Online Learning of Network Bottlenecks via Minimax Paths

Citation for the original published paper (version of record):
Åkerblom, N., Hoseini, F., Haghir Chehreghani, M. (2023). Online Learning of Network Bottlenecks via Minimax Paths. Machine Learning, 112(1): 131-150.
http://dx.doi.org/10.1007/s10994-022-06270-0

Machine Learning (2023) 112:131–150
https://doi.org/10.1007/s10994-022-06270-0

Online learning of network bottlenecks via minimax paths

Niklas Åkerblom[1,2] · Fazeleh Sadat Hoseini[1] · Morteza Haghir Chehreghani[1]

Received: 10 December 2021 / Revised: 6 July 2022 / Accepted: 7 October 2022 /
Published online: 8 November 2022

Abstract
In this paper, we study bottleneck identification in networks via extracting minimax paths. Many real-world networks have stochastic weights for which full knowledge is not available in advance. Therefore, we model this task as a combinatorial semi-bandit problem to which we apply a combinatorial version of Thompson Sampling and establish an upper bound on the corresponding Bayesian regret. Due to the computational intractability of the problem, we then devise an alternative problem formulation which approximates the original objective. Finally, we experimentally evaluate the performance of Thompson Sampling with the approximate formulation on real-world directed and undirected networks.

Keywords Online Learning · Combinatorial Semi-bandit · Thompson Sampling · Bottleneck Identification

1 Introduction

Bottleneck identification constitutes an important task in network analysis, with applications including transportation planning and management (Berman & Handler, 1987), routing in computer networks (Shacham, 1992) and various bicriterion path problems (Hansen, 1980). The path-specific bottleneck, on a path between a source and a target node in a network, is defined as the edge with a maximal cost or weight according to some criterion such as transfer time, load, commute time, distance, etc. The goal of bottleneck identification and avoidance is then to find a path whose bottleneck is minimal. Thus, one may

Editors: Krzysztof Dembczynski and Emilie Devijver.

Niklas Åkerblom, Fazeleh Sadat Hoseini have contributed equally to this work.

✉ Niklas Åkerblom
niklas.akerblom@chalmers.se

✉ Fazeleh Sadat Hoseini
fazeleh@chalmers.se

Morteza Haghir Chehreghani
morteza.chehreghani@chalmers.se

[1] Chalmers University of Technology, Gothenburg, Sweden

[2] Volvo Car Corporation, Gothenburg, Sweden

model bottleneck identification as the problem of computing the minimax edge over the given network/graph, to obtain an edge with a minimal largest gap between the source and target nodes. Equivalently, it can be formulated as a widest path problem or maximum capacity path problem (Pollack, 1960) where the edge weights have been negated.

In transportation, identifying bottlenecks is important for city governments and traffic managers to monitor and improve overall performance. For example, when driving between two locations, identifying a bottleneck means finding the section of road that slows down the traffic between two locations. Consider a social network where there is an information flow between two nodes (source and destination). Then we can define the bottleneck as a link in all the paths between these two nodes that has the weakest connection and causes the information flow to fail. So, bottlenecks here are related to the possibility of information loss.

The aforementioned formulations assume that the network or the graph is fully specified, i.e., that all the edge weights are fully known. However, in practice, the edge weights might not be known in advance or they might include some inherent uncertainty. In this paper, we tackle such situations by developing an online learning framework to learn the edge weight distributions of the underlying network, while solving the bottleneck identification problem for different problem instances.

For example, in the transportation scenario, city governments often have access to fleets of vehicles utilized for various municipal services. These may be used to sequentially and continuously to gain knowledge about traffic flow from the environment, while it is still desirable to avoid causing unnecessary inconvenience and stress (Hennessy & Wiesenthal, 1999) to the employees operating the vehicles by excessively exploring congested paths. If care is taken to spread the costs over time, exploration may be performed continuously without having a specific end time known in advance (i.e., the *time horizon* of the sequential decision making problem).

For this purpose, we view this as a multi-armed bandit (MAB) problem and focus on Thompson Sampling (TS) (Thompson, 1933), a method that suits probabilistic online learning well. Thompson Sampling is an early Bayesian method for addressing the tradeoff between exploration and exploitation in sequential decision making problems. It balances these by randomly sampling available actions according to their posterior probability of being optimal, given prior beliefs and observations from previously selected actions. An action is more likely to be sampled if the posterior distribution over the expected reward of that action has high uncertainty (exploration) or high mean (exploitation).

The method has only recently been thoroughly evaluated through experimental studies (Chapelle & Li, 2011; Graepel et al., 2010) and theoretical analyses (Kaufmann et al., 2012; Agrawal et al., 2012; Russo & Van Roy, 2014), where it has been shown to be asymptotically optimal in the sense that it matches well-known lower bounds of these types of problems (Lai & Robbins, 1985). Furthermore, the algorithm does not assume knowledge of the time horizon, i.e., it is an *anytime* algorithm.

Among many other problem settings, Thompson Sampling has been adapted to online versions of combinatorial optimization problems with retained theoretical guarantees (Wang & Chen, 2018), where one application is to find shortest paths in graphs (Liu & Zhao, 2012; Gai et al., 2012; Zou et al., 2014; Åkerblom et al., 2020).

Another commonly used method for these problems is Upper Confidence Bound (UCB) (Auer, 2003), which utilizes optimism to balance exploration and exploitation. UCB has been adapted to combinatorial settings (Chen et al., 2013), and also exists in Bayesian variants (Kaufmann et al., 2012). Recently, a variant of UCB has been studied for bottleneck avoidance problems in a combinatorial pure exploration setting (Du et al., 2021). They consider a different problem setting and method than those we present in this paper, though their bottleneck reward function is similar to the one we use in our approximation method. The main difference between their setting and the standard combinatorial semi-bandit setting in how agents interact with the environment, is that instead of being restricted to selecting sets of actions respecting combinatorial constraints, they allow agents to sequentially try individual arms to identify the best feasible solution to the combinatorial problem. This is not applicable to our setting, since we may not observe the feedback of individual edges without also traversing a path containing those edges, potentially incurring cost from some other edge on that path.

Moreover, the objective in a pure exploration problem is to find the best action as quickly as possible, with either a fixed time budget or confidence level, using agents dedicated for this task. While identifying the best path is desirable in our problem setting as well, we are specifically interested in the case where existing agents are utilized and where using them exclusively for exploration is too costly. For that reason, we focus on *anytime* methods capable of distributing exploratory actions over time.

In this paper, we model the online bottleneck identification task as a stochastic combinatorial semi-bandit problem, for which we develop a combinatorial variant of Thompson Sampling. We then derive an upper bound on the corresponding Bayesian regret that is tight up to a polylogarithmic factor, which is consistent with the existing lower bounds for combinatorial semi-bandit problems. We face the issue of computational intractability with the *exact* problem formulation. We thus propose an *approximation* scheme, along with a theoretical analysis of its properties. Finally, we experimentally investigate the performance of the proposed method on directed and undirected real-world networks from transport and collaboration domains.

2 Bottleneck identification model

In this section, we first introduce the bottleneck identification problem over a fixed network and then describe a probabilistic model to be used in stochastic and uncertain situations.

2.1 Bottleneck identification over a network

We model a network by a graph $G(V, E, w)$, where V denotes the set of vertices (nodes) and each $e = (u, v) \in E$ indicates an edge between vertices u and v where $u, v \in V$ and $u \neq v$. Moreover, $w : E \rightarrow \mathbb{R}$ is a weight function defined for each edge of the graph, where for convenience, we use w_e to denote the weight of edge e. If G is directed, the pair (u, v) is ordered, otherwise, it is not (i.e., $(u, v) \equiv (v, u)$ for undirected graphs). A path p from vertex u (source) to vertex v (target) over G is a sequence of vertices $< v_1, v_2, \dots, v_{k-1}, v_k >$ where $v_1 = u$, $v_k = v$ and $(v_i, v_{i+1}) \in E, \forall i \in [1, k-1]$. It can also be seen as a sequence of edges $< (v_1, v_2), (v_2, v_3), \dots, (v_{k-1}, v_k) >$.

As previously mentioned, a bottleneck on a path p can be described as an edge with a maximal weight on that path. To find the smallest feasible bottleneck edge between the source node u and the target node v, we consider all the paths between them. For each path, we pick an edge with a maximal weight, to obtain all path-specific bottleneck edges. We then identify the smallest path-specific bottleneck edge in order to find the best feasible bottleneck edge, i.e., such that bottleneck edges with higher weights are avoided.

Therefore, given graph G, the bottleneck edge between $u \in V$ and $v \in V$ can be identified via extracting the minimax edge between them. With $P_{u,v}$ denoting the set of all possible paths from u to v over G, the bottleneck weight (incurred by the bottleneck edge) can be computed by

$$b(u, v; G) = \min_{p \in P_{u,v}} \max_{e \in p} w_e. \tag{1}$$

The quantity in Eq. 1 satisfies the (ultra) metric properties under some basic assumptions on the edge weights such as symmetry and nonnegativity. Hence, it is sometimes used as a proper distance measure to extract manifolds and elongated clusters in a non-parametric way (Haghir Chehreghani, 2020; Kim & Choi, 2007).

However, in our setting, such conditions do not need to be fulfilled by the edge weights. In general, we tolerate positive as well as negative edge weights, and we assume the graph might be directed, i.e., the edge weights are not necessarily symmetric. Therefore, despite the absence of (ultra) metric properties, the concept of minimax edges is still relevant for bottleneck identification.

To compute the minimax edge, one does not need to investigate all possible paths between the source and target nodes, which might be computationally infeasible. As studied by Hu (1961), minimax edges and paths over an arbitrary undirected graph are equal to the minimax edges over any minimum spanning tree (MST) computed over that graph. This equivalence simplifies the calculation of minimax edges, as there is only one path between every two vertices over an MST, whose maximal edge weight yields the minimax edge, i.e., the desired bottleneck.

For directed graphs, an MST might not represent the minimax edges in a straightforward manner. Hence, we instead rely on a modification (Berman & Handler, 1987) of Dijkstra's algorithm (Dijkstra, 1959) to extract minimax paths rather than the shortest paths.

2.2 Probabilistic model for bottleneck identification

In this paper, we study bottleneck identification in uncertain and stochastic settings. Therefore, instead of considering the weights w_e for $e \in E$ to be fixed, we view them as stochastic with fixed, albeit unknown, distribution parameters. Additionally, we assume that the weight of each edge follows a Gaussian distribution with known and finite variance. The Gaussian edge weight assumption is common for many important problem settings, like minimization of travel time (Seshadri & Srinivasan, 2010) or energy consumption (Åkerblom et al., 2020) in road networks. Furthermore, we assume that all edge weights are mutually independent. Hence,

$$w_e \sim \mathcal{N}(\theta_e^*, \sigma_e^2),$$

where θ_e^* denotes the unknown mean of edge e, and σ_e^2 is the known variance. To reduce cumbersome notation in the proofs, since the variance is assumed to be finite, we let $\sigma_e^2 \leq 1$ (by scaling the edge weight distributions). However, we emphasize that we *do not* assume that w_e and θ_e^* are bounded or non-negative.

It is convenient to be able to make use of prior knowledge in online learning problems where the action space is large, which motivates a Bayesian approach where we assume that the unknown mean θ_e^* is sampled from a known prior distribution:

$$\theta_e^* \sim \mathcal{N}(\mu_{e,0}, \varsigma_{e,0}^2).$$

We use a Gaussian prior for θ_e^* since it is conjugate to the Gaussian likelihood and allows for efficient recursive updates of posterior parameters upon a new weight observation $w_{e,t}$ at time t:

$$\varsigma_{e,t+1}^2 \leftarrow \left(\frac{1}{\varsigma_{e,t}^2} + \frac{1}{\sigma_e^2} \right)^{-1}, \tag{2}$$

$$\mu_{e,t+1} \leftarrow \varsigma_{e,t+1}^2 \left(\frac{\mu_{e,t}}{\varsigma_{e,t}^2} + \frac{w_{e,t}}{\sigma_e^2} \right). \tag{3}$$

Since our long-term objective is to find a path which minimizes the *expected* maximum edge weight along that path, we need a *framework* to sequentially select paths to update these parameters and learn enough information about the edge weight distributions.

The assumptions in this section might seem restrictive, and indeed, when the edge weights represent e.g., traffic congestion in a road network, it is reasonable to believe that edges are not independent, especially for neighboring road segments. There are ways of extending this setting to capture such dependencies, while retaining similar regret guarantees for the studied methods. Such extensions include the contextual setting, where expected edge weights are assumed to follow parameterized functions of contextual features (e.g., time-of-day, local ambient temperature, precipitation) revealed to the agent in each time step, before each action is taken. We leave such extensions to future work, though we note that the proofs in this work may be extended in a straightforward manner, analogous to the analysis of linear contextual Thompson Sampling by Russo and Van Roy (2014). Similarly, Thompson Sampling may be extended to the case where both the mean and variance are unknown, by assignment of a joint prior distribution over the parameters (Riquelme et al., 2018).

3 Online bottleneck learning framework

Consider a stochastic combinatorial semi-bandit problem (Cesa-Bianchi & Lugosi, 2012) with time horizon T, formulated as a problem of cost minimization rather than reward maximization. There is a set of base arms $\mathcal{A}$ (where we let $d := |\mathcal{A}|$) from which we may, at each time step $t \in [T]$, select a subset (or *super arm*) $\boldsymbol{a}_t \subseteq \mathcal{A}$. The selection is further restricted such that $\boldsymbol{a}_t \in \mathcal{I} \subseteq 2^{\mathcal{A}}$, where $\mathcal{I}$ is called the set of *feasible* super arms.

Upon selection of $\boldsymbol{a}_t$, the environment reveals a feedback $X_{i,t}$ drawn from some fixed and unknown distribution for each base arm $i \in \boldsymbol{a}_t$ (i.e., *semi-bandit* feedback). Furthermore, we then receive a super arm cost from the environment, $c(\boldsymbol{a}_t) := \max_{i \in \boldsymbol{a}_t} X_{i,t}$, i.e., the maximum of all base arm feedback for the selected super arm and the current time step. The objective is to select super arms $\boldsymbol{a}_t$ to minimize $\mathbb{E}\left[\sum_{t=1}^{T} c(\boldsymbol{a}_t)\right]$. This objective is typically reformulated as an equivalent *regret* minimization problem, where the (expected) regret is defined as

$$\text{Regret}(T) := \left(\sum_{t\in[T]} \mathbb{E}\left[c(\boldsymbol{a}_t)\right] \right) - T \cdot \min_{\boldsymbol{a}\in\mathcal{I}} \mathbb{E}[c(\boldsymbol{a})]. \tag{4}$$

To connect this to the probabilistic bottleneck identification model introduced in the previous section, we let each edge $e \in E$ in the graph G correspond to exactly one base arm $i \in \mathcal{A}$. For the online minimax path problem, the feasible set of super arms is then the set of all admissible paths in the graph, where the paths are directed or undirected depending on the type of graph. The feedback of each base arm i is simply the Gaussian weight of the matching edge e, with known variance σ_i^2 and unknown mean θ_i^*.

We denote the expected cost of a super arm $f_{\boldsymbol{\theta}}(\boldsymbol{a})$, where $\boldsymbol{\theta}$ is a mean vector and $f_{\boldsymbol{\theta}}(\boldsymbol{a}) := \mathbb{E}\left[\max_{i\in\boldsymbol{a}} C_i\right]$ such that $C_i \sim \mathcal{N}(\theta_i, \sigma_i^2)$. For Bayesian bandit settings and algorithms, it is common to consider the notion of *Bayesian regret*, with an additional expectation over problem instances drawn from the prior distribution (where we denote the prior distribution λ, over mean vectors $\boldsymbol{\theta}^*$):

$$\text{BayesRegret}(T) := \mathbb{E}_{\boldsymbol{\theta}^*\sim\lambda}\left[\mathbb{E}\left[\left(\sum_{t\in[T]} f_{\boldsymbol{\theta}^*}(\boldsymbol{a}_t) \right) - T \cdot \min_{\boldsymbol{a}\in\mathcal{I}} f_{\boldsymbol{\theta}^*}(\boldsymbol{a}) \,\middle|\, \boldsymbol{\theta}^* \right]\right]. \tag{5}$$

3.1 Thompson sampling with exact objective

It is not sufficient to find the super arm $\boldsymbol{a}$ which minimizes $f_{\boldsymbol{\mu}_t}(\boldsymbol{a})$ in each time step t, since a strategy which is *greedy* with respect to possibly imperfect current cost estimates may converge to a sub-optimal super arm. Thompson Sampling is one of several methods developed to address the trade-off between exploration and exploitation in stochastic online learning problems. It has been shown to exhibit good performance in many formulations, e.g., linear contextual bandits and combinatorial semi-bandits.

The steps performed in each time step t by Thompson Sampling, adapted to our setting, are described in Algorithm 1 . First, a mean vector $\tilde{\boldsymbol{\theta}}$ is sampled from the current posterior distribution (or from the prior in the first time step). Then, an arm $\boldsymbol{a}_t$ is selected which minimizes the expected cost $f_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{a}_t)$ with respect to the sampled mean vector. These first two steps are equivalent to selecting the arm according to the posterior probability of it being optimal.

In combinatorial semi-bandit problems, the method of finding the best super arm according to the sampled parameters is often called an *oracle*.

When the super arm $\boldsymbol{a}_t$ is played, the environment reveals the feedback $X_{i,t}$ if and only if $i \in \boldsymbol{a}_t$, which is a property called *semi-bandit feedback*. Finally, these observations are used to update the posterior distribution parameters.

Algorithm 1 TS for minimax paths (exact)

```
Input: Prior parameters $\boldsymbol{\mu}_0, \varsigma_0$
For each base arm, play a super arm which contains it.
for $t \leftarrow 1, \dots, T$ do
    for $i \in \mathcal{A}$ do
        $\tilde{\theta}_i \leftarrow$ Sample from posterior $\mathcal{N}\left(\mu_{i,t-1}, \varsigma^2_{i,t-1}\right)$
    end for
    $\boldsymbol{a}_t \leftarrow \arg\min_{\boldsymbol{a} \in \mathcal{I}} f_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{a})$
    Play arm $\boldsymbol{a}_t$, observe feedback $X_{j,t}$ for $j \in \boldsymbol{a}_t$
    Compute $\boldsymbol{\mu}_t, \varsigma_t$ with feedback using Eqs. 2 and 3
end for
```

3.2 Regret analysis of Thompson sampling for minimax paths

We use the technique to analyze the Bayesian regret of Thompson Sampling for general bandit problems introduced by Russo and Van Roy (2014) and further elaborated by Slivkins (2019), carefully adapting it to our problem setting. This technique was originally devised to enable convenient conversion of existing UCB regret analyses to Thompson Sampling, but can also be applied to new TS applications. Here, we do a novel extension to combinatorial bandits with minimax super-arm cost functions, which includes establishing concentration properties for the mean estimates of the *non-linear* super-arm costs. In the rest of this section, we outline the most important steps of the proof of Theorem 1, leaving technical details to the supplementary material (Online Resource 1). In the analysis, for convenience, we assume that $T \geq d$.

Theorem 1 *The Bayesian regret of Algorithm* 1 *is* $\mathcal{O}(d\sqrt{T \log T})$.

We initially define a sequence of upper and lower confidence bounds, for each time step t:

$$U_t(\boldsymbol{a}) := f_{\hat{\boldsymbol{\theta}}_{t-1}}(\boldsymbol{a}) + \max_{i \in \boldsymbol{a}} \sqrt{\frac{32 \log T}{N_{t-1}(i)}}$$

$$L_t(\boldsymbol{a}) := f_{\hat{\boldsymbol{\theta}}_{t-1}}(\boldsymbol{a}) - \max_{i \in \boldsymbol{a}} \sqrt{\frac{32 \log T}{N_{t-1}(i)}}$$

where $\hat{\theta}_{i,t}$ is the average feedback of base arm $i \in \mathcal{A}$ until time t, $\hat{\boldsymbol{\theta}}_t$ is the average feedback vector for all arms in $\mathcal{A}$, and $N_t(i)$ is the number of times base arm $i \in \mathcal{A}$ has been played as part of a super arm until time t.

Lemma 2 *For Algorithm* 1, *we have that*:

$$\text{BayesRegret}(T) = \sum_{t\in[T]} \mathbb{E}\left[U_t(\boldsymbol{a}_t) - L_t(\boldsymbol{a}_t)\right] + \sum_{t\in[T]} \mathbb{E}\left[f_{\boldsymbol{\theta}^*}(\boldsymbol{a}_t) - U_t(\boldsymbol{a}_t)\right] + \sum_{t\in[T]} \mathbb{E}\left[L_t(\boldsymbol{a}^*) - f_{\boldsymbol{\theta}^*}(\boldsymbol{a}^*)\right].$$

This Bayesian regret decomposition is a direct application of Proposition 1 of Russo and Van Roy (2014). It utilizes the fact that given the history of selected arms and received feedback until time t, the played super arm $\boldsymbol{a}_t$ and the best possible super arm $\boldsymbol{a}^* := \arg\min_{\boldsymbol{a}\in\mathcal{I}} f_{\boldsymbol{\theta}^*}(\boldsymbol{a})$ are identically distributed under Thompson Sampling. Furthermore, also given the history, $U_t(\boldsymbol{a})$ and $L_t(\boldsymbol{a})$ are deterministic functions of the super arm $\boldsymbol{a}$. This enables the decomposition of the regret into terms of the expected confidence width, the expected overestimation of the super arm with least mean cost, and the expected underestimation of the selected super arm. By showing that $f_{\boldsymbol{\theta}^*}(\boldsymbol{a}) \in [L_t(\boldsymbol{a}), U_t(\boldsymbol{a})]$ with high probability, we can bound the last two of these terms.

Lemma 3 *For any* $t \in [T]$, *we have that* $\mathbb{E}\left[f_{\boldsymbol{\theta}^*}(\boldsymbol{a}_t) - U_t(\boldsymbol{a}_t)\right] \leq \frac{4d}{T}$ *and* $\mathbb{E}\left[L_t(\boldsymbol{a}^*) - f_{\boldsymbol{\theta}^*}(\boldsymbol{a}^*)\right] \leq \frac{4d}{T}$.

Both terms are bounded in the same way, for which we need a few intermediary results. Focusing on the underestimation of the played super arm, we can see that:

$$\begin{aligned} &\mathbb{E}\left[f_{\boldsymbol{\theta}^*}(\boldsymbol{a}_t) - U_t(\boldsymbol{a}_t)\right] \\ &= \mathbb{E}\left[f_{\boldsymbol{\theta}^*}(\boldsymbol{a}_t) - f_{\hat{\boldsymbol{\theta}}_{t-1}}(\boldsymbol{a}_t) - \max_{i\in\boldsymbol{a}_t} \sqrt{\frac{32\log T}{N_{t-1}(i)}}\right]. \end{aligned}$$

First, in Lemma 4, the difference between the true mean cost $f_{\boldsymbol{\theta}^*}(\boldsymbol{a})$ of a super arm $\boldsymbol{a}$ and the corresponding estimated mean $f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{a})$ is bounded. The resulting upper bound is the maximum of the differences of the true and estimated means of each individual base arm feedback, such that:

Lemma 4 *For any super arm* $\boldsymbol{a} \in \mathcal{I}$ *and time step* $t \in [T]$, *we have that* $|f_{\boldsymbol{\theta}^*}(\boldsymbol{a}) - f_{\hat{\boldsymbol{\theta}}_{t-1}}(\boldsymbol{a})| \leq 2\max_{i\in\boldsymbol{a}} |\theta_i^* - \hat{\theta}_{i,t-1}|$.

This is achieved by decomposing the absolute value into a sum of the positive and negative portions of the difference, then bounding each individually. Focusing on the positive portion by assuming that $f_{\boldsymbol{\theta}^*}(\boldsymbol{a}) \geq f_{\hat{\boldsymbol{\theta}}_{t-1}}(\boldsymbol{a})$, and letting $Z_i \sim \mathcal{N}(\hat{\theta}_{i,t-1}, \sigma_i^2)$, $Y_i \sim \mathcal{N}(\theta_i^*, \sigma_i^2)$, $\delta_{i,t-1} := \theta_i^* - \hat{\theta}_{i,t-1}$ and $Q_i := Y_i - \delta_{i,t-1}$, for $i \in \boldsymbol{a}$, we can see that:

$$\begin{aligned}
& f_{\theta^*}(\boldsymbol{a}) - f_{\hat{\theta}_{t-1}}(\boldsymbol{a}) \\
& = \mathbb{E}\left[\max_{i \in \boldsymbol{a}} Y_i\right] - \mathbb{E}\left[\max_{i \in \boldsymbol{a}} Z_i\right] \\
& = \mathbb{E}\left[\max_{i \in \boldsymbol{a}} (Q_i + \delta_{i,t-1})\right] - \mathbb{E}\left[\max_{i \in \boldsymbol{a}} Z_i\right] \\
& \leq \mathbb{E}\left[\max_{i \in \boldsymbol{a}} Q_i\right] + \max_{i \in \boldsymbol{a}} \delta_{i,t-1} - \mathbb{E}\left[\max_{i \in \boldsymbol{a}} Z_i\right] \\
& = \max_{i \in \boldsymbol{a}} \delta_{i,t-1} \quad .
\end{aligned}$$

The negative portion is bounded in the same way, directly leading to the result of Lemma 4. With this result, we can proceed with Lemma 3, where we let $[x]^+ := \max(0, x)$:

$$\begin{aligned}
& \mathbb{E}\left[2 \max_{i \in \boldsymbol{a}_t} |\delta_{i,t-1}| - \max_{i \in \boldsymbol{a}_t} \sqrt{\frac{32 \log T}{N_{t-1}(i)}}\right] \\
& \leq \mathbb{E}\left[2 \max_{i \in \boldsymbol{a}_t} \left[|\delta_{i,t-1}| - \sqrt{\frac{8 \log T}{N_{t-1}(i)}}\right]^+\right] \\
& \leq 2 \sum_{i \in \mathcal{A}} \mathbb{E}\left[\left[|\delta_{i,t-1}| - \sqrt{\frac{8 \log T}{N_{t-1}(i)}}\right]^+\right] \\
& = 2 \sum_{i \in \mathcal{A}} \left(\Pr\left\{|\delta_{i,t-1}| > \sqrt{\frac{8 \log T}{N_{t-1}(i)}}\right\} \cdot \right.
\end{aligned} \tag{6}$$

$$\left. \mathbb{E}\left[|\delta_{i,t-1}| - \sqrt{\frac{8 \log T}{N_{t-1}(i)}} \;\middle|\; |\delta_{i,t-1}| > \sqrt{\frac{8 \log T}{N_{t-1}(i)}}\right]\right) \tag{7}$$

The probability in Eq. 6 is of the event that the difference between the estimated and true means of an arm i exceeds the confidence radius $\sqrt{8 \log T / N_{t-1}(i)}$, while Eq. 7 is the expected difference conditional on that event. We bound Eq. 6 with Lemma 5 and Eq. 7 with Lemma 6.

Lemma 5 $\Pr\left\{\forall t \in [T]\ \forall i \in \mathcal{A},\ \ |\delta_{i,t-1}| \leq \sqrt{\frac{8 \log T}{N_{t-1}(i)}}\right\} \geq 1 - \frac{2}{T}$.

It is now sufficient to show that the difference $\delta_{i,t-1}$ is small for all base arms $i \in \mathcal{A}$ with high probability, which we accomplish using a standard concentration analysis through application of Hoeffding's inequality and union bounds.

Lemma 6 *For any $t \in [T]$ and $i \in \mathcal{A}$, we have*

$$\mathbb{E}\left[|\delta_{i,t-1}| - \sqrt{\frac{8 \log T}{N_{t-1}(i)}} \;\middle|\; |\delta_{i,t-1}| > \sqrt{\frac{8 \log T}{N_{t-1}(i)}}\right] \leq 1.$$

Though the rewards are unbounded, this expectation can be bounded by utilizing the fact that the mean of a truncated Gaussian distribution is increasing in the mean of the distribution before truncation, by Theorem 2 of Horrace (2015). We can see that:

$$\begin{aligned}
&\mathbb{E}\left[|\delta_{i,t-1}| - \sqrt{\frac{8\log T}{N_{t-1}(i)}} \;\middle|\; |\delta_{i,t-1}| > \sqrt{\frac{8\log T}{N_{t-1}(i)}}\right] \\
&= \mathbb{E}\left[\delta_{i,t-1} - \sqrt{\frac{8\log T}{N_{t-1}(i)}} \;\middle|\; \delta_{i,t-1} - \sqrt{\frac{8\log T}{N_{t-1}(i)}} > 0\right] \\
&\le \mathbb{E}\left[\delta_{i,t-1} \;\middle|\; \delta_{i,t-1} > 0\right].
\end{aligned}$$

We know that $\delta_{i,t-1}$ is zero-mean Gaussian with variance at most one, hence $\mathbb{E}\left[\delta_{i,t-1} \;\middle|\; \delta_{i,t-1} > 0\right] \le 1$.

With the result from Lemma 3, the last two terms of the regret decomposition in Lemma 2 are bounded by constants in T. Focusing on the remaining term, we just need to show that $\sum_{t\in[T]} \mathbb{E}\left[U_t(\boldsymbol{a}_t) - L_t(\boldsymbol{a}_t)\right] \le \mathcal{O}(d\sqrt{T\log T})$ to prove Theorem 1:

$$\begin{aligned}
&\sum_{t\in[T]} \mathbb{E}\left[U_t(\boldsymbol{a}_t) - L_t(\boldsymbol{a}_t)\right] \\
&= \sqrt{128\log T} \sum_{t\in[T]} \mathbb{E}\left[\max_{i\in\boldsymbol{a}_t} \frac{1}{\sqrt{N_{t-1}(i)}}\right] \\
&\le \sqrt{128\log T} \sum_{t\in[T]} \mathbb{E}\left[\sum_{i\in\boldsymbol{a}_t} \frac{1}{\sqrt{N_{t-1}(i)}}\right] \\
&= \sqrt{128\log T} \sum_{i\in\mathcal{A}} \mathbb{E}\left[\sum_{t:i\in\boldsymbol{a}_t} \frac{1}{\sqrt{N_{t-1}(i)}}\right] \\
&\le \sqrt{128\log T} \sum_{i\in\mathcal{A}} \mathbb{E}\left[2\sqrt{N_T(i)}\right] \\
&\le \sqrt{128\log T} \cdot \mathbb{E}\left[2\sqrt{d\sum_{i\in\mathcal{A}} N_T(i)}\right] \\
&\le \sqrt{128\log T} \cdot \mathbb{E}\left[2\sqrt{d^2 T}\right] \\
&= 2d\sqrt{128T\log T} \\
&= \mathcal{O}(d\sqrt{T\log T}).
\end{aligned}$$

We note that the final upper bound is tight up to a polylogarithmic factor, according to existing lower bounds for combinatorial semi-bandit problems (Kveton et al., 2015).

3.3 Thompson sampling with approximate objective

Unfortunately, exact expressions for computing the expected maximum of Gaussian random variables only exist when the variables are few. In other words, we cannot compute $f_\theta(\boldsymbol{a})$ exactly for a super arm $\boldsymbol{a}$ containing many base arms, necessitating some form of approximation approach. While it is possible to approximate $f_\theta(\boldsymbol{a})$ through e.g., Monte

Carlo simulations, we want to be able to perform the cost minimization step using a computationally efficient oracle.

We note that, even with the capability to exactly compute $f_\theta(\boldsymbol{a})$, it would not be feasible to solve the minimization problem in line 6 of Algorithm 1. The expected cost $f_\theta(\boldsymbol{a})$ of a super arm $\boldsymbol{a}$ (i.e., the expected maximum base arm feedback) depends not only on the individual expected values of the base arm feedback distributions, but also on the shape of the joint distribution of all base arms in $\boldsymbol{a}$. Due to this fact, the stochastic version of the minimization problem lacks the property of optimal substructure (i.e., an optimal path does not necessarily consist of optimal sub-paths). For the deterministic version of the problem, as defined in Eq. 1, the presence of this property enables the usage of computationally efficient dynamic programming strategies, like Dijkstra's algorithm, which is consequently infeasible with the objective in Algorithm 1.

Therefore, we propose the approximation method outlined in Algorithm 2, where the minimization step of line 6 has been modified from Algorithm 1 with an alternative super arm cost function $\tilde{f}_{\tilde{\theta}}(\boldsymbol{a}) := \max_{i \in \boldsymbol{a}} \tilde{\theta}_i$. Switching objectives, from finding the super arm which minimizes the expected maximum base arm feedback, to instead minimize the maximum expected feedback, has the benefit of allowing us to utilize the efficient deterministic minimax path algorithms introduced earlier for both directed and undirected graphs. For directed graphs, the modified version of Dijkstra's algorithm by Berman and Handler (1987) has a worst-case running time of $\mathcal{O}(|E| + |V| \log |V|)$ with an efficient implementation using Fibonacci heaps (Fredman & Tarjan, 1987). Similarly, for undirected graphs, finding an MST (and subsequently a minimax path) can be achieved using Prim's algorithm (Prim, 1957), with the same running time of $\mathcal{O}(|E| + |V| \log |V|)$ if Fibonacci heaps are used. The other operations performed for each $t \in [T]$ in Algorithm 2 (i.e., the posterior samples and updates) have a combined running time of, at worst, $\mathcal{O}(|E|)$. The same oracles are also used for the baseline algorithms evaluated in Sections 4.1 and 4.2, with comparable running times.

It is possible to use alternative notions of regret to evaluate combinatorial bandit algorithms with approximate oracles (Chen et al., 2013, 2016). For our experimental evaluation of Algorithm 2, we introduce the following definition of approximate regret:

$$\text{ApproxRegret(T)} := \left(\sum_{t \in [T]} \tilde{f}_{\theta^*}(\boldsymbol{a}_t) \right) - T \cdot \min_{\boldsymbol{a} \in \mathcal{I}} \tilde{f}_{\theta^*}(\boldsymbol{a}).$$

An alternative Bayesian bandit algorithm which can be used with the alternative objective is BayesUCB (Kaufmann et al., 2012), which we use as a baseline for our experiments. Like Thompson Sampling, BayesUCB has been adapted to combinatorial semi-bandit settings (Nuara et al., 2018; Åkerblom et al., 2020). Whereas Thompson Sampling in Algorithm 2 encourages exploration by applying the oracle to parameters sampled from the posterior distribution, with BayesUCB, the oracle is instead applied to optimistic estimates based on the posterior distribution. In practice, this is accomplished for our cost minimization problem by using lower quantiles of the posterior distribution of each base arm. This principle of selecting plausibly optimal arms is called *optimism in the face of uncertainty* and is the underlying idea of all bandit algorithms based on UCB.

We note that while in BayesUCB, as outlined in Algorithm 1 of Kaufmann et al. (2012), the horizon is used to calculate UCB values, the authors of that work also explain that upper quantiles of order $1 - 1/t$ (calculated without the horizon) achieve good results in practice. For that reason, we use lower quantiles of order $1/t$ in the version of BayesUCB studied in this work, making it an *anytime* algorithm, like Thompson Sampling.

```
Algorithm 2 TS for minimax paths (approximation)
Input: Prior parameters $\boldsymbol{\mu}_0, \varsigma_0$
For each base arm, play a super arm which contains it.
for $t \leftarrow 1, \dots, T$ do
    for $i \in \mathcal{A}$ do
        $\tilde{\theta}_i \leftarrow$ Sample from posterior $\mathcal{N}\left(\mu_{i,t-1}, \varsigma^2_{i,t-1}\right)$
    end for
    $\boldsymbol{a_t} \leftarrow \arg\min_{\boldsymbol{a} \in \mathcal{I}} \max_{i \in \boldsymbol{a}} \tilde{\theta}_i$
    Play arm $\boldsymbol{a}_t$, observe feedback $X_{j,t}$ for $j \in \boldsymbol{a}_t$
    Compute $\boldsymbol{\mu}_t, \varsigma_t$ with feedback using Eq. 2 and 3
end for
```

To connect the different objectives in Algorithm 1 and Algorithm 2, we note that by Jensen's inequality, $\tilde{f}_{\tilde{\theta}}(\boldsymbol{a}) \leq f_{\tilde{\theta}}(\boldsymbol{a})$ and that the approximation objective consequently will underestimate super arm costs. However, we establish an upper bound on this difference through Theorem 7.

Theorem 7 *Given the optimal super arm $\boldsymbol{a}^*$ for Algorithm 1 and the optimal super arm $\tilde{\boldsymbol{a}}^*$ for Algorithm 2, we have that $f_{\theta^*}(\tilde{\boldsymbol{a}}^*) - f_{\theta^*}(\boldsymbol{a}^*) \leq \sqrt{2 \log d}$.*

For any super arm $\boldsymbol{a} \in \mathcal{I}$, let Y_i for $i \in \boldsymbol{a}$ be Gaussian random variables with $Y_i \sim \mathcal{N}(\theta^*_i, \sigma^2_i)$. Furthermore, let $W_i := Y_i - \theta^*_i$, such that $W_i \sim \mathcal{N}(0, \sigma^2_i)$. Then, the following holds:

$$\begin{aligned}
&\mathbb{E}\left[\max_{i \in \boldsymbol{a}} Y_i\right] \\
&= \mathbb{E}\left[\max_{i \in \boldsymbol{a}} (W_i + \theta^*_i)\right] \\
&\leq \mathbb{E}\left[\max_{i \in \boldsymbol{a}} W_i\right] + \max_{i \in \boldsymbol{a}} \mathbb{E}\left[Y_i\right] \\
&\leq \sqrt{2 \log d} + \max_{i \in \boldsymbol{a}} \mathbb{E}\left[Y_i\right],
\end{aligned}$$

where the last inequality is due to Lemma 9 of Orabona et al. (2015) and since $\sigma^2_i \leq 1$ for all $i \in \boldsymbol{a}$. We also note that, by Jensen's inequality, we have $\max_{i \in \boldsymbol{a}} \mathbb{E}\left[Y_i\right] \leq \mathbb{E}\left[\max_{i \in \boldsymbol{a}} Y_i\right]$. Moreover, by definition we know that $\boldsymbol{a}^* = \arg\min_{\boldsymbol{a} \in \mathcal{I}} \mathbb{E}\left[\max_{i \in \boldsymbol{a}} Y_i\right]$ and $\tilde{\boldsymbol{a}}^* = \arg\min_{\boldsymbol{a} \in \mathcal{I}} \max_{i \in \boldsymbol{a}} \mathbb{E}\left[Y_i\right]$. Consequently, we have,

$$\max_{i \in \tilde{\boldsymbol{a}}^*} \mathbb{E}\left[Y_i\right] \leq \max_{i \in \boldsymbol{a}^*} \mathbb{E}\left[Y_i\right] \leq \mathbb{E}\left[\max_{i \in \boldsymbol{a}^*} Y_i\right] \leq \mathbb{E}\left[\max_{i \in \tilde{\boldsymbol{a}}^*} Y_i\right] \leq \sqrt{2 \log d} + \max_{i \in \tilde{\boldsymbol{a}}^*} \mathbb{E}\left[Y_i\right].$$

Hence, we can conclude that

$$f_{\theta^*}(\tilde{\boldsymbol{a}}^*) - f_{\theta^*}(\boldsymbol{a}^*) = \mathbb{E}\left[\max_{i \in \tilde{\boldsymbol{a}}^*} Y_i\right] - \mathbb{E}\left[\max_{i \in \boldsymbol{a}^*} Y_i\right] \leq \sqrt{2 \log d}.$$

Table 1 A description of the road networks.

	Road network			
	Eindhoven	Manhattan	Oslo	Salzburg
#Node	7501	4593	8153	2921
#Edge	10776	8130	11192	3848
Avg. Degree	2.873	3.540	2.745	2.634

In other words, Theorem 7 holds and the optimal solutions of the exact Algorithm 1 and the approximate Algorithm 2 differ by at most $\sqrt{2\log d}$. This bound is independent of the mean vector $\boldsymbol{\theta}^*$, depending only on the number of base arms and that the variance is bounded.

4 Experimental results

In this section, we conduct bottleneck identification experiments using Algorithm 2 for two real-world applications, i) road (transport) networks, and ii) collaboration (social) networks. These experiments are performed with an extended version of the simulation framework by Russo et al. (2018) and evaluated using our approximate definition of regret. In addition, we compare Algorithm 1 to Algorithm 2 through a toy example.

4.1 Road networks

A bottleneck in a network is a segment of a path in the network that obstructs or stops flow. Identification of bottlenecks in a road network is a vital tool for traffic planners to analyze the network and prevent congestion. In this application, our goal is to find the bottleneck between a source and a target, i.e., a road segment which is necessary to pass and also has minimal traffic flow. In the road network model, we let the nodes represent intersections and the directed edges represent road segments, with travel time divided by distance (seconds per meter) as edge weights. The bottleneck between a pair of intersections is the minimum bottleneck over all paths connecting them, where the bottleneck for each of these paths is the largest weight over all road segments along it. Note that in order for the bottleneck between a pair of intersections to have a meaning, there needs to exist at least one path connecting them.

Table 2 Average cumulative regret and corresponding standard error (SE) over five runs, at the horizon $T = 6000$, for Thompson Sampling (TS), BayesUCB (B-UCB), ϵ_t-greedy agents (e-GR-N and e-GR-E), and Greedy (GR) agent.

	Road network			
	Eindhoven	Manhattan	Oslo	Salzburg
TS	271.8 ± 46.3	449.0 ± 53.0	226.7 ± 69.1	131.7 ± 13.4
B-UCB	483.8 ± 112.9	670.5 ± 68.9	339.7 ± 88.3	259.2 ± 41.4
e-GR-N	838.8 ± 191.2	1232.9 ± 72.1	379.3 ± 115.9	653.8 ± 175.7
e-GR-E	928.7 ± 191.7	1120.0 ± 113.8	405.7 ± 113.1	609.4 ± 87.0
GR	936.9 ± 223.2	1116.7 ± 142.9	511.2 ± 140.1	1159.0 ± 155.2

Fig. 1 Cumulative regret averaged over 5 runs with shaded standard error bars, for Thompson Sampling (TS), Bayes UCB (B-UCB), ϵ_t-greedy agents (e-GR-N and e-GR-E), and greedy (GR) with horizon $T = 6000$, on Eindhoven (**a**), Manhattan (**c**), Oslo (**e**) and Salzburg (**g**) road networks. Visualizations of the paths explored by the TS agent are shown in red, for Eindhoven (**b**), Manhattan (**d**), Oslo (**f**) and Salzburg (**h**) road networks. Opacity illustrates the exploration of each of the road segments. ▶

We collect road networks of four cities, shown in Table 1, from OpenStreetMap contributors (2017), where the average travel time as well as the distance is provided for each (directed) edge. We simulate an environment with the stochastic edge weights sampled from $w_e \sim \mathcal{N}(\theta^*_e, \sigma^2_e)$, where the observation noise is $\sigma_e = 0.4$. For the experiments, the environment samples the true unknown mean θ^*_e from the known prior $\theta^*_e \sim \mathcal{N}(\mu_{e,0}, {\varsigma_{e,0}}^2)$, where $\varsigma_{e,0} = 0.4$s/m, and $\mu_{e,0}$ is the average travel time divided by distance provided by OpenStreetMap (OSM).

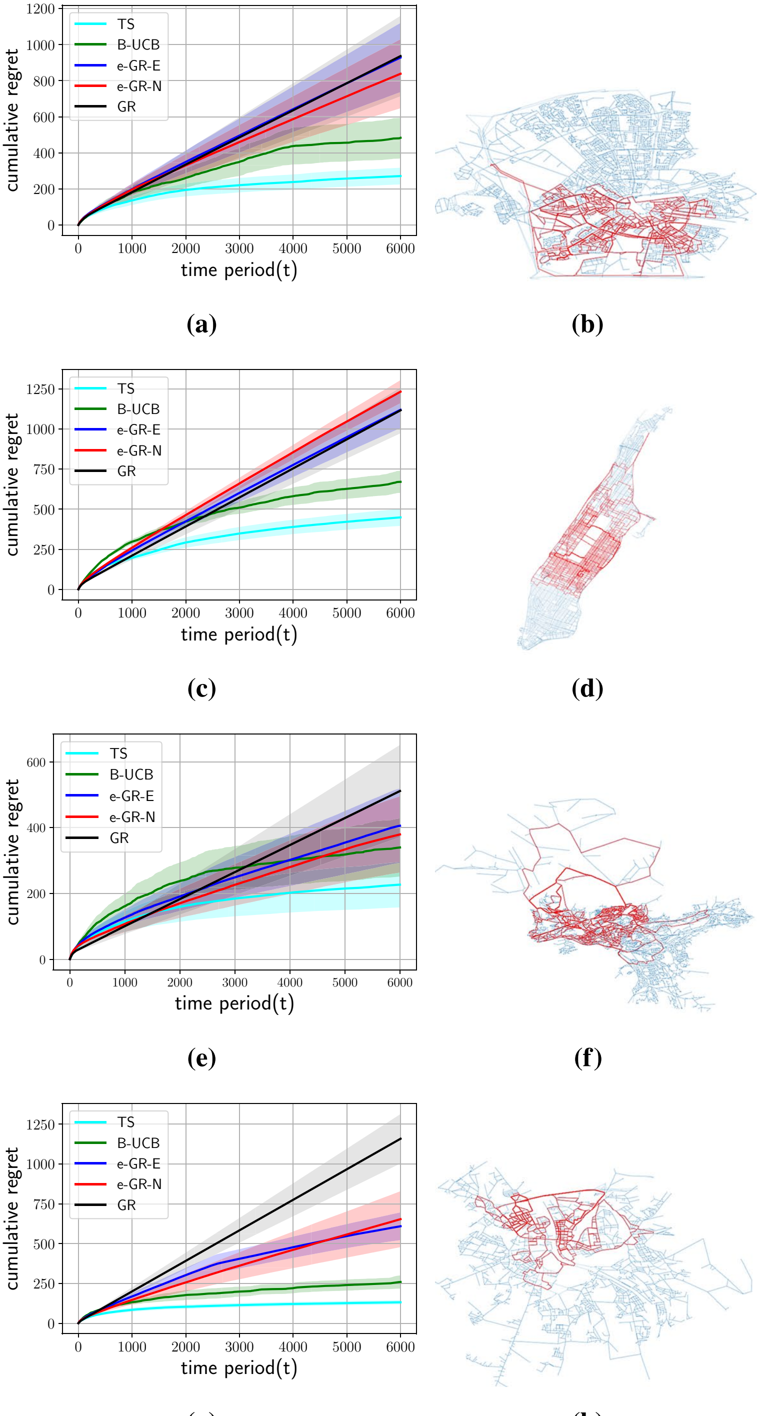
We consider one greedy agent (GR) and two ϵ_t-greedy agents (e-GR) as baselines. The greedy agent (GR) always chooses the path with the lowest current estimate of expected cost. In each time step, each e-GR agent, with probability ϵ_t decreasing with t (specifically, we let $\epsilon_t = \min(1, 1/\sqrt{t})$), chooses a random path, and acts like the greedy agent otherwise. In our experiments, we implement the two e-GR agents based on the combinatorial version of ϵ_t-greedy introduced in Algorithm 1 in the Supplementary Material of Chen et al. (2013). The first e-GR agent chooses a path between the source and the target containing a uniformly chosen random node (e-GR-N), and the second e-GR agent chooses a path with a uniformly selected random edge (e-GR-E). We evaluate how the performance of the Thompson Sampling agent (TS) and the BayesUCB agent (B-UCB) compare to the baselines. We run the simulations with all five agents for each road network and report the cumulative regret at a given horizon T, averaged over five repetitions. The horizon is chosen such that the instant regret is almost stabilized for the agents.

Table 2 shows the average cumulative regrets and their corresponding standard error over five runs at the horizon T. For all four road networks, the TS agent incurs the lowest average cumulative regret and standard error over five runs. Then, B-UCB follows TS and yields a better result than the baselines (GR and both e-GR variants).

Figure 1 illustrates the average cumulative regret with standard error (SE) bars on the road networks of the four aforementioned cities. For Eindhoven, Figure 1a shows the average cumulative regret, where at horizon $T = 6000$ the TS agent yields the lowest cumulative regret. Then, B-UCB follows TS and achieves a better result compared to the other baselines. As time progresses, we can see that first TS and then B-UCB start saturating by performing sufficient exploration. With respect to the SE bars, there are differences between the five agents. The TS agent has the smallest SE bars. Figure 1b visualizes the Eindhoven road network, where the paths explored by the TS agent are shown in red. The road segments explored (tried) more often by the TS agent are displayed more opaque. Figure 1c, e, and g show the average cumulative regret with SE bars for Manhattan, Oslo, and Salzburg, respectively. The results show that TS incurs the lowest cumulative regret and smallest SE bars. Then, B-UCB follows TS in both aspects and obtains a better result than the other baselines.

4.2 Collaboration network

We consider a collaboration network from computational geometry (Geom) (Jones, 2002) as an application of our approach to social networks. More specifically, we use the version


TS
B-UCB
e-GR-E
e-GR-N
GR
cumulative regret
time period(t)
0
200
400
600
800
1000
1200
250
500
750
1250
0
1000
2000
3000
4000
5000
6000

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

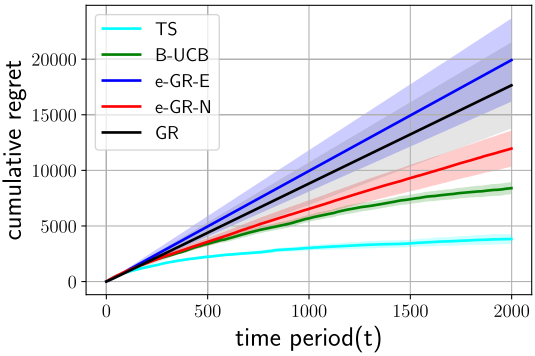


Fig. 2 Cumulative regret averaged over 5 runs with shaded standard error bars, for Thompson Sampling (TS), Bayes UCB (B-UCB), ϵ_t-greedy agnets (e-GR-N and e-GR-E), and greedy (GR) with horizon $T = 2000$ for the collaboration network.

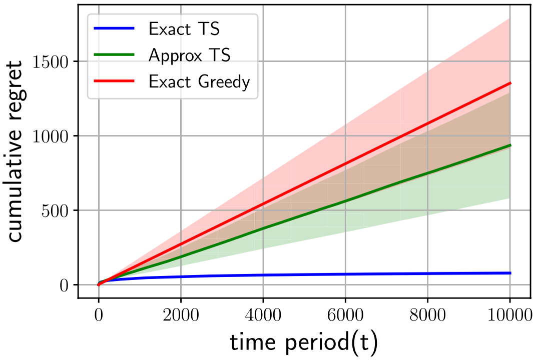


Fig. 3 Experimental results with average cumulative regret on the toy example, with $T = 10000$, on exact TS, approximate TS and exact greedy.

provided by Handcock et al. (2003) and distributed among the Pajek datasets (Batagelj et al., 2006) where certain author duplicates, occurring in minor or major name variations, have been merged. The Handcock et al. (2003) version is based on the BibTeX bibliography (Beebe, 2002), to which the database from Jones (2002) has been exported. The network has 9072 vertices representing the authors and 22577 edges with the edge weights representing the number of mutual works between a pair of authors.

We simulate an environment where each edge weight is sampled as $w_e \sim \mathcal{N}(\theta_e^*, \sigma_e^2)$, within which θ_e^* is regarded as the true (negative) mean number of shared publications between a pair of authors linked by the edge e, and the observation noise is $\sigma_e = 5$. Furthermore, in this experiment, while the true negative mean number of mutual publications are assumed (by the agent) to be distributed according to the prior $\theta_e^* \sim \mathcal{N}(\mu_{e,0}, \varsigma_{e,0}^2)$ with $\varsigma_{e,0} = 10$, we instead generate the mean from a wider prior $\theta_e^* \sim \mathcal{N}(\mu_{e,0}, 20^2)$, simulating a scenario where the prior belief of the agent is too high. The assumed mean $\mu_{e,0}$ of the prior is however consistent with the distribution from which θ_e^* is sampled, and is directly determined by the pairwise negative number of mutual collaborations from the dataset by Handcock et al. (2003).

Figure 2 shows the cumulative regret, averaged over five runs for the different agents with horizon $T = 2000$, again chosen such that the regret is stabilized for all agents. One can see that the TS agent reaches the lowest cumulative regret, similar to the experimental studies on road networks.

4.3 Exact objective toy example

While it is not feasible to evaluate Algorithm 1 on graphs representing real-life transportation or social networks, it is possible for small synthetic graphs. We construct a graph consisting of 6 nodes and 10 edges, with the source and target nodes connected by four paths of length 2 and four paths of length 3. For each edge e, we use the sample the mean from a standard Gaussian prior, such that $\theta_e^* \sim \mathcal{N}(0, 1)$. The stochastic weights are then generated in each time step t such that $w_{e,t} \sim \mathcal{N}(\theta_e^*, 1)$.

In order to calculate the expected cost of each path, we use existing exact expressions for the expected maximum of two (Clark, 1961) and three (Lo, 2020; DasGupta, 2021) independent Gaussian random variables. Instead of using an oracle, we simply enumerate the paths to find the one with minimum expected cost.

In Figure 3, we compare Algorithm 1 (TS with exact objective) and Algorithm 2 (TS with approximate objective) using the exact notion of (cumulative) regret as defined in Eq. 4. Furthermore, we include a greedy baseline which also uses the exact objective. We use a horizon of $T = 10000$ and average the results over 20 experiments, wherein each algorithm is applied to a problem instance sampled from the prior.

We can see that the regret of exact TS quickly saturates, while approximate TS and the greedy method tend to end up in sub-optimal solutions. For approximate TS, this is to be expected since optimal arms for the exact and approximate problems may be different. It is worth noting, however, that approximate TS performs better than the exact greedy method on average.

5 Conclusion

We developed an online learning framework for bottleneck identification in networks via minimax paths. In particular, we modeled this task as a combinatorial semi-bandit problem for which we proposed a combinatorial version of Thompson Sampling. We then established an upper bound on the Bayesian regret of the Thompson Sampling method. To deal with the computational intractability of the problem, we devised an alternative problem formulation which approximates the original objective. Finally, we investigated the framework on several directed and undirected real-world networks from transport and collaboration domains. Our experimental results demonstrate its effectiveness compared to alternatives such as greedy and B-UCB methods.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10994-022-06270-0.

Acknowledgements This work is partially funded by the Strategic Vehicle Research and Innovation Programme (FFI) of Sweden, through the project EENE (reference number: 2018-01937). We want to thank the reviewers for their helpful comments and suggestions. We also want to thank Emilio Jorge, Emil Carlsson and Tobias Johansson for insightful discussions around the proofs.

Author contributions All authors have contributed significantly to the conception, design and writing of this work. The theoretical analysis was primarily performed by Niklas Åkerblom, and the experimental study was primarily performed by Fazeleh Sadat Hoseini.

Funding Open access funding provided by Chalmers University of Technology. Niklas Åkerblom is employed by Volvo Car Corporation with funding from Sweden's Innovation Agency (Vinnova), through the Strategic Vehicle Research and Innovation Programme (FFI) of Sweden (project EENE, reference number: 2018-01937).

Availability of data and material Only public datasets, clearly cited in the text, are used.

Declarations

Conflict of interest There is no conflict of interest.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

Code availability Contact the authors for access to the code.

Supplementary information The detailed proof of Theorem 1 is provided in Online Resource 1.

References

Agrawal, S., Goyal, N. (2012) Analysis of thompson sampling for the multi-armed bandit problem. In: Mannor, S., Srebro, N., Williamson, R.C. (eds.) Proceedings of the 25th Annual Conference on Learning Theory. Proceedings of Machine Learning Research. vol. 23, pp. 39–13926. PMLR, Edinburgh, Scotland.

Åkerblom, N., Chen, Y., Haghir Chehreghani, M. (2020) An online learning framework for energy-efficient navigation of electric vehicles. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI), pp. 2051–2057. 10.24963/ijcai.2020/284

Auer, P. (2003). Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research., 3*(null), 397–422.

Batagelj, V., Mrvar, A. (2006) Pajek datasets. http://vlado.fmf.uni-lj.si/pub/networks/data/. Accessed: 2021-09-08

Beebe, N.H.F. (2002) Nelson H. F. Beebe's Bibliographies Page. http://www.math.utah.edu/~beebe/bibliographies.html. Accessed: 2021-09-08

Berman, O., & Handler, G. Y. (1987). Optimal minimax path of a single service unit on a network to nonservice destinations. *Transportation Science, 21*(2), 115–122. https://doi.org/10.1287/trsc.21.2.115

Cesa-Bianchi, N., & Lugosi, G. (2012). Combinatorial bandits. *Journal of Computer and System Sciences, 78*(5), 1404–1422. https://doi.org/10.1016/j.jcss.2012.01.001.JCSS

Chapelle, O., & Li, L. (2011). An empirical evaluation of thompson sampling. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems.* London: Springer.

Chen, W., Hu, W., Li, F., Li, J., Liu, Y., Lu, P. (2016) Combinatorial multi-armed bandit with general reward functions. In: Lee, D, Sugiyama, M, Luxburg, U, Guyon, I, Garnett, R (eds) Advances in Neural Information Processing Systems. p 29

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

Clark, C. E. (1961). The greatest of a finite set of random variables. *Operations Research, 9*(2), 145–162. https://doi.org/10.1287/opre.9.2.145

DasGupta, A. (2021) A Formula for the Expected Value of the Maximum of Three Independent Normals and a Sparse High Dimensional Case. https://www.stat.purdue.edu/~dasgupta/orderstat.pdf. Accessed: 2021-09-07 (n.d.)

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik, 1*(1), 269–271. https://doi.org/10.1007/bf01386390

Du, Y., Kuroki, Y., Chen, W. (2021) Combinatorial Pure Exploration with Bottleneck Reward Function. arXiv. 1048550/ARXIV.2102.12094

Fredman, M. L., & Tarjan, R. E. (1987). Fibonacci heaps and their uses in improved network optimization algorithms. *Journal ACM, 34*(3), 596–615. https://doi.org/10.1145/28869.28874

Gai, Y., Krishnamachari, B., & Jain, R. (2012). Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Transactions on Networking, 20*(5), 1466–1478. https://doi.org/10.1109/TNET.2011.2181864

Graepel, T., Candela, J.Q.n., Borchert, T., Herbrich, R. (2010) 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, pp. 13–20. Omnipress, Madison, WI, USA

Haghir Chehreghani, M. (2020). Unsupervised representation learning with minimax distance measures. *Machine Learning, 109*(11), 2063–2097. https://doi.org/10.1007/s10994-020-05886-4

Handcock, M.S., Hunter, D., Butts, C.T., M., G.S., Morris, M. (2003) Statnet: An R package for the Statistical Modeling of Social Networks. http://www.csde.washington.edu/statnet. Accessed: 2021-09-08

Hansen, P. (1980). Bicriterion path problems. In G. Fandel & T. Gal (Eds.), *Multiple criteria decision making theory and application* (pp. 109–127). Berlin: Springer.

Hennessy, D. A., & Wiesenthal, D. L. (1999). Traffic congestion, driver stress, and driver aggression. *Aggressive Behavior, 25*(6), 409–423. https://doi.org/10.1002/(SICI)1098-2337(1999)25:6<409::AID-AB2>3.0.CO;2-0

Horrace, W. C. (2015). Moments of the truncated normal distribution. *Journal of Productivity Analysis, 43*(2), 133–138. https://doi.org/10.1007/s11123-013-0381-8

Hu, T. C. (1961). The maximum capacity route problem. *Operations Research, 9*, 898–900. https://doi.org/10.1287/opre.9.6.898

Jones, B. (2002) Computational geometry database. http://jeffe.cs.illinois.edu/compgeom/biblios.html. Accessed: 2021-09-08

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

Kaufmann, E., Korda, N., Munos, R. (2012) Thompson sampling: An asymptotically optimal finite-time analysis. In: Algorithmic Learning Theory - 23rd International Conference, ALT, pp. 199–213. 10.1007/978-3-642-34106-9_18

Kim, K.-H., Choi, S. (2007) Neighbor search with global geometry: A minimax message passing algorithm. In: Proceedings of the 24th International Conference on Machine Learning. ICML '07, pp. 401–408, New York, NY, USA. 10.1145/1273496.1273547

Kveton, B., Wen, Z., Ashkan, A., Szepesvari, C. (2015) Tight Regret Bounds for Stochastic Combinatorial Semi-Bandits. In: Lebanon, G., Vishwanathan, S.V.N. (eds.) Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 38, pp. 535–543. San Diego, California, USA

Lai, T. L., & Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics, 6*(1), 4–22. https://doi.org/10.1016/0196-8858(85)90002-8

Lo, C. (2020) Improving hardware design reuse through design-space exploration. PhD thesis, University of Toronto (Canada)

Liu, K., Zhao, Q. (2012) 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). pp. 232–237.

Nuara, A., Trovo, F., Gatti, N., Restelli, M. (2018) A combinatorial-bandit algorithm for the online joint bid/budget optimization of pay-per-click advertising campaigns. In: Thirty-Second AAAI Conference on Artificial Intelligence. 10.1609/aaai.v32i1.11888

OpenStreetMap contributors (2017) Planet dump retrieved from. www.openstreetmap.org. Accessed: 2021-09-08

Orabona, F., Pal, D. (2015) Optimal Non-Asymptotic Lower Bound on the Minimax Regret of Learning with Expert Advice. arXiv. 1048550/ARXIV.1511.02176

Pollack, M. (1960). The maximum capacity through a network. *Operations Research*. https://doi.org/10.1287/opre.8.5.733

Prim, R. C. (1957). Shortest connection networks and some generalizations. *The Bell System Technical Journal, 36*(6), 1389–1401. https://doi.org/10.1002/j.1538-7305.1957.tb01515.x

Riquelme, C., Tucker, G., Snoek, J. (2018) Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. In: International Conference on Learning Representations.

Russo, D., & Van Roy, B. (2014). Learning to optimize via posterior sampling. *Mathematics of Operations Research, 39*(4), 1221–1243. https://doi.org/10.1287/moor.2014.0650

Russo, D. J., Roy, B. V., Kazerouni, A., Osband, I., & Wen, Z. (2018). A tutorial on thompson sampling. *Foundations and Trends in Machine Learning., 11*(1), 1–96. https://doi.org/10.1561/2200000070

Seshadri, R., & Srinivasan, K. K. (2010). Algorithm for determining most reliable travel time path on network with normally distributed and correlated link travel times. *Transportation research record, 2196*(1), 83–92. https://doi.org/10.3141/2196-09

Shacham, N. (1992). Multicast routing of hierarchical data. In: [Conference Record] SUPERCOMM/ICC 92 Discovering a New World of Communications, pp. 1217–12213. doi: https://doi.org/10.1109/ICC.1992.268047

Slivkins, A. (2019). Introduction to multi-armed bandits. *Foundations and Trends in Machine Learning, 12*(1–2), 1–286. https://doi.org/10.1561/2200000068

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

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

Zou, Z., Proutiere, A., Johansson, M. (2014) Online shortest path routing: The value of information. In: 2014 American Control Conference. pp. 2142–2147. 10.1109/ACC.2014.6859133