since the arm is played multiple times, excess delayed feedback, not immediately used by the inner algorithm, is also received. The outer algorithm stores the excess feedback in a queue data structure (where the order in which elements are inserted is also the order in which they are later retrieved, i.e., *First In, First Out*, or FIFO). This allows the inner algorithm to retrieve feedback from the queue the next time the arm is selected, instead of having to wait for delayed feedback. We outline QPM-D adapted to our problem in Algorithm 5.

Algorithm 5 QPM-D for Algorithm 1.

```
Create an empty Queue[$\boldsymbol{a}$] for each $\boldsymbol{a} \in \mathcal{I}$.
Let $\boldsymbol{b} \in \mathcal{I}$ be the first super-arm selected by Algorithm 1.
for $t \leftarrow 1, \ldots, T$ do
    Predict:
    while Queue[$\boldsymbol{b}$] is non-empty do
        Update Algorithm 1 with one reward from Queue[$\boldsymbol{b}$].
        Let $\boldsymbol{b}$ be the next super-arm selected by Algorithm 1.
    There are no queued rewards for $\boldsymbol{b}$, so perform arm $\boldsymbol{a}_t \leftarrow \boldsymbol{b}$ at time $t$ to receive rewards (possibly delayed) by the environment.
    Update:
    Let $\mathcal{D}_t$ be the set of (delayed) rewards received at time $t$ and each $(s, r_s(\boldsymbol{a}_s)) \in \mathcal{D}_t$ be the timestamped reward $r_s(\boldsymbol{a}_s)$ resulting from the arm $\boldsymbol{a}_s$ at
    time $s$.
    for $(s, r_s(\boldsymbol{a}_s)) \in \mathcal{D}_t$ do
        Add the reward $r_s(\boldsymbol{a}_s)$ to Queue[$\boldsymbol{a}_s$].
```

Theorem 2. *Let K be the number of agents, T be the horizon and $\text{Regret}_k(T)$ be the regret of each agent $k \in [K]$. In the synchronous multi-agent online shortest path setting (i.e., a fleet of K agents / vehicles working in parallel in each time step), the total fleet regret incurred by invoking Algorithm 5 satisfies $\sum_{k=1}^{K} \text{Regret}_k(T) \leq \mathcal{O}(|\mathcal{P}|K + \text{Regret}(TK))$.*

Proof. The result is obtained as a corollary of Theorem 6 in [33] which converts online algorithms for the non-delayed case to ones that can handle delays in the feedback (i.e., Algorithm 5), while retaining their theoretical guarantees. We consider the online shortest path problem as a standard stochastic bandit problem where the paths are the arms, and handle the multi-agent setting using Algorithm 5, like a sequential setting with delayed feedback. Let κ_t denote the feedback delay of the action at time t. Then according to [33] we have

$$\begin{aligned}\sum_{k=1}^{K} \text{Regret}_k(T) &\leq \text{Regret}(TK) + \sum_{\boldsymbol{p} \in \mathcal{P}} \mathcal{O}\left(\max_t \kappa_t\right) \\ &\leq \text{Regret}(TK) + \sum_{\boldsymbol{p} \in \mathcal{P}} \mathcal{O}(K) \\ &\leq \mathcal{O}(|\mathcal{P}|K + \text{Regret}(TK)). \quad \square\end{aligned}$$

While the additional first term of the regret is constant in T, it is also linear in $|\mathcal{P}|$, which may be exponential w.r.t. $|\mathcal{E}|$.

4.2. Thompson Sampling with batched feedback

In order to remove the exponential factor in Theorem 2, we outline a second approach. While the synchronous multi-agent setting *can* be cast as a delayed feedback problem, the general delay model is not actually necessary. Since the updates are synchronous, viewing it as a *batched* problem setting is sufficient. In this setting, rewards for selected arms are received periodically at fixed intervals, i.e., like *tumbling windows*. We note that this problem formulation can be useful beyond the multi-agent setting, e.g., in environments where feedback may be delayed due to wireless connection problems.

The regret analysis is not as straightforward as the one for Theorem 2. We combine ideas on batched bandit algorithms and analyses from [52], [35] and [36] with the general proof framework for deriving Bayesian regret bounds introduced by [32]. Before considering the multi-agent case, we start by outlining Thompson Sampling for the batched combinatorial semi-bandit setting in Algorithm 6. Here, we first consider a general stochastic combinatorial semi-bandit problem (i.e., not limited to the online shortest path problem) where rewards for each base arm $i \in \mathcal{A}$ are drawn from $\mathcal{N}\left(\theta_i^*, \sigma_i^2\right)$, with $\theta_i^* \sim \mathcal{N}\left(\mu_{i,0}, \varsigma_{i,0}^2\right)$ and finite (and known) variance σ_i^2. Also, we let B be the total number of batches, each of size K, such that $T = BK$. Furthermore, we denote the last time step in each batch $b \in [B]$ as t_b, i.e., $t_b = bK$. We also define the *history* H_t as the sequence of actions and rewards until time step t, such that $H_t = (\boldsymbol{a}_1, r_1(\boldsymbol{a}_1), \dots, \boldsymbol{a}_{t-1}, r_{t-1}(\boldsymbol{a}_{t-1}))$. Since the actions and rewards are random variables, H_t is a random variable as well. We denote a realization of H_t as H, i.e., a *fixed* history of actions and rewards.

Algorithm 6 Batched Thompson Sampling for combinatorial semi-bandits.

```
Require: Time horizon $T$, number of batches $B$, prior parameters $\boldsymbol{\mu}_0, \boldsymbol{\varsigma}_0$.
for $b \leftarrow 1, \dots, B$ do
    for $t \leftarrow t_{b-1} + 1, \dots, t_b$ do
        for $i \in \mathcal{A}$ do
            $\tilde{\theta}_i \leftarrow$ Sample from posterior $\mathcal{N}\left(\mu_{i,t_{b-1}}, \varsigma_{i,t_{b-1}}^2\right)$
        $\boldsymbol{a}_t \leftarrow \arg\max_{\boldsymbol{a} \in \mathcal{I}} f_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{a})$
        Play super-arm $\boldsymbol{a}_t$
    Observe batched rewards $r_{t_{b-1}+1}, \dots r_{t_b}$. Append corresponding arms and rewards to the history of selected super-arms and received rewards, such that $H_{t_b+1} = (\boldsymbol{a}_1, r_1(\boldsymbol{a}_1), \dots, \boldsymbol{a}_{t_b}, r_{t_b}(\boldsymbol{a}_{t_b}))$.
    Compute posterior parameters $\boldsymbol{\mu}_{t_b}, \boldsymbol{\varsigma}_{t_b}$ given the history $H_{t_b+1}$.
```

In this problem setting and algorithm, the rewards for all arms performed during a batch are received at the end of that batch. Hence, in each time step, parameters are sampled from the posterior distribution given the rewards observed at the end of the *previous batch*.

4.2.1. Regret analysis

We analyze the regret of this algorithm in the proof of Theorem 3, where

Theorem 3. *For Algorithm 6, with horizon T and batch size K, we have $\text{BayesRegret}(T) = \tilde{\mathcal{O}}(|\mathcal{A}|\ K + |\mathcal{A}|\sqrt{T})$.*

In order to prove Theorem 3, we need a few intermediary lemmas and assumptions. For base arm i, let $\hat{\theta}_{i,t}$ be the average reward of i until time step t, and $N_t(i)$ be the number plays of i until time step t.

Assumption 1. For each base arm $i \in \mathcal{A}$, the variance σ_i^2 is finite, and $\sigma_i^2 \leq 1$.

Since we assume that the variance σ_i^2 of each base arm $i \in \mathcal{A}$ is finite, we let, for convenience of notation, $\sigma_i^2 \leq 1$ for all $i \in \mathcal{A}$ (which can be achieved by scaling the feedback distributions of all base arms).

Assumption 2. Given the horizon T and the number of base arms $|\mathcal{A}|$, we have $T \geq |\mathcal{A}|$.

Assumption 3. Each base arm $i \in |\mathcal{A}|$ has been played once initially, such that $N_0(i) = 1$.

Assumptions 2 and 3 are mainly for convenience, to reduce the complexity of the proofs, whereas the finite variance assumption is needed for the concentration inequality we utilize in the proof of Lemma 7. We begin the analysis by defining upper and lower confidence bounds (for a super-arm $\boldsymbol{a}$ and history H_t, as defined in Algorithm 6):

$$U(\boldsymbol{a}, H_t) := f_{\hat{\boldsymbol{\theta}}_{t-1}}(\boldsymbol{a}) + \sum_{i \in \boldsymbol{a}} \sqrt{\frac{8 \log T}{N_{t-1}(i)}}$$

$$L(\boldsymbol{a}, H_t) := f_{\hat{\boldsymbol{\theta}}_{t-1}}(\boldsymbol{a}) - \sum_{i \in \boldsymbol{a}} \sqrt{\frac{8 \log T}{N_{t-1}(i)}}.$$

Using these definitions, we can decompose the regret in a way similar to [32] as follows:

Lemma 4. *Algorithm 6 has*

$$\begin{aligned} BayesRegret(T) = &\sum_{b=1}^{B} \sum_{t=t_{b-1}+1}^{t_b} \mathbb{E}\left[U(\boldsymbol{a}_t, H_{t_{b-1}+1}) - L(\boldsymbol{a}_t, H_{t_{b-1}+1})\right] + \\ &\sum_{b=1}^{B} \sum_{t=t_{b-1}+1}^{t_b} \mathbb{E}\left[L(\boldsymbol{a}_t, H_{t_{b-1}+1}) - f_{\boldsymbol{\theta}^*}(\boldsymbol{a}_t)\right] + \\ &\sum_{b=1}^{B} \sum_{t=t_{b-1}+1}^{t_b} \mathbb{E}\left[f_{\boldsymbol{\theta}^*}(\boldsymbol{a}^*) - U(\boldsymbol{a}^*, H_{t_{b-1}+1})\right]. \end{aligned}$$

Proof. By the definition of Bayesian regret, we have that:

$$\begin{aligned} &\text{BayesRegret}(T) \\ &= \sum_{t=1}^{T} \mathbb{E}\left[f_{\boldsymbol{\theta}^*}(\boldsymbol{a}^*) - f_{\boldsymbol{\theta}^*}(\boldsymbol{a}_t)\right] \\ &= \sum_{b=1}^{B} \sum_{t=t_{b-1}+1}^{t_b} \mathbb{E}\left[f_{\boldsymbol{\theta}^*}(\boldsymbol{a}^*) - f_{\boldsymbol{\theta}^*}(\boldsymbol{a}_t)\right] \end{aligned}$$

(Tower rule)

$$\begin{aligned} &= \sum_{b=1}^{B} \sum_{t=t_{b-1}+1}^{t_b} \mathbb{E}_{H \sim P\left(H_{t_{b-1}+1}\right)}\left[\mathbb{E}\left[f_{\boldsymbol{\theta}^*}(\boldsymbol{a}^*) - f_{\boldsymbol{\theta}^*}(\boldsymbol{a}_t) \,\middle|\, H_{t_{b-1}+1} = H\right]\right] \\ &= \sum_{b=1}^{B} \sum_{t=t_{b-1}+1}^{t_b} \mathbb{E}_{H \sim P\left(H_{t_{b-1}+1}\right)}\left[\mathbb{E}\left[U(\boldsymbol{a}_t, H) - U(\boldsymbol{a}_t, H) + f_{\boldsymbol{\theta}^*}(\boldsymbol{a}^*) - f_{\boldsymbol{\theta}^*}(\boldsymbol{a}_t) \,\middle|\, H_{t_{b-1}+1} = H\right]\right] \end{aligned}$$

(Conditioned on the history $H_{t_{b-1}+1}$, up to and including the last batch $b-1$, all super-arms $\boldsymbol{a}_t$ for $t = t_{b-1}+1, \ldots, t_b$ and the optimal super-arm $\boldsymbol{a}^*$ are identically distributed. We have that $\mathbb{E}\left[U(\boldsymbol{a}_t, H) \mid H_{t_{b-1}+1} = H\right] = \mathbb{E}\left[U(\boldsymbol{a}^*, H) \mid H_{t_{b-1}+1} = H\right]$, since U is a deterministic function of a super-arm and a history)

$$
\begin{aligned}
&= \sum_{b=1}^{B} \sum_{t=t_{b-1}+1}^{t_b} \mathbb{E}_{H \sim P\left(H_{t_{b-1}+1}\right)} \left[\mathbb{E} \left[U(\boldsymbol{a}_t, H) - U(\boldsymbol{a}^*, H) + f_{\boldsymbol{\theta}^*}(\boldsymbol{a}^*) - f_{\boldsymbol{\theta}^*}(\boldsymbol{a}_t) \,\middle|\, H_{t_{b-1}+1} = H \right] \right] \\
&= \sum_{b=1}^{B} \sum_{t=t_{b-1}+1}^{t_b} \mathbb{E}\left[U(\boldsymbol{a}_t, H_{t_{b-1}+1}) - L(\boldsymbol{a}_t, H_{t_{b-1}+1}) \right] \\
&+ \sum_{b=1}^{B} \sum_{t=t_{b-1}+1}^{t_b} \mathbb{E}\left[L(\boldsymbol{a}_t, H_{t_{b-1}+1}) - f_{\boldsymbol{\theta}^*}(\boldsymbol{a}_t) \right] \\
&+ \sum_{b=1}^{B} \sum_{t=t_{b-1}+1}^{t_b} \mathbb{E}\left[f_{\boldsymbol{\theta}^*}(\boldsymbol{a}^*) - U(\boldsymbol{a}^*, H_{t_{b-1}+1}) \right]. \quad \square
\end{aligned}
$$

To bound the last two terms of the decomposed Bayesian regret, we use the following lemma.

Lemma 5. *For any batch $b = 1, \dots, B$ and any time step $t = t_{b-1} + 1, \dots, t_b$, we have that*

$$
\begin{aligned}
&\mathbb{E}\left[L(\boldsymbol{a}_t, H_{t_{b-1}+1}) - f_{\boldsymbol{\theta}^*}(\boldsymbol{a}_t) \right] \leq \frac{2}{T} \\
&\mathbb{E}\left[f_{\boldsymbol{\theta}^*}(\boldsymbol{a}^*) - U(\boldsymbol{a}^*, H_{t_{b-1}+1}) \right] \leq \frac{2}{T}.
\end{aligned}
$$

Proof. Both $\mathbb{E}\left[L(\boldsymbol{a}_t, H_{t_{b-1}+1}) - f_{\boldsymbol{\theta}^*}(\boldsymbol{a}_t) \right] \leq \frac{2}{T}$ and $\mathbb{E}\left[f_{\boldsymbol{\theta}^*}(\boldsymbol{a}^*) - U(\boldsymbol{a}^*, H_{t_{b-1}+1}) \right] \leq \frac{2}{T}$ are proven in the same way, so we focus on the first inequality:

$$
\begin{aligned}
&\mathbb{E}\left[L(\boldsymbol{a}_t, H_{t_{b-1}+1}) - f_{\boldsymbol{\theta}^*}(\boldsymbol{a}_t) \right] \\
&= \mathbb{E}\left[f_{\hat{\boldsymbol{\theta}}_{t_{b-1}}}(\boldsymbol{a}_t) - f_{\boldsymbol{\theta}^*}(\boldsymbol{a}_t) - \sum_{i \in \boldsymbol{a}_t} \sqrt{\frac{8 \log T}{N_{t_{b-1}}(i)}} \right] \\
&= \mathbb{E}\left[\sum_{i \in \boldsymbol{a}_t} \hat{\theta}_{i,t_{b-1}} - \sum_{i \in \boldsymbol{a}_t} \theta_i^* - \sum_{i \in \boldsymbol{a}_t} \sqrt{\frac{8 \log T}{N_{t_{b-1}}(i)}} \right] \\
&= \mathbb{E}\left[\sum_{i \in \boldsymbol{a}_t} \left(\hat{\theta}_{i,t_{b-1}} - \theta_i^* - \sqrt{\frac{8 \log T}{N_{t_{b-1}}(i)}} \right) \right]
\end{aligned}
$$

(We let $[x]^+ := \max(0, x)$)

$$
\begin{aligned}
&\leq \mathbb{E}\left[\sum_{i \in \boldsymbol{a}_t} \left[\hat{\theta}_{i,t_{b-1}} - \theta_i^* - \sqrt{\frac{8 \log T}{N_{t_{b-1}}(i)}} \right]^+ \right] \\
&\leq \mathbb{E}\left[\sum_{i \in \mathcal{A}} \left[\hat{\theta}_{i,t_{b-1}} - \theta_i^* - \sqrt{\frac{8 \log T}{N_{t_{b-1}}(i)}} \right]^+ \right] \\
&= \sum_{i \in \mathcal{A}} \mathbb{E}\left[\left[\hat{\theta}_{i,t_{b-1}} - \theta_i^* - \sqrt{\frac{8 \log T}{N_{t_{b-1}}(i)}} \right]^+ \right] \\
&\leq \sum_{i \in \mathcal{A}} \mathbb{E}\left[\left[|\hat{\theta}_{i,t_{b-1}} - \theta_i^*| - \sqrt{\frac{8 \log T}{N_{t_{b-1}}(i)}} \right]^+ \right] \\
&= \sum_{i \in \mathcal{A}} \mathbb{E}\left[|\hat{\theta}_{i,t_{b-1}} - \theta_i^*| - \sqrt{\frac{8 \log T}{N_{t_{b-1}}(i)}} \,\middle|\, |\hat{\theta}_{i,t_{b-1}} - \theta_i^*| \geq \sqrt{\frac{8 \log T}{N_{t_{b-1}}(i)}} \right] \cdot \Pr\left\{ |\hat{\theta}_{i,t_{b-1}} - \theta_i^*| \geq \sqrt{\frac{8 \log T}{N_{t_{b-1}}(i)}} \right\}
\end{aligned}
$$

(By Lemma 6 and Lemma 7)

$$\leq \sum_{i \in \mathcal{A}} \frac{2}{T^2}$$

(By Assumption 2, $|\mathcal{A}| \leq T$)

$$\leq \frac{2}{T}. \quad \square$$

To bound the expected overestimation (or, correspondingly, underestimation) in the second-to-last inequality of the proof for Lemma 5, we derive two intermediate results in Lemma 6 and Lemma 7. For both of the lemmas, we let $\bar{\nu}_{i,x}$ be the average reward of base arm i over the first x times it has been played, i.e., contained in any played super-arm. In other words, for each batch $b \in [B]$ we have that $\hat{\theta}_{i,t_{b-1}} = \bar{\nu}_{i,N_{t_{b-1}}(i)}$. Additionally, for the proofs of both lemmas, we note that the average $\bar{\nu}_{i,x}$ is Gaussian with mean θ_i^* and variance σ_i^2/x. Since, by Assumption 1, we have that $\sigma_i^2 \leq 1$, this implies that $(\bar{\nu}_{i,x} - \theta_i^*)$ has mean 0 and variance ≤ 1.

Lemma 6. *For any batch $b \in [B]$ and base arm $i \in \mathcal{A}$, it holds that*

$$\mathbb{E}\left[|\hat{\theta}_{i,t_{b-1}} - \theta_i^*| - \sqrt{\frac{8 \log T}{N_{t_{b-1}}(i)}} \,\middle|\, |\hat{\theta}_{i,t_{b-1}} - \theta_i^*| \geq \sqrt{\frac{8 \log T}{N_{t_{b-1}}(i)}} \right] \leq 1.$$

Proof. We have that:

$$\mathbb{E}\left[|\hat{\theta}_{i,t_{b-1}} - \theta_i^*| - \sqrt{\frac{8 \log T}{N_{t_{b-1}}(i)}} \,\middle|\, |\hat{\theta}_{i,t_{b-1}} - \theta_i^*| \geq \sqrt{\frac{8 \log T}{N_{t_{b-1}}(i)}} \right]$$

(Tower rule)

$$= \mathbb{E}_{x \sim P\left(N_{t_{b-1}}(i)\right)}\left[\mathbb{E}\left[|\bar{\nu}_{i,x} - \theta_i^*| - \sqrt{\frac{8 \log T}{x}} \,\middle|\, |\bar{\nu}_{i,x} - \theta_i^*| \geq \sqrt{\frac{8 \log T}{x}} \wedge N_{t_{b-1}}(i) = x \right]\right]$$

($\mathbb{E}\left[\bar{\nu}_{i,x}\right] = \theta_i^*$, hence $\bar{\nu}_{i,x} - \theta_i^*$ is 0-mean Gaussian, and $\bar{\nu}_{i,x} - \theta_i^* \overset{d}{=} \theta_i^* - \bar{\nu}_{i,x}$)

$$= \mathbb{E}_{x \sim P\left(N_{t_{b-1}}(i)\right)}\left[\mathbb{E}\left[\bar{\nu}_{i,x} - \theta_i^* - \sqrt{\frac{8 \log T}{x}} \,\middle|\, \bar{\nu}_{i,x} - \theta_i^* \geq \sqrt{\frac{8 \log T}{x}} \wedge N_{t_{b-1}}(i) = x \right]\right]. \tag{4.1}$$

For any fixed integer $x > 0$, we have that $\left(\bar{\nu}_{i,x} - \theta_i^* - \sqrt{\frac{8 \log T}{x}}\right)$ is Gaussian with expected value $\left(-\sqrt{\frac{8 \log T}{x}}\right) < 0$. The inner expectation in Eq. (4.1) is the expected value of the corresponding truncated (below 0) Gaussian distribution, which (by, e.g., Theorem 2 of [53]) is increasing in $\left(-\sqrt{\frac{8 \log T}{x}}\right)$. Consequently,

$$\mathbb{E}\left[\bar{\nu}_{i,x} - \theta_i^* - \sqrt{\frac{8 \log T}{x}} \,\middle|\, \bar{\nu}_{i,x} - \theta_i^* - \sqrt{\frac{8 \log T}{x}} \geq 0 \right]$$

$$\leq \mathbb{E}\left[\bar{\nu}_{i,x} - \theta_i^* \,\middle|\, \bar{\nu}_{i,x} - \theta_i^* \geq 0 \right]$$

(Mean of truncated Gaussian, see [53], with $\mathbb{E}\left[\bar{\nu}_{i,x} - \theta_i^*\right] = 0$ and $\mathbf{Var}\left[\bar{\nu}_{i,x} - \theta_i^*\right] = \sigma_i^2/x$)

$$= \frac{\sigma_i}{\sqrt{x}} \frac{\phi(0)}{1 - \Phi(0)}$$

(By Assumption 1)

$$\leq \frac{\phi(0)}{1 - \Phi(0)} \approx 0.798$$

$$\leq 1. \tag{4.2}$$

The claim follows by bounding the inner expectation of Eq. (4.1) using Eq. (4.2). $\square$

Lemma 7. *$Pr\left\{ \exists b \in [B]\, \exists i \in \mathcal{A}, |\theta_i^* - \hat{\theta}_{i,t_{b-1}}| \geq \sqrt{\frac{8 \log T}{N_{t_{b-1}}(i)}} \right\} \leq \frac{2}{T^2}$.*

Proof. We perform a standard concentration analysis using union bounds and Hoeffding inequality, adapted for the batched feedback setting:

$$\Pr\left\{\exists b \in [B]\,\exists i \in \mathcal{A}, |\theta_i^* - \hat{\theta}_{i,t_{b-1}}| \geq \sqrt{\frac{8\log T}{N_{t_{b-1}}(i)}}\right\}$$

$$\leq \Pr\left\{\exists x \in [t_{B-1}]\,\exists i \in \mathcal{A}, |\theta_i^* - \bar{\nu}_{i,x}| \geq \sqrt{\frac{8\log T}{x}}\right\}$$

$$\leq \Pr\left\{\exists x \in [T]\,\exists i \in \mathcal{A}, |\theta_i^* - \bar{\nu}_{i,x}| \geq \sqrt{\frac{8\log T}{x}}\right\}$$

(Union bound)

$$\leq \sum_{x=1}^{T}\sum_{i\in\mathcal{A}} \Pr\left\{|\theta_i^* - \bar{\nu}_{i,x}| \geq \sqrt{\frac{8\log T}{x}}\right\}$$

(Hoeffding inequality for 1-subgaussian random variables, since $\theta_i^* - \bar{\nu}_{i,x}$ is Gaussian with mean 0 and variance ≤ 1, by Assumption 1)

$$\leq \sum_{x=1}^{T}\sum_{i\in\mathcal{A}} \frac{2}{T^4}$$

(By Assumption 2, $|\mathcal{A}| \leq T$)

$$\leq \frac{2}{T^2}. \quad \square$$

With the last two terms of the regret decomposition of Lemma 4 bounded using Lemma 5, we may focus on the first term. We can bound it in the following way:

Lemma 8. $\sum_{b=1}^{B}\sum_{t=t_{b-1}+1}^{t_b} \mathbb{E}\left[U(\boldsymbol{a}_t, H_{t_{b-1}+1}) - L(\boldsymbol{a}_t, H_{t_{b-1}+1})\right] \leq 4\sqrt{8\log T}\cdot(|\mathcal{A}|\,K + |\mathcal{A}|\sqrt{T})$.

Proof.

$$\sum_{b=1}^{B}\sum_{t=t_{b-1}+1}^{t_b} \mathbb{E}\left[U(\boldsymbol{a}_t, H_{t_{b-1}+1}) - L(\boldsymbol{a}_t, H_{t_{b-1}+1})\right]$$

$$= 2\sum_{b=1}^{B}\sum_{t=t_{b-1}+1}^{t_b} \mathbb{E}\left[\sum_{i\in\boldsymbol{a}_t}\sqrt{\frac{8\log T}{N_{t_{b-1}}(i)}}\right]$$

$$= 2\sum_{b=1}^{B}\sum_{t=t_{b-1}+1}^{t_b} \left(\mathbb{E}\left[\sum_{i\in\boldsymbol{a}_t}\left(\sqrt{\frac{8\log T}{N_{t_{b-1}}(i)}} - \sqrt{\frac{8\log T}{N_{t-1}(i)}}\right)\right] + \mathbb{E}\left[\sum_{i\in\boldsymbol{a}_t}\sqrt{\frac{8\log T}{N_{t-1}(i)}}\right]\right)$$

$$= 2\sum_{b=1}^{B}\sum_{t=t_{b-1}+1}^{t_b} \mathbb{E}\left[\sum_{i\in\boldsymbol{a}_t}\left(\sqrt{\frac{8\log T}{N_{t_{b-1}}(i)}} - \sqrt{\frac{8\log T}{N_{t-1}(i)}}\right)\right] + 2\sum_{t=1}^{T}\mathbb{E}\left[\sum_{i\in\boldsymbol{a}_t}\sqrt{\frac{8\log T}{N_{t-1}(i)}}\right].$$

The first term in the last expression above bounds the regret resulting from the batch delays, while the second term bounds the regret of the Thompson Sampling algorithm for the corresponding non-batched combinatorial semi-bandit setting. We start by bounding the first term:

$$2\sum_{b=1}^{B}\sum_{t=t_{b-1}+1}^{t_b} \mathbb{E}\left[\sum_{i\in\boldsymbol{a}_t}\left(\sqrt{\frac{8\log T}{N_{t_{b-1}}(i)}} - \sqrt{\frac{8\log T}{N_{t-1}(i)}}\right)\right]$$

$$\leq 2\sum_{b=1}^{B}\sum_{t=t_{b-1}+1}^{t_b} \mathbb{E}\left[\sum_{i\in\boldsymbol{a}_t}\left(\sqrt{\frac{8\log T}{N_{t_{b-1}}(i)}} - \sqrt{\frac{8\log T}{N_{t_b}(i)}}\right)\right]$$

$$\le 2\sum_{b=1}^{B}\sum_{t=t_{b-1}+1}^{t_b}\mathbb{E}\left[\sum_{i\in\mathcal{A}}\left(\sqrt{\frac{8\log T}{N_{t_{b-1}}(i)}}-\sqrt{\frac{8\log T}{N_{t_b}(i)}}\right)\right]$$

$$=2\sum_{b=1}^{B}\sum_{t=t_{b-1}+1}^{t_b}\sum_{i\in\mathcal{A}}\mathbb{E}\left[\left(\sqrt{\frac{8\log T}{N_{t_{b-1}}(i)}}-\sqrt{\frac{8\log T}{N_{t_b}(i)}}\right)\right]$$

$$=2K\sum_{b=1}^{B}\sum_{i\in\mathcal{A}}\mathbb{E}\left[\left(\sqrt{\frac{8\log T}{N_{t_{b-1}}(i)}}-\sqrt{\frac{8\log T}{N_{t_b}(i)}}\right)\right]$$

$$=2K\sum_{i\in\mathcal{A}}\mathbb{E}\left[\left(\sqrt{\frac{8\log T}{N_{t_0}(i)}}-\sqrt{\frac{8\log T}{N_{t_B}(i)}}\right)\right]$$

$$\le 2K\sum_{i\in\mathcal{A}}\mathbb{E}\left[\left(\sqrt{\frac{8\log T}{N_{t_0}(i)}}\right)\right]$$

(By Assumption 3, $N_{t_0}=1$)

$$=2K\sum_{i\in\mathcal{A}}\mathbb{E}\left[\left(\sqrt{8\log T}\right)\right]$$

$$=2K|\mathcal{A}|\sqrt{8\log T}$$

We can then continue by bounding the second term:

$$2\sum_{t=1}^{T}\mathbb{E}\left[\sum_{i\in\boldsymbol{a}_t}\sqrt{\frac{8\log T}{N_{t-1}(i)}}\right]$$

$$=2\sqrt{8\log T}\sum_{t\in[T]}\mathbb{E}\left[\sum_{i\in\boldsymbol{a}_t}\frac{1}{\sqrt{N_{t-1}(i)}}\right]$$

$$=2\sqrt{8\log T}\sum_{i\in\mathcal{A}}\mathbb{E}\left[\sum_{t:i\in\boldsymbol{a}_t}\frac{1}{\sqrt{N_{t-1}(i)}}\right]$$

(See the proof of Lemma 1 in [32])

$$\le 2\sqrt{8\log T}\sum_{i\in\mathcal{A}}\mathbb{E}\left[2\sqrt{N_T(i)}\right]$$

(Cauchy-Schwarz inequality)

$$\le 2\sqrt{8\log T}\cdot\mathbb{E}\left[2\sqrt{|\mathcal{A}|\sum_{i\in\mathcal{A}}N_T(i)}\right]$$

$$\le 2\sqrt{8\log T}\cdot\mathbb{E}\left[2\sqrt{|\mathcal{A}|^2T}\right]$$

$$=4|\mathcal{A}|\sqrt{8T\log T}.$$

This completes the proof of the lemma. □

With these lemmas, we can finish the proof of Theorem 3:

Proof of Theorem 3. We bound terms in the regret decomposition of Lemma 4 using Lemma 5 and Lemma 8, such that:

$$\begin{aligned}&\mathrm{BayesRegret}(T)\\&\le 4+4\sqrt{8\log T}\cdot(|\mathcal{A}|\,K+|\mathcal{A}|\sqrt{T})\\&\le\tilde{\mathcal{O}}(|\mathcal{A}|\,K+|\mathcal{A}|\sqrt{T}).\quad\square\end{aligned}$$

The result in Theorem 3 applies to a setting with unbounded Gaussian rewards. While general, it does not directly correspond to either of the models described in Section 2. However, it is straightforward to modify the proof so that it applies to a setting with rectified Gaussian base arm rewards (i.e., for a batched version of Algorithm 3).

Proposition 9. *The Bayesian regret of Algorithm 6, modified to sample arms as in Algorithm 3, with horizon T and batch size K, satisfies* $BayesRegret(T) = \tilde{O}(|\mathcal{A}|\ K + |\mathcal{A}|\sqrt{T})$.

Proof. Let $f^R_{\boldsymbol{\theta}}(\boldsymbol{a}) := -\sum_{i\in\boldsymbol{a}} \mathbb{E}_{z_i\sim\mathcal{N}^R(-\theta_i,\sigma_i^2)}[z_i]$ be the expected super-arm reward function for a combinatorial semi-bandit with rectified Gaussian base arm feedback. Note that, to connect the super-arm reward function to the rectified Gaussian model in Section 2.2 and the online shortest path problem formulation, we let base arm feedback be negative, with rectification above 0. The first term of the regret decomposition in Lemma 4 is bounded in Lemma 8 using only the confidence width term of the upper and lower confidence bounds, not involving the estimated expected super-arm rewards. Hence, under the assumption that we can use the same confidence bounds as in the (non-rectified) Gaussian setting, we only need to ensure that the bounds of the last two terms of the regret decomposition still hold. We can do this with a modification of the proof of Lemma 5.

$$\begin{aligned}
&\mathbb{E}\left[L(\boldsymbol{a}_t, H_{t_{b-1}+1}) - f^R_{\boldsymbol{\theta}^*}(\boldsymbol{a}_t)\right]\\
&= \mathbb{E}\left[f^R_{\hat{\boldsymbol{\theta}}_{t_{b-1}}}(\boldsymbol{a}_t) - f^R_{\boldsymbol{\theta}^*}(\boldsymbol{a}_t) - \sum_{i\in\boldsymbol{a}_t}\sqrt{\frac{8\log T}{N_{t_{b-1}}(i)}}\right]\\
&= \mathbb{E}\left[-\sum_{i\in\boldsymbol{a}_t}\mathbb{E}_{\hat{z}_i\sim\mathcal{N}^R(-\hat{\theta}_{i,t_{b-1}},\sigma_i^2)}\left[\hat{z}_i\right] + \sum_{i\in\boldsymbol{a}_t}\mathbb{E}_{z_i\sim\mathcal{N}^R(-\theta^*,\sigma_i^2)}[z_i] - \sum_{i\in\boldsymbol{a}_t}\sqrt{\frac{8\log T}{N_{t_{b-1}}(i)}}\right]\\
&= \mathbb{E}\left[\sum_{i\in\boldsymbol{a}_t}\left(-\mathbb{E}_{\hat{z}_i\sim\mathcal{N}^R(-\hat{\theta}_{i,t_{b-1}},\sigma_i^2)}\left[\hat{z}_i\right] + \mathbb{E}_{z_i\sim\mathcal{N}^R(-\theta^*,\sigma_i^2)}[z_i] - \sqrt{\frac{8\log T}{N_{t_{b-1}}(i)}}\right)\right]
\end{aligned}$$

(We have that $-\mathbb{E}_{\hat{z}_i\sim\mathcal{N}^R(-\hat{\theta}_{i,t_{b-1}},\sigma_i^2)}\left[\hat{z}_i\right] + \mathbb{E}_{z_i\sim\mathcal{N}^R(-\theta^*,\sigma_i^2)}[z_i] \le \hat{\theta}_{i,t_{b-1}} - \theta_i^*$, since $0 < \frac{\partial}{\partial\theta}\mathbb{E}_{z\sim\mathcal{N}^R(\theta,\sigma_i^2)}[z] < 1$)

$$\le \mathbb{E}\left[\sum_{i\in\boldsymbol{a}_t}\left(\hat{\theta}_{i,t_{b-1}} - \theta_i^* - \sqrt{\frac{8\log T}{N_{t_{b-1}}(i)}}\right)\right]. \tag{4.3}$$

After Eq. (4.3), the rest of the proof of Lemma 5 holds unmodified. Hence, the bound of Theorem 3 also holds in the case of rectified Gaussian base arm feedback. □

We can extend this result to the multi-agent online shortest path setting through the following corollary (where the set of edges $\mathcal{E}$ corresponds to the set of base arms $\mathcal{A}$ used throughout the proof of Theorem 3). We note that recently, a similar result has been derived in [54] for frequentist regret in a linear contextual bandit setting.

Corollary 10. *Let K be the number of agents, T be the horizon and $Regret_k(T)$ be the regret of each agent $k \in [K]$. In the synchronous multi-agent online shortest path setting (i.e., a fleet of K agents / vehicles working in parallel in each time step), the total fleet regret incurred by invoking Algorithm 6 satisfies* $\sum_{k=1}^K BayesRegret_k(T) \le \tilde{O}\left(|\mathcal{E}|\ K + |\mathcal{E}|\sqrt{TK}\right)$.

Proof. We prove this in the same way as the proof for Theorem 2, but use Theorem 3 instead of the result for QPM-D in [33]. □

For completeness, we also formally state the Bayesian regret upper bound mentioned in Section 3.2.1 as the following corollary of Theorem 3 and Proposition 9, with batch size 1.

Corollary 11. *The Bayesian regret of Algorithm 1 is upper bounded by*

$$BayesRegret(T) \le \tilde{O}\left(|\mathcal{E}|\sqrt{T}\right).$$

This corollary matches the bound from Proposition 3 of [32], which can be applied to any combinatorial semi-bandit problem with a linear super-arm reward function, when seen as a special case of the linear bandit problem. However, our analysis does not assume that the prior distributions have bounded support.

One way to discuss the optimality of the upper bounds derived in Theorem 3 and Proposition 9, is to compare them with existing lower bounds. To our knowledge, there is no established lower bound for the specific setting studied in this work (i.e., the batched feedback combinatorial semi-bandit problem). However, there are related bounds that one could either possibly derive a lower bound from, or discuss the upper bound in terms of. Perchet et al. derived a lower bound (Theorem 4 in [34]) for the excess regret due to the delay in the two-armed bandit problem, which is a special case of our problem. Furthermore, there are lower bounds for the (non-delayed) combinatorial semi-bandit problem (e.g., by Kveton et al., Proposition 2 in [55]), which induce a mandatory term in any lower bound for this problem. Combining these two will result in a lower bound to which the upper bound we derive in Theorem 3 is not tight in the excess regret term, since the upper bound includes a linear dependence on the number of base arms. We conjecture that it should also be possible to adapt the lower bound (for linear contextual bandits with adversarially generated contexts) by Ren et al. in Theorem 1 of [36], which includes a square-root factor (i.e., $\sqrt{|\mathcal{A}|}$ with the notation used in our work) for the excess regret term. It is notable that under both of these conjectured lower bounds, the $\tilde{O}\left(\sqrt{T}\right)$ term of our upper bound is optimal up to polylogarithmic factors.

5. Experimental results

In this section, we describe different experimental studies. For real-world experiments, we extend the simulation framework presented in [56] to network/graph bandits with general directed graphs, in order to enable exploration scenarios in realistic road networks. Furthermore, we add the ability to generate synthetic networks of specified size to this framework, in order to compare with the derived regret bounds (as the ground truth is provided for the synthetic networks). In all experiments, Dijkstra's algorithm is used to compute the shortest paths through the networks.

5.1. Real-world experiments

For the experiments in real-world road networks, we study one scenario with realistic energy consumption distributions handled by the agents using misspecified wide prior distributions, and another scenario where the prior distributions are completely known and utilized by the agents. In the second scenario, the parameters of the underlying energy consumption distributions are sampled from the prior distributions before each experiment run, whereas in the first scenario, the underlying distributions are fixed over multiple runs. Based on the second setting, we also consider a third setting where the energy consumption of different edges is correlated.

For each of the settings, we perform experiments using data from three cities: Luxembourg, Monaco and Turin. For Luxembourg, specifically, we study two problem instances (denoted #1 and #2) with different source and target vertices. We utilize, respectively for each of the cities, the Luxembourg SUMO Traffic (LuST) [22], Monaco SUMO Traffic (MoST) [23] and Turin SUMO Traffic (TuST) [24] scenarios to provide realistic traffic patterns and vehicle speed distributions for each hour of the day. This is used in conjunction with altitude data [57], and vehicle parameters from an electric vehicle. The resulting graph $\mathcal{G}$ for Luxembourg has $|\mathcal{V}| = 2247$ nodes and $|\mathcal{E}| = 5651$ edges, representing a road network with 955 km of highways, arterial roads and residential streets.

We use the default vehicle parameters provided for the energy consumption model in [6], with vehicle front surface area $A = 8\ \text{m}^2$, air drag coefficient $C_d = 0.7$ and rolling resistance coefficient $C_r = 0.0064$. The vehicle is a medium duty truck with vehicle mass $m = 14750$ kg, which is the curb weight added to half of the payload capacity.

We approximate the powertrain efficiency during traction by $\eta^+ = 0.88$ and powertrain efficiency during regeneration by $\eta^- = 1.2$. In addition, we use the constant gravitational acceleration $g = 9.81\ \text{m/s}^2$ and air density $\rho = 1.2\ \text{kg/m}^3$.

5.1.1. Prior distribution misspecified by agent

In this set of experiments, with results shown in Fig. 1 and Table 1, we study a scenario where agents do not have access to the true prior distributions of the environment. To simulate the ground truth of the energy consumption, we take the average speed v_e of each edge e from a full 24 hour scenario in each city road network. In particular, for LuST we observe the values during a peak hour (8 AM), with approximately 5500 vehicles active in the network. This hour is selected to increase the risk of traffic congestion, hence finding the optimal path becomes more challenging. We also get the variance of the speed of each road segment from the SUMO scenarios. Using this information, we sample the speed value for each visited edge and use the energy consumption model to generate the rewards for the arms.

For the probabilistic model, we assume σ_e to be proportional to E_e in Eq. (2.1), such that $\sigma_e^2 = (\varphi E_e)^2$, where we set $\varphi = 0.1$. For the prior distribution of an edge $e \in \mathcal{E}$, we misspecify it by using the speed limit of e as v_e, indicating that the real average speed is unknown. Then $\mu_{e,0} = -E_e$ and $\varsigma_{e,0}^2 = (\vartheta \mu_{e,0})^2$, where $\vartheta = 0.25$.

As a baseline, we consider the greedy algorithm for both the rectified Gaussian and Log-Gaussian models, where the exploration rule is to always choose the path with the lowest currently estimated expected energy consumption, similar to the recent method in [7].

We run the simulations for the BayesUCB, TS and greedy algorithms with a horizon of $T = 2000$ (i.e., $T = 2000$ time steps). Table 1 and Figs. 1b, 1d, 1f and 1h show the cumulative regret for the rectified Gaussian and Log-Gaussian models (indicated in all tables and figures with prefixes "N-" and "LN-", respectively, before the name of each algorithm), where

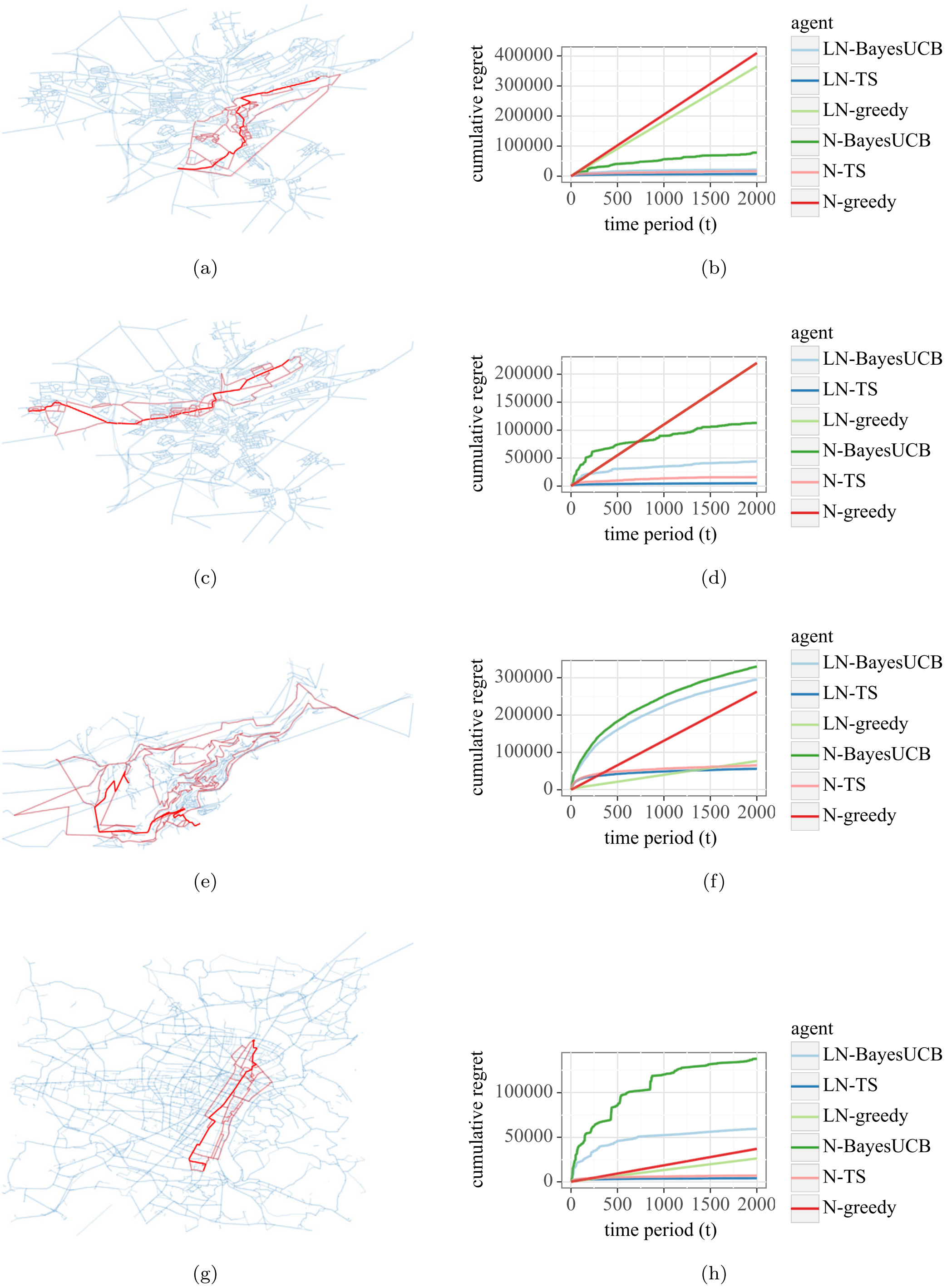


Fig. 1. Experimental results on the real-world road networks in the scenario where agents use misspecified priors. For Luxembourg #1, Luxembourg #2, Monaco and Turin, respectively, (a), (c), (e) and (g) show the exploration of Thompson Sampling in the road networks, where the red lines indicate the edges visited by the agent during exploration. Paths more frequently traveled are indicated with darker shades of red. Plots (b), (d), (f) and (h) show the average cumulative regret results for Thompson Sampling (TS), BayesUCB and probabilistic greedy algorithms, applied using rectified Gaussian (prefix N) and Log-Gaussian (prefix LN) energy consumption models. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Table 1
Average and standard deviation of regret at $T = 2000$ of agents with misspecified prior distributions. Bold average values indicate the agent with the lowest regret in each scenario.

City	Luxembourg #1		Luxembourg #2	
Agent	AVG	SD	AVG	SD
LN-BayesUCB	21514.64	469.83	43892.62	970.91
LN-TS	**7176.85**	780.50	**4995.93**	465.94
LN-greedy	364785.95	0.00	220100.48	0.00
N-BayesUCB	78264.31	3310.40	112856.00	1583.15
N-TS	16349.08	3082.75	16011.48	2135.49
N-greedy	409337.40	0.00	220100.48	0.00
City	Monaco		Turin	
Agent	AVG	SD	AVG	SD
LN-BayesUCB	295057.14	2750.97	59497.53	327.61
LN-TS	**56110.55**	1822.57	**4056.06**	566.55
LN-greedy	76880.85	25947.25	26217.88	17912.42
N-BayesUCB	329570.78	3748.02	138046.11	2033.91
N-TS	65407.80	4739.13	6938.36	556.36
N-greedy	262622.60	0.00	37024.61	0.00

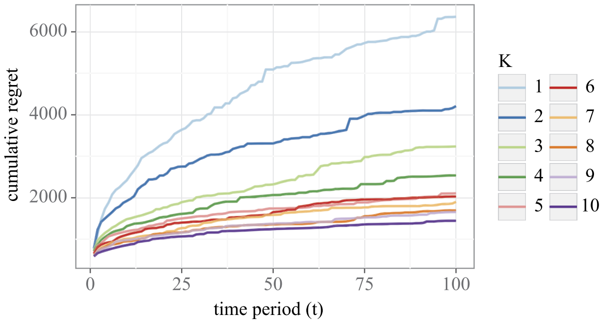


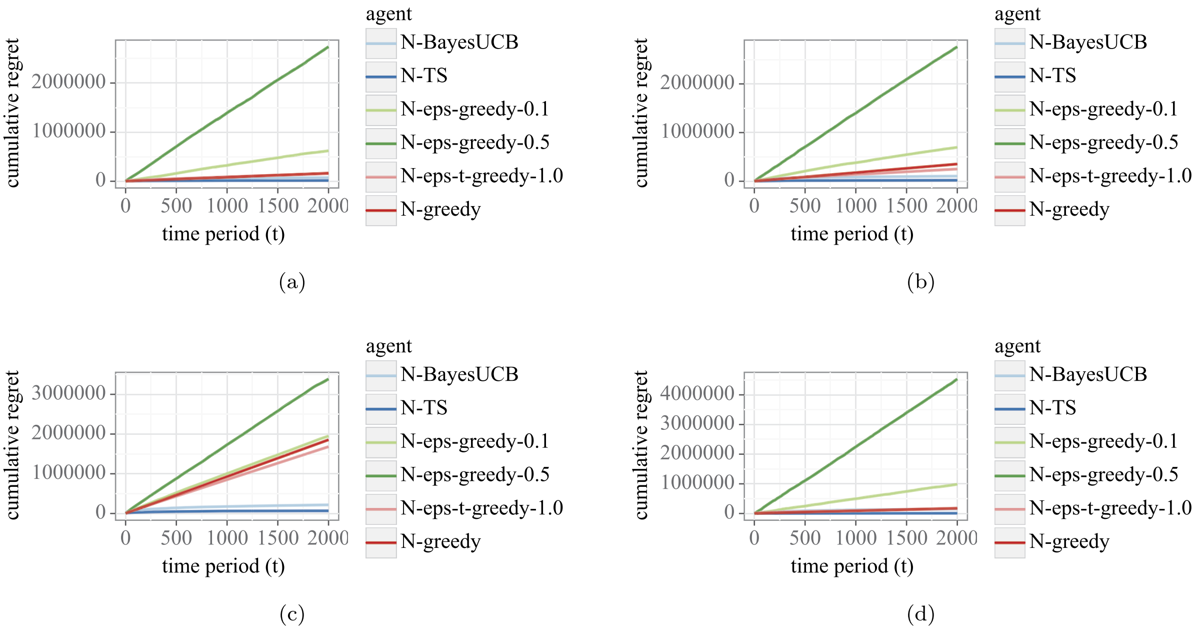
Fig. 2. Experimental results in the multi-agent setting, with each line showing the average cumulative regret (horizon $T = 100$) for each agent in fleets of size K, using Thompson Sampling.

the regret is averaged over 10 runs for each agent in each city. The intuition is that the energy saved by using the TS and BayesUCB agents instead of the baseline greedy agent is the difference in regret, expressed in watt-hours. It is clear that Thompson Sampling with the Log-Gaussian model has the best performance in terms of cumulative regret, but the other non-greedy agents also achieve good results. To illustrate that Thompson Sampling explores the road network in a reasonable way, Figs. 1a, 1c, 1e and 1g visualize the road network and the paths visited by this exploration algorithm in each city. Each plot displays all paths visited by the agent during a single experiment, where more frequently traveled paths are indicated with darker shades of red. We observe that in Figs. 1a, 1c and 1g, no significant detours are performed, in the sense that most paths are close to the optimal path. While there are some detours shown in Fig. 1e, we note that the distances in Monaco are small compared to the other cities, and that Fig. 1f indicates that the detours do not result in much additional regret.

For the multi-agent case, we use LuST and a horizon of $T = 100$ and 10 scenarios where we vary the number of concurrent agents by $K \in [1, 10]$ in each scenario. The cumulative regret averaged over the agents in these scenarios is shown in Fig. 2 for each K. In the figure, the final cumulative regret for each agent decreases sharply with the addition of just a few agents to the fleet. This continues until there are five agents, after which there seems to be diminishing returns in adding more agents. While there is some overhead (parallelism cost), just enabling two agents to share knowledge with each other decreases their average cumulative regret at $t = T$ by almost a third. This observation highlights the benefit of providing collaboration early in the exploration process, which is also supported by the regret bound in Corollary 10.

5.1.2. Prior distribution known by agent

In Section 5.1.1 we had realistic unknown energy consumption distributions (fixed across all experiment runs), handled by the agents using misspecified prior distributions. For the second set of experiments, with results shown in Fig. 3 and Table 2, we instead assume that the prior distributions are completely known by the agents. In other words, the environment samples the unknown mean vector $\boldsymbol{\theta}^*$ from the prior before all of the agents are applied to the problem instance specified by $\boldsymbol{\theta}^*$. Again, the regret results are averaged over 10 runs of each agent, in this setting resulting in an estimate of the Bayesian regret for each agent.



(a) (b) (c) (d)

Fig. 3. Experimental results on the real-world road networks in the scenario where agents use known priors. For Luxembourg #1, Luxembourg #2, Monaco and Turin, respectively, (a), (b), (c) and (d) show the average cumulative regret results for the Thompson Sampling (TS), BayesUCB, ϵ_t-greedy with fixed $\epsilon_t = 0.1$ (eps-greedy-0.1), $\epsilon_t = 0.5$ (eps-greedy-0.5), ϵ_t-greedy with decaying ϵ_t, and probabilistic greedy algorithms, with rectified Gaussian energy consumption models.

Table 2
Average and standard deviation of regret at $T = 2000$ of agents with known prior distributions. Bold average values indicate the agent with the lowest regret in each scenario.

City	Luxembourg #1		Luxembourg #2	
Agent	AVG	SD	AVG	SD
N-BayesUCB	72712.10	5476.08	105222.10	7097.73
N-TS	**15062.19**	3711.03	**20747.41**	3942.83
N-eps-greedy-0.1	621259.95	93672.80	697639.05	115938.33
N-eps-greedy-0.5	2737852.81	97107.13	2762055.80	160802.01
N-eps-t-greedy-1.0	155814.50	117049.01	250879.22	224149.22
N-greedy	164903.85	119811.93	353713.63	307249.42
City	Monaco		Turin	
Agent	AVG	SD	AVG	SD
N-BayesUCB	217256.84	70577.34	147010.93	8855.33
N-TS	**67054.53**	53218.99	**7874.74**	3559.68
N-eps-greedy-0.1	1949999.04	2840684.71	980895.00	134349.00
N-eps-greedy-0.5	3385113.34	2767510.10	4532613.64	201149.99
N-eps-t-greedy-1.0	1682411.76	3029371.15	167010.98	138348.82
N-greedy	1856111.47	2954720.83	178214.82	186056.16

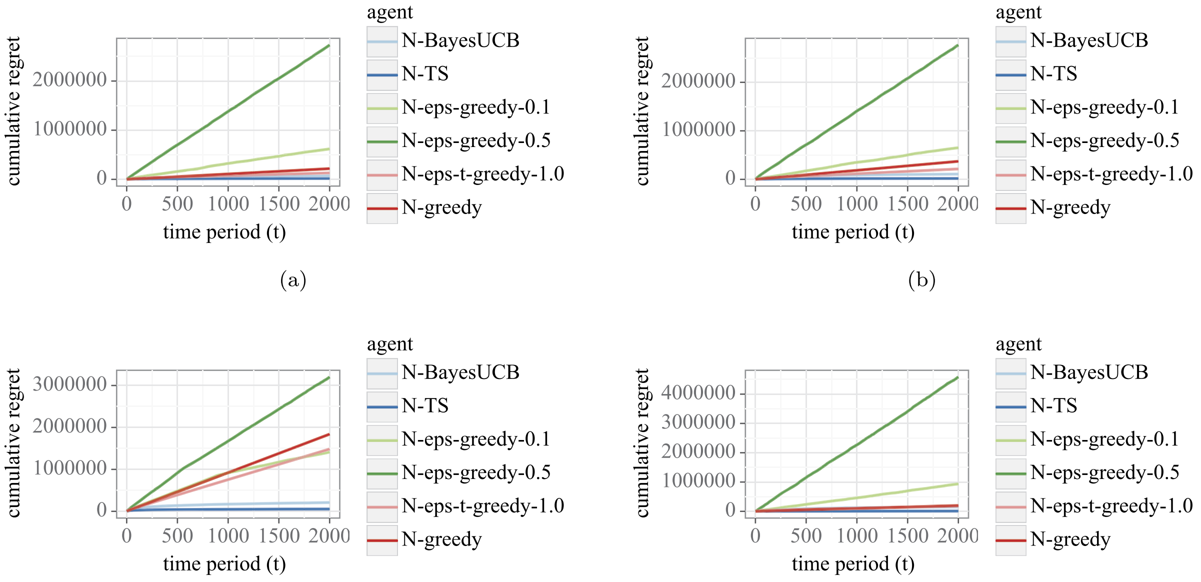
Since we assume that each agent is aware of the true prior distribution in this problem setting, we settle on the rectified Gaussian model of energy consumption for these experiments, with Gaussian prior distributions. As replacements for the Log-Gaussian agents, we increase the number of baselines by implementing a version of ϵ-greedy adapted to combinatorial semi-bandits, based on Algorithm 1 introduced in the supplementary material of [18].

As outlined in Algorithm 7, at each time step t with probability ϵ_t, we select an edge $(u_h, u_{h'}) \in \mathcal{E}$ uniformly at random. We then find the shortest paths with respect to the posterior mean vector, between (1) the source vertex of the problem instance and u_h, and (2) $u_{h'}$ and the target vertex. The resulting concatenated path, including the edge $(u_h, u_{h'})$, is used to explore the road network graph. With probability $1 - \epsilon_t$, we instead greedily select the shortest path between the source and target vertices, exploiting the current posterior mean estimates.

We evaluate agents using constant values of ϵ_t (0.1 and 0.5), as well as an agent ϵ_t decaying in t (with $\epsilon_t = \frac{1}{t}$). We motivate the latter with Theorem 4 in the supplementary material of [18], where the authors show a sub-linear upper bound on the expected regret of their ϵ_t-greedy algorithm, with ϵ_t in the order of $\frac{1}{t}$ (with an additional constant factor derived from information about the problem instance).

Algorithm 7 ϵ_t-greedy for combinatorial semi-bandits.

```
Require: Time horizon $T$, prior parameters $\boldsymbol{\mu}_0, \boldsymbol{\varsigma}_0$, exploration probability $\epsilon_t$ for $t \in [T]$.
for $t \leftarrow 1, \dots, T$ do
    Sample $x \sim \text{Bernoulli}(\epsilon_t)$
    if $x = 1$ then
        Sample an edge $(u_h, u_{h'})$ uniformly from $\mathcal{E}$.
        $\boldsymbol{p}_1 \leftarrow$ Shortest path w.r.t. $\boldsymbol{\mu}_{t-1}$, between source vertex and $u_h$.
        $\boldsymbol{p}_2 \leftarrow$ Shortest path w.r.t. $\boldsymbol{\mu}_{t-1}$, between $u_{h'}$ and target vertex.
        $\boldsymbol{a}_t \leftarrow$ Concatenate $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$.
    else
        $\boldsymbol{a}_t \leftarrow$ Shortest path w.r.t. $\boldsymbol{\mu}_{t-1}$, between source and target vertices.
    Play $\boldsymbol{a}_t$, update posterior parameters $\boldsymbol{\mu}_t, \boldsymbol{\varsigma}_t$ using observed rewards $r_t(\boldsymbol{a}_t)$.
```



(a)

(b)

(c)

(d)

Fig. 4. Experimental results on the real-world road networks in the scenario where there is correlation between edges in the environments. For Luxembourg #1, Luxembourg #2, Monaco and Turin, respectively, (a), (b), (c) and (d) show the average cumulative regret results for the Thompson Sampling (TS), BayesUCB, ϵ_t-greedy with fixed $\epsilon_t = 0.1$ (eps-greedy-0.1), $\epsilon_t = 0.5$ (eps-greedy-0.5), ϵ_t-greedy with decaying ϵ_t, and probabilistic greedy algorithms, with rectified Gaussian energy consumption models.

As shown in Figs. 3a, 3b, 3c and 3d, the results from the experiments with the TS, BayesUCB and (pure) greedy agents closely match the corresponding experiments in the misspecified prior problem setting of the previous section, while ϵ_t-greedy with decaying ϵ_t has comparable performance to the greedy agent. The ϵ_t-greedy agents with constant ϵ_t perform consistently worse than the other agents. Also supported by Table 2, the regret of the TS agent still saturates rapidly and achieves the best average regret out of the evaluated agents for all cities.

5.1.3. *Networks with correlated edge weights*

To demonstrate that the proposed framework performs well even when a few environment assumptions are relaxed, we run an additional set of experiments in a variation of the setting described in Section 5.1.2, with results shown in Fig. 4 and Table 3. Whereas in the previous sections the stochastic weights of all edges are assumed to be mutually independent, we now introduce correlation between edge weights. An example of this in real-world road networks can be that traffic congestion on one road segment is likely to affect nearby road segments as well.

As in the previous section, a mean vector $\boldsymbol{\theta}^*$ unknown to the agents is generated by the environment, where each element is sampled independently from the (Gaussian) prior distribution of each edge in the road network. Subsequently, we randomly assign all edges in $\mathcal{E}$ to a set of $|\mathcal{E}|/2$ pairs of edges. We let the energy consumption of the individual edges in each such pair of edges $(e, e') \in \mathcal{E} \times \mathcal{E}$ be perfectly correlated, but we define the marginal distributions according to the model in Section 2.2. In each time step, we jointly sample the energy consumption for each pair (e, e') from a two-dimensional distribution with mean vector $\boldsymbol{\theta}^*_{(e,e')}$ and covariance matrix $\Sigma_{(e,e')}$, defined as

$$\boldsymbol{\theta}^*_{(e,e')} = \begin{bmatrix} \theta^*_e \\ \theta^*_{e'} \end{bmatrix}, \quad \Sigma_{(e,e')} = \begin{bmatrix} \sigma_e^2 & \sigma_e \sigma_{e'} \\ \sigma_e \sigma_{e'} & \sigma_{e'}^2 \end{bmatrix}.$$

Table 3
Average and standard deviation of regret at $T = 2000$ of agents, where there is correlation between edges in the environments. Bold average values indicate the agent with the lowest regret in each scenario.

City	Luxembourg #1		Luxembourg #2	
Agent	AVG	SD	AVG	SD
N-BayesUCB	73439.29	7133.48	108704.02	9960.32
N-TS	**18313.43**	2235.25	**18351.64**	4221.91
N-eps-greedy-0.1	619592.02	100767.70	652295.17	115089.44
N-eps-greedy-0.5	2732324.05	160432.90	2768669.58	141189.83
N-eps-t-greedy-1.0	130141.09	85763.04	214824.01	173082.45
N-greedy	218541.56	208088.80	371926.16	326922.34
City	Monaco		Turin	
Agent	AVG	SD	AVG	SD
N-BayesUCB	209794.25	57753.94	148316.20	8128.75
N-TS	**54797.54**	22390.01	**8809.96**	4657.47
N-eps-greedy-0.1	1414164.08	1406368.66	936641.81	91197.23
N-eps-greedy-0.5	3202314.86	1601363.76	4584797.59	162979.92
N-eps-t-greedy-1.0	1485022.56	2844251.84	178088.28	142590.73
N-greedy	1840158.61	3102682.97	199636.06	186795.70

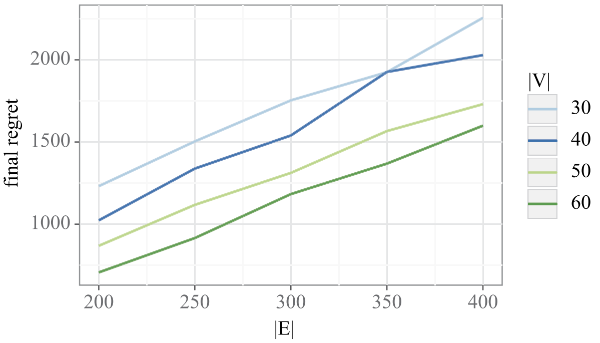


Fig. 5. Final cumulative regret ($T = 2000$) on synthetic networks as a function of $|\mathcal{E}|$.

Beyond the generation of correlated energy consumption by the environment, the experiments are set up exactly as in Section 5.1.2. The agents are assumed to be unaware of the correlation, and only attempt to estimate the parameters of the marginal distributions. As shown in Figs. 4a, 4b, 4c and 4d, as well as in Table 3, when compared with results in the previous section, the performance of the agents is not noticeably affected by the presence of correlation.

5.2. Synthetic networks

In order to evaluate the regret bound in Proposition 1, we design synthetic directed acyclic network instances $\mathcal{G}(\mathcal{V}, \mathcal{E}, \boldsymbol{w})$ according to a specified number of vertices n and number of edges o (with the constraint that $n - 1 \leq o \leq n(n-1)/2$). We start the procedure by adding n vertices $u_1, \ldots, u_n$ to $\mathcal{V}$. Then for each $h \in [1, n-1]$ we add an edge (u_h, u_{h+1}) to $\mathcal{E}$. This ensures that the network contains a path with all vertices in $\mathcal{V}$. Finally, we add $o - n$ edges $(u_h, u_{h'})$ uniformly at random to $\mathcal{E}$, such that $h \neq h'$, $h + 1 \neq h'$ and $h < h'$.

Since these networks are synthetic, instead of modeling probabilistic energy consumption, we design instances where it is difficult for an exploration algorithm to find the path with the lowest expected cost. Given a synthetic network $\mathcal{G}$ generated according to the aforementioned procedure, we select $\boldsymbol{p} = \langle u_1, \ldots, u_n \rangle$ to be the optimal path. In other words, $\boldsymbol{p}$ contains every vertex $u \in \mathcal{V}$. The reward distribution for each edge e in $\boldsymbol{p}$ is chosen to be $\mathcal{N}(-\tilde{E}_e | \theta_e^*, \sigma_e^2)$ with $\theta_e^* = -10$ and $\sigma_e^2 = 4$. For $(u_h, u_{h'}) \in E$ where $(u_h, u_{h'}) \notin \boldsymbol{p}$, we set $\theta_e^* = -11(h' - h)$, where $h' - h$ is the number of vertices skipped by the shortcut. This guarantees that no matter the size of the network and the number of edges that form shortcuts between vertices in $\boldsymbol{p}$, $\boldsymbol{p}$ will always have a lower expected cost than any other path in $\mathcal{G}$.

For the agent prior $\mathcal{N}(\theta_e^* | \mu_{e,0}, \varsigma_{e,0}^2)$, we set $\mu_{e,0} = -11(h' - h)$ and $\varsigma_{e,0}^2 = 8$. This choice of prior mean implies according to our prior beliefs, every path from the source u_1 to the target u_n will initially have the same estimated expected cost.

We run the synthetic network experiment with $T = 2000$ time steps, varying the number of vertices $|\mathcal{V}| \in \{30, 40, 50, 60\}$ and edges $|\mathcal{E}| \in \{200, 250, 300, 350, 400\}$. In Fig. 5, each plot represents the cumulative regret at $T = 2000$ for a fixed $|\mathcal{V}|$, as a function of $|\mathcal{E}|$. We observe that the regret increases no more than linearly with the number of edges, which is consistent with the theoretical regret bound in Corollary 11.

6. Conclusion

We developed a Bayesian online learning framework for the problem of energy efficient navigation of electric vehicles. Our Bayesian model assumes a rectified Gaussian or Log-Gaussian energy model. To learn the unknown parameters of the model, we adapted exploration methods such as Thompson Sampling and BayesUCB within the online learning framework. We extended the framework to multi-agent and batched feedback settings, and established theoretical regret bounds. Finally, we demonstrated the performance of the framework with several real-world and synthetic experiments.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

Niklas Åkerblom is a PhD student employed by Volvo Car Corporation. This work is funded by the Strategic Vehicle Research and Innovation Programme (FFI) of Sweden, through the project EENE (reference number: 2018-01937).

Appendix A. Notation

The notation used throughout the paper is summarized below, in Table A.1. Note that since each edge $e \in \mathcal{E}$ corresponds to a base arm $i \in \mathcal{A}$, these are used interchangeably as subscript indices to various variables.

Table A.1
Summary of the notation used throughout the paper.

Notation	Description
$\boldsymbol{a}$	A super-arm
$\boldsymbol{a}_t$	Super-arm selected at time t
$\boldsymbol{a}^*$	Optimal super-arm
a	An arm
a_t	Arm selected at time t
b	Batch index (in batched feedback setting)
$\boldsymbol{b}$	A super-arm (alternative)
$\boldsymbol{c}$	A cycle (path)
d	Number of base arms
e	An edge
g	Gravitational acceleration (m/s^2)
h	Vertex index
h'	Vertex index (alternative)
i	A base arm
j	A base arm (alternative)
k	Agent index (in multi-agent setting)
l_e	Length of edge e (m)
m	Vehicle mass (kg)
n	Final (vertex) index (of, e.g., a path)
o	Final number of edges in synthetic network setting
$\boldsymbol{p}$	A path (connected sequence of vertices / edges)
s	A time step / round (alternative)
t	A time step / round
t_b	Last time step / round of batch b
u	A vertex
v	Vehicle speed (m/s)
$\boldsymbol{w}_t$	Edge weight vector at time t
$w_{e,t}$	Weight of edge e at time t
z_e	Rectified (Gaussian) energy consumption edge e
A	Front surface area of vehicle (m^2)
B	Number of batches
C_r	Rolling resistance coefficient of edge e
C_d	Air drag coefficient of vehicle
E_e	Approximated energy consumption of edge e (Wh)
$\tilde{E}_e$	Stochastic energy consumption of edge e (Wh)
H_t	History (random) of actions and rewards until time t
H	A realized (fixed) history of actions and rewards
K	Number of agents in the multi-agent setting

Table A.1 (*continued*)

Notation	Description
M	A Markov Decision Process (MDP)
T	Time horizon
α_e	Inclination angle of edge e (radians)
β	Probability threshold parameter of quantile function
Δ_t	Instant regret (suboptimality gap) at time t
ϵ	Exploration probability of ϵ-greedy algorithm
ϵ_t	Exploration probability of ϵ_t-greedy algorithm at time t
η	Powertrain efficiency of vehicle
η^+	Powertrain efficiency of vehicle during traction
η^-	Powertrain efficiency of vehicle during braking
$\boldsymbol{\theta}$	A mean vector
$\hat{\boldsymbol{\theta}}_t$	Average reward vector until time t
$\tilde{\boldsymbol{\theta}}$	Sampled mean reward vector
$\boldsymbol{\theta}^*$	True mean reward vector
$\boldsymbol{\theta}^*_{(i,j)}$	True mean reward vector of correlated base arms i and j
$\hat{\theta}_{i,t}$	Average reward of base arm i until time t
$\tilde{\theta}_i$	Sampled mean reward of base arm i
θ_i^*	True mean reward of base arm i
ϑ	Factor of the mean to calculate prior standard deviation
κ_t	Feedback delay of arm selected at time t
λ	A distribution
$\boldsymbol{\mu}_0$	Prior mean vector
$\boldsymbol{\mu}_t$	Posterior mean vector at time t
$\mu_{i,0}$	Prior mean of base arm i
$\mu_{i,t}$	Posterior mean of base arm i at time t
$\bar{\nu}_{i,x}$	Average reward of base arm i over first x plays
ρ	Air density (kg/m^3)
σ_i^2	Noise variance of base arm i
$\Sigma_{(i,j)}$	Covariance matrix of correlated base arms i and j
$\boldsymbol{\varsigma}_0$	Prior standard deviation vector
$\boldsymbol{\varsigma}_t$	Posterior standard deviation vector at time t
$\varsigma_{i,0}^2$	Prior variance of base arm i
$\varsigma_{i,t}^2$	Posterior variance of base arm i at time t
τ	Episode length of reinforcement learning problem
φ	Factor of the mean to calculate noise standard deviation
ψ_i	Gaussian prior mean of base arm i
$\mathcal{A}$	Set of base arms
$\mathcal{D}_t$	Set of delayed rewards at time t
$\mathcal{E}$	Set of edges
$\mathcal{G}(\mathcal{V}, \mathcal{E}, \boldsymbol{w})$	A (weighted) graph with vertices $\mathcal{V}$, edges $\mathcal{E}$, and weights $\boldsymbol{w}$
$\mathcal{I}$	Set of super-arms
$\mathcal{M}$	Set of Markov Decision Processes (MDPs)
$\mathcal{P}$	Set of paths
$\mathcal{V}$	Set of vertices
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^+$	Set of non-negative real numbers
$\text{BayesRegret}(T)$	Bayesian regret until horizon T
$\text{BayesRegret}_k(T)$	Bayesian regret of agent k until horizon T
$\text{Bernoulli}(\cdot)$	Bernoulli distribution
$\mathbb{E}[\cdot]$	Expected value of random variable
$f_{\boldsymbol{\theta}}(\boldsymbol{a})$	Expected reward of super-arm $\boldsymbol{a}$, given the mean vector $\boldsymbol{\theta}$
$f^R_{\boldsymbol{\theta}}(\boldsymbol{a})$	Expected reward of super-arm $\boldsymbol{a}$, given the mean vector $\boldsymbol{\theta}$, under rectified Gaussian base arm feedback
$L(\boldsymbol{a}, H)$	Lower confidence bound of super-arm $\boldsymbol{a}$ given history H
$\mathcal{LN}(\cdot)$	Log-Gaussian distribution
$\mathbf{Mode}[\cdot]$	Mode of random variable
$\mathcal{N}(\cdot)$	Gaussian distribution
$\mathcal{N}^R(\cdot)$	Rectified Gaussian distribution
$N_t(i)$	Number of plays of base arm i until time t
$\mathcal{O}(\cdot)$	Order of a function
$\tilde{\mathcal{O}}(\cdot)$	Order of a function (excluding polylogarithmic factors)
$P(\cdot)$	Probability distribution of random variable
$\Pr\{\cdot\}$	Probability of event
$Q(\beta, \lambda)$	Quantile function of distribution λ with probability threshold β
$\text{Queue}[\boldsymbol{a}]$	Delayed feedback queue of super-arm $\boldsymbol{a}$
$r_t(\boldsymbol{a})$	Reward of super-arm $\boldsymbol{a}$ at time t
$\text{Regret}(T)$	Frequentist regret until horizon T
$\text{Regret}_k(T)$	Frequentist regret of agent k until horizon T
$U(\boldsymbol{a}, H)$	Upper confidence bound of super-arm $\boldsymbol{a}$ given history H
$\mathbf{Var}[\cdot]$	Variance of random variable
$\phi(x)$	Standard Gaussian probability density function (PDF)
$\Phi(x)$	Standard Gaussian cumulative distribution function (CDF)

References

[1] N. Åkerblom, Y. Chen, M. Haghir Chehreghani, An online learning framework for energy-efficient navigation of electric vehicles, in: C. Bessiere (Ed.), Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, International Joint Conferences on Artificial Intelligence Organization, 2020, pp. 2051–2057, main track.

[2] A. Artmeier, J. Haselmayr, M. Leucker, M. Sachenbacher, The shortest path problem revisited: optimal routing for electric vehicles, in: R. Dillmann, J. Beyerer, U.D. Hanebeck, T. Schultz (Eds.), KI 2010: Advances in Artificial Intelligence, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 309–316.

[3] M. Sachenbacher, M. Leucker, A. Artmeier, J. Haselmayr, Efficient energy-optimal routing for electric vehicles, in: Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI'11, AAAI Press, 2011, pp. 1402–1407.

[4] P.E. Hart, N.J. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, IEEE Trans. Syst. Sci. Cybern. 4 (2) (1968) 100–107, https://doi.org/10.1109/TSSC.1968.300136.

[5] M. Baum, J. Sauer, D. Wagner, T. Zündorf, Consumption profiles in route planning for electric vehicles: theory and applications, in: C.S. Iliopoulos, S.P. Pissis, S.J. Puglisi, R. Raman (Eds.), 16th International Symposium on Experimental Algorithms (SEA 2017), in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 75, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2017, pp. 19:1–19:18.

[6] R. Basso, B. Kulcsár, B. Egardt, P. Lindroth, I. Sanchez-Diaz, Energy consumption estimation integrated into the electric vehicle routing problem, Transp. Res., Part D, Transp. Environ. 69 (2019) 141–167, https://doi.org/10.1016/j.trd.2019.01.006.

[7] R. Basso, B. Kulcsár, I. Sanchez-Diaz, Electric vehicle routing problem with machine learning for energy prediction, Transp. Res., Part B, Methodol. 145 (2021) 24–55, https://doi.org/10.1016/j.trb.2020.12.007.

[8] W. Thompson, On the likelihood that one unknown probability exceeds another in view of the evidence of two samples, Biometrika 25 (3–4) (1933) 285–294, https://doi.org/10.2307/2332286.

[9] T. Graepel, J.Q.n. Candela, T. Borchert, R. Herbrich, Web-scale Bayesian click-through rate prediction for sponsored search advertising in Microsoft's bing search engine, in: Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10, Omnipress, Madison, WI, USA, 2010, pp. 13–20.

[10] O. Chapelle, L. Li, An empirical evaluation of Thompson sampling, in: Proceedings of the 24th International Conference on Neural Information Processing Systems, NIPS'11, Curran Associates Inc., Red Hook, NY, USA, 2011, pp. 2249–2257.

[11] Y. Chen, J.-M. Renders, M. Haghir Chehreghani, A. Krause, Efficient online learning for optimizing value of information: theory and application to interactive troubleshooting, in: G. Elidan, K. Kersting (Eds.), Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence (UAI 2017), vol. 2, Curran Associates, Inc., Red Hook, NY, 2017, pp. 966–983.

[12] S. Agrawal, N. Goyal, Analysis of Thompson sampling for the multi-armed bandit problem, in: S. Mannor, N. Srebro, R.C. Williamson (Eds.), Proceedings of the 25th Annual Conference on Learning Theory, in: Proceedings of Machine Learning Research, vol. 23, PMLR, Edinburgh, Scotland, 2012, pp. 39.1–39.26.

[13] E. Kaufmann, N. Korda, R. Munos, Thompson sampling: an asymptotically optimal finite-time analysis, in: N.H. Bshouty, G. Stoltz, N. Vayatis, T. Zeugmann (Eds.), Algorithmic Learning Theory, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 199–213.

[14] S. Bubeck, C.-Y. Liu, Prior-free and prior-dependent regret bounds for Thompson sampling, in: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1, NIPS'13, Curran Associates Inc., Red Hook, NY, USA, 2013, pp. 638–646.

[15] I. Osband, B. Van Roy, Why is posterior sampling better than optimism for reinforcement learning?, in: D. Precup, Y.W. Teh (Eds.), Proceedings of the 34th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, PMLR, vol. 70, 2017, pp. 2701–2710.

[16] S. Wang, W. Chen, Thompson sampling for combinatorial semi-bandits, in: J. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, PMLR, vol. 80, 2018, pp. 5114–5122.

[17] P. Auer, Using confidence bounds for exploitation-exploration trade-offs, J. Mach. Learn. Res. 3 (2002) 397–422.

[18] W. Chen, Y. Wang, Y. Yuan, Combinatorial multi-armed bandit: general framework and applications, in: S. Dasgupta, D. McAllester (Eds.), Proceedings of the 30th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 28, PMLR, Atlanta, Georgia, USA, 2013, pp. 151–159.

[19] E. Kaufmann, O. Cappe, A. Garivier, On Bayesian upper confidence bounds for bandit problems, in: N.D. Lawrence, M. Girolami (Eds.), Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, in: Proceedings of Machine Learning Research, vol. 22, PMLR, La Palma, Canary Islands, 2012, pp. 592–600.

[20] E. Kaufmann, On Bayesian index policies for sequential resource allocation, Ann. Stat. 46 (2) (2018) 842–865.

[21] P.B. Reverdy, V. Srivastava, N.E. Leonard, Modeling human decision making in generalized Gaussian multiarmed bandits, Proc. IEEE 102 (4) (2014) 544–571, https://doi.org/10.1109/JPROC.2014.2307024.

[22] L. Codecá, R. Frank, S. Faye, T. Engel, Luxembourg SUMO traffic (LuST) scenario: traffic demand evaluation, IEEE Intell. Transp. Syst. Mag. 9 (2) (2017) 52–63.

[23] L. Codecá, J. Härri, Towards multimodal mobility simulation of C-ITS: the Monaco SUMO traffic scenario, in: VNC 2017, IEEE Vehicular Networking Conference, November 27-29, 2017, Torino, Italy, Torino, Italy, 2017, pp. 97–100.

[24] M. Rapelli, C. Casetti, G. Gagliardi, Vehicular traffic simulation in the city of Turin from raw data, IEEE Trans. Mob. Comput. (2021), https://doi.org/10.1109/TMC.2021.3075985.

[25] Y. Gai, B. Krishnamachari, R. Jain, Combinatorial network optimization with unknown variables: multi-armed bandits with linear rewards and individual observations, IEEE/ACM Trans. Netw. 20 (5) (2012) 1466–1478, https://doi.org/10.1109/TNET.2011.2181864.

[26] K. Liu, Q. Zhao, Adaptive shortest-path routing under unknown and stochastically varying link states, in: 2012 10th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2012, pp. 232–237.

[27] Z. Zou, A. Proutiere, M. Johansson, Online shortest path routing: the value of information, in: 2014 American Control Conference, 2014, pp. 2142–2147.

[28] V. Dani, T.P. Hayes, S.M. Kakade, Stochastic linear optimization under bandit feedback, in: 21st Annual Conference on Learning Theory, 2008, pp. 355–366.

[29] Y. Abbasi-Yadkori, D. Pál, C. Szepesvári, Improved algorithms for linear stochastic bandits, in: Proceedings of the 24th International Conference on Neural Information Processing Systems, NIPS'11, Curran Associates Inc., Red Hook, NY, USA, 2011, pp. 2312–2320.

[30] A. Gopalan, S. Mannor, Y. Mansour, Thompson sampling for complex online problems, in: E.P. Xing, T. Jebara (Eds.), Proceedings of the 31st International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 32, PMLR, Bejing, China, 2014, pp. 100–108.

[31] Z. Wen, B. Kveton, A. Ashkan, Efficient learning in large-scale combinatorial semi-bandits, in: F. Bach, D. Blei (Eds.), Proceedings of the 32nd International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 37, PMLR, Lille, France, 2015, pp. 1113–1122.

[32] D. Russo, B. Van Roy, Learning to optimize via posterior sampling, Math. Oper. Res. 39 (4) (2014) 1221–1243.

[33] P. Joulani, A. Gyorgy, C. Szepesvari, Online learning under delayed feedback, in: S. Dasgupta, D. McAllester (Eds.), Proceedings of the 30th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 28, PMLR, Atlanta, Georgia, USA, 2013, pp. 1453–1461.

[34] V. Perchet, P. Rigollet, S. Chassang, E. Snowberg, Batched bandit problems, Ann. Stat. 44 (2) (2016) 660–681, https://doi.org/10.1214/15-AOS1381.

[35] Y. Han, Z. Zhou, Z. Zhou, J. Blanchet, P.W. Glynn, Y. Ye, Sequential batch learning in finite-action linear contextual bandits, arXiv preprint arXiv:2004.06321, 2020, https://doi.org/10.48550/ARXIV.2004.06321.

[36] Z. Ren, Z. Zhou, J.R. Kalagnanam, Batched learning in generalized linear contextual bandits with general decision sets, IEEE Control Syst. Lett. 6 (2022) 37–42, https://doi.org/10.1109/LCSYS.2020.3047601.
[37] Z. Ren, Z. Zhou, Dynamic batch learning in high-dimensional sparse linear contextual bandits, arXiv preprint arXiv:2008.11918, 2020, https://doi.org/10.48550/ARXIV.2008.11918.
[38] N. Åkerblom, F.S. Hoseini, M. Haghir Chehreghani, Online learning of network bottlenecks via minimax paths, Mach. Learn. 112 (2023) 131–150, https://doi.org/10.1007/s10994-022-06270-0.
[39] L. Guzzella, A. Sciarretta, et al., Vehicle Propulsion Systems, Springer, 2007.
[40] E.W. Dijkstra, A note on two problems in connexion with graphs, Numer. Math. 1 (1959) 269–271, https://doi.org/10.1007/BF01386390.
[41] A. Shimbel, Structure in communication nets, in: Proceedings of the Symposium on Information Networks, Polytechnic Institute of Brooklyn, 1954, pp. 199–203.
[42] L.R. Ford Jr, Network flow theory, Tech. Rep. P-932, Rand Corporation, Santa Monica, CA, 1956.
[43] R. Bellman, On a routing problem, Q. Appl. Math. 16 (1958) 87–90, https://doi.org/10.1090/qam/102435.
[44] D.B. Johnson, Efficient algorithms for shortest paths in sparse networks, J. ACM 24 (1) (1977) 1–13, https://doi.org/10.1145/321992.321993.
[45] X. Wu, D. Freese, A. Cabrera, W.A. Kitch, Electric vehicles' energy consumption measurement and estimation, Transp. Res., Part D, Transp. Environ. 34 (2015) 52–67, https://doi.org/10.1016/j.trd.2014.10.007.
[46] H.S.B. Herath, P. Kumar, Using copula functions in Bayesian analysis: a comparison of the lognormal conjugate, Eng. Econ. 60 (2) (2015) 89–108, https://doi.org/10.1080/0013791X.2014.962719.
[47] N. Cesa-Bianchi, G. Lugosi, Combinatorial bandits, in: jCSS Special Issue: Cloud Computing 2011, J. Comput. Syst. Sci. 78 (5) (2012) 1404–1422, https://doi.org/10.1016/j.jcss.2012.01.001.
[48] I. Osband, B. Van Roy, D. Russo, (More) efficient reinforcement learning via posterior sampling, in: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13, Curran Associates Inc., Red Hook, NY, USA, 2013, pp. 3003–3011.
[49] G.H. Polychronopoulos, J.N. Tsitsiklis, Stochastic shortest path problems with recourse, Networks 27 (2) (1996) 133–143, https://doi.org/10.1002/(SICI)1097-0037(199603)27:2<133::AID-NET5>3.0.CO;2-L.
[50] T. Mandel, Y.-E. Liu, E. Brunskill, Z. Popović, The queue method: handling delay, heuristics, prior data, and evaluation in bandits, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI'15, AAAI Press, 2015, pp. 2849–2856.
[51] Z. Huang, B. Hu, J. Pan, Caching by user preference with delayed feedback for heterogeneous cellular networks, IEEE Trans. Wirel. Commun. 20 (3) (2021) 1655–1667, https://doi.org/10.1109/TWC.2020.3035377.
[52] Z. Gao, Y. Han, Z. Ren, Z. Zhou, Batched multi-armed bandits problem, in: Proceedings of the 33rd International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA, 2019, pp. 503–513.
[53] W.C. Horrace, Moments of the truncated normal distribution, J. Product. Anal. 43 (2) (2015) 133–138, https://doi.org/10.1007/s11123-013-0381-8.
[54] J. Chan, A. Pacchiano, N. Tripuraneni, Y.S. Song, P. Bartlett, M.I. Jordan, Parallelizing contextual linear bandits, arXiv preprint arXiv:2105.10590, 2021, https://doi.org/10.48550/ARXIV.2105.10590.
[55] B. Kveton, Z. Wen, A. Ashkan, C. Szepesvari, Tight regret bounds for stochastic combinatorial semi-bandits, in: G. Lebanon, S.V.N. Vishwanathan (Eds.), Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, in: Proceedings of Machine Learning Research, vol. 38, PMLR, San Diego, California, USA, 2015, pp. 535–543.
[56] D.J. Russo, B. Van Roy, A. Kazerouni, I. Osband, Z. Wen, A tutorial on Thompson sampling, Found. Trends Mach. Learn. 11 (1) (2018) 1–96, https://doi.org/10.1561/2200000070.
[57] T.G. Farr, M. Kobrick, Shuttle radar topography mission produces a wealth of data, Eos Trans. AGU 81 (48) (2000) 583–585, https://doi.org/10.1029/EO081i048p00583.