

Bayesian adversarial multi-node bandit for optimal smart grid protection against cyber attacks [★]

Jianyu Xu ^a, Bin Liu ^b, Huadong Mo ^c, Daoyi Dong ^c

^a*International Business School Suzhou, Xi'an Jiaotong-Liverpool University, Suzhou, China*

^b*Department of Management Science, University of Strathclyde, G1 1XQ, Glasgow, UK*

^c*School of Engineering and Information Technology, University of New South Wales, Canberra, ACT 2600, Australia*

Abstract

The cybersecurity of smart grids has become one of key problems in developing reliable modern power and energy systems. This paper introduces a non-stationary adversarial cost with a variation constraint for smart grids and enables us to investigate the problem of optimal smart grid protection against cyber attacks in a relatively practical scenario. In particular, a Bayesian multi-node bandit (MNB) model with adversarial costs is constructed and a new regret function is defined for this model. An algorithm called Thompson-Hedge algorithm is presented to solve the problem and the superior performance of the proposed algorithm is proven in terms of the convergence rate of the regret function. The applicability of the algorithm to real smart grid scenarios is verified and the performance of the algorithm is also demonstrated by numerical examples.

Key words: Multi-node bandit; reinforcement learning; Bayesian updating; cyber attack; smart grid.

1 Introduction

The upgrade of traditional grids to smart grids has brought many benefits to the overall management of power and energy systems, including higher reliability, better efficiency, improved integration of renewable energy resources, more flexible choice for stakeholders and lower operation cost (Konstantelos et al., 2016; Pogaku et al., 2007; Yu et al., 2015). However, the core technologies, *e.g.*, communication techniques and SCADA systems (Abiri-Jahromi et al., 2020; Khalili et al., 2020; Rana, Xiang and Wang, 2018; Todescato et al., 2020), which deliver advantages of smart grids, also open the grids to vulnerabilities that already exist in the Information and Communications Technology world. Now, those vulnerabilities pose threats to smart grids, such as denial of service (DoS) attacks, false data injection,

replay attacks, privacy data theft and sabotage of critical infrastructure (Gallo et al., 2020; Mo and Sansavini, 2017; Zhu and Martínez, 2013). In addition, the failures in a smart grid caused by cyber attacks can easily cascade to other highly dependent critical infrastructure sectors, such as transportation systems, wastewater systems, health care systems and banking systems, resulting in extensive physical damage and social and economic disruption (Abiri-Jahromi et al., 2020; Che et al., 2018).

While governments, the private sector and academia are recognising the cyber vulnerability of smart grids, the likelihood and impact of a cyber attack are difficult to quantify. Furthermore, for a smart grid, there may be mandatory standards and operation requirements from the grid stakeholders. Current risk management strategies are generally qualitative or heuristic (Patsakis et al., 2018). In these strategies, some assumptions, *e.g.*, constant reward with respect to successful anti cyber attack (Rana, Li and Su, 2017; Smith and Paté-Cornell, 2018), may be unrealistic for most smart grids.

This paper presents a probabilistic risk analysis framework to enhance the smart grid cyber security. In particular, the dynamical and stochastic characteristics of smart grids, such as uncertain demands, are taken into ac-

[★] This work is partially supported by Fudan Scholar and Special Research Grant, UNSW, and by the Australian Research Council's Discovery Projects funding scheme under Project DP190101566. Corresponding author Huadong Mo. Tel. +61-2-62688683. Fax +61-2-62688683

Email addresses: Jianyu.Xu@xjtlu.edu.cn (Jianyu Xu), b.liu@strath.ac.uk (Bin Liu), huadong.mo@adfa.edu.au (Huadong Mo), d.dong@adfa.edu.au (Daoyi Dong).

count to investigate the effect of defending strategies on the real operation cost. The Optimal Power Flow model (Zhang and Papachristodoulou, 2015) is applied to a 11-node radial smart grid originated from the Elia grid, Belgium. Compared with the existing studies that focus on the inherent risk (Mo and Sansavini, 2017; Zhang and Papachristodoulou, 2015), such as the natural degradation and uncertain renewable energy resources for better maintenance actions and power dispatch, this paper concerns about impact of the external threat - cyber attacks on the operation cost for effective deployment of cyber defense teams. In previous work, the cost of each attack on a node is usually assumed to be a constant (Smith and Paté-Cornell, 2018). Nevertheless, through investigating some practical scenarios, it has been found that the costs are more likely to be decided by some adversarial factors from the nature. Therefore, an adversarial cost sequence associated with each node is assumed, and a widely used variation constraint is introduced on each cost sequence. To cope with the objective of sequential decision strategy, the problem is formulated using the reinforcement learning framework (Li et al., 2020; Littman, 2015; Sutton and Barto, 2018). Specifically, the Bayesian prior method (Smith and Paté-Cornell, 2018) is employed for the model parameters and the problem is formulated as a Bayesian adversarial multi-node bandit (MNB) model. In addition, a Bayesian minmax type regret function is constructed, which is subject to the learning context.

Research on online algorithms for adversarial MNB problems started from Auer et al. (2002), which is later extended to general adversarial reinforcement learning models by (Even-Dar et al., 2009). Recently, Besbes et al. (2019) implemented the classical method in Auer et al. (2002) for the adversarial bandit problem assuming cost sequences with restricted variation and achieved the state-of-the-art performance for such problems. An obvious gap when using these methods in the problem of our work is that they do not adapt to the Bayesian framework and cannot take advantages of the additional information provided by the Bayesian assumptions. As a result, these algorithms may be ineffective in Bayesian problems in terms of the convergence rate of the regret. An alternative feasible method is proposed in Smith and Paté-Cornell (2018), where model parameters are considered to be system states included in the state space. Therefore, the problem is no longer Bayesian and can be solved using existing algorithms for stochastic MNB. However, this method suffers from a tremendous state space and is always computationally hard even in problems with moderate sizes.

To cope with the abovementioned technical challenge, an online learning method is developed in this study which integrates the Thompson sampling method (Russo and Van Roy, 2014) with the classical Hedge algorithm (Auer et al., 2002; Russo and Van Roy, 2014). Our algorithm takes advantages of both algorithms. In particular, Thompson sampling method is used to cope with the

Bayesian framework and Hedge algorithm is applied to the adversarial costs. A theoretical bound on the regret function is proved in the proposed algorithm, which is superior to the typical regret bound by the state-of-the-art algorithms for the same problem. The applicability and numerical performance of the proposed algorithm is also illustrated through real data and simulation studies.

The main contributions of our work are summarized as follows.

1. A relatively practical adversarial reward with restricted variation constraint is proposed for the MNB model. The prior information of the model parameters is incorporated through a Bayesian framework and a Bayesian adversarial MNB model is formulated.
2. A Bayesian sup regret is defined as the criterion function for decision making objective of the proposed problem. A new online algorithm, called Thompson-Hedge algorithm, is developed to solve the problem. The convergence rate of the Bayesian sup regret for the proposed algorithm is theoretically proven.
3. Real smart grid data are used to demonstrate the applicability of the proposed method and numerical results show that the proposed algorithm can achieve the state-of-the-art performance.

This paper is organized as follows. Section 2 formulates the problem as a Bayesian reinforcement learning model and constructs the regret function. In Section 3, an algorithm called Thompson-Hedge algorithm is developed for our learning model and a theoretical upper bound on the regret function is established. Section 4 verifies the feasibility of our model and algorithm using a real data set. In Section 5, a comparative study is investigated on the performance between our algorithm and a state-of-the-art algorithm. Section 6 presents concluding remarks.

2 Problem formulation

In the electricity network with cyber attacks, an attacker launches a coordinated attack using multiple attack vectors, because the smart grid communication networks are physically distributed and highly heterogeneous. The successful rate of such attack behavior is data-driven, which follows a Poisson distribution, as demonstrated by the empirical study from U.S. Department of Energy (Smith and Paté-Cornell, 2018). The network defender aims to optimally allocate cyber defense teams among nodes in the network, via probing one node per day. Such a defending action thwarts all attempted cyber attacks to that node on that day, and also helps update his/her belief about the uncertain successful rates of attack. The above interaction between the attacker and the defender has the sequential decision-making nature and leads itself to a Bayesian MNB model. This model employs

proactively defense teams that traditionally respond to cyber threats after they occur.

The considered attack scenario in a realistic smart grid is that the DoS attacks block the demand response (DR) messages and dispatch commands (Pillitteri and Brewer, 2014; Smith and Paté-Cornell, 2018). The DoS attacks can be accomplished by flooding the communication channel, *e.g.*, the one between the demand response automation server and customer systems or the one between the control center and power plant router, with other messages and commands, or by tampering with the communication channel (Amin et al., 2013; Pillitteri and Brewer, 2014; Smith and Paté-Cornell, 2018). Above actions can prevent legitimate DR messages and dispatch commands being received and transmitted, *i.e.*, depriving authorized access or control to customer systems and power plant, resulting in demand not being responded and power plant not being controlled (Pillitteri and Brewer, 2014). Therefore, the impact of successful DoS attacks on the optimal power flow model of the smart grid is described by making the target node temporarily unavailable and disconnected from the grid, which is illustrated in Section 4.1.

Consider a smart grid with N nodes that suffers cyber attacks. Let $\mathbb{N} \triangleq \{1, \dots, N\}$ be the set of all nodes. At each time $t = 1, 2, \dots$, the operator's action is to choose a node to probe. If $i \in \mathbb{N}$ is probed, the operator observes a (random) number of $K_{i,t}$ cyber attacks on node i .

The historical cyber incident data from US Department of Energy (Fig. 2 of Smith and Paté-Cornell (2018)) has demonstrated that the arrival interval of successful cyber attacks on per node can be well described by a truncated Poisson distribution, and there is no record of multiple cyber attacks (maximum is 3) on per node per day. Therefore, the physical meaning is that in real applications, there should not be more than 3 cyber attacks arriving per node in a defined time interval. In addition, the Palm-Khintchine theorem (Heyman and Sobel, 2004; Smith and Paté-Cornell, 2018) justifies that the aggregate arrivals from many sources (no need to be strict Poisson) approach a truncated Poisson distribution in the limit time interval. It is assumed that for all $i \in \mathbb{N}$, $\{K_{i,t}, t = 1, 2, \dots\}$ is a sequence of independent and identically distributed random variable sequence drawn from a Poisson distribution truncated at $m > 0$ with rate λ_i , *i.e.*, for all $k = 1, 2, \dots$, and $t = 1, 2, \dots$,

$$Pr(K_{i,t} = k | \lambda_i) = \begin{cases} \frac{\lambda_i^k}{k!} e^{-\lambda_i}, & k < m \\ \sum_{j=m}^{\infty} \frac{\lambda_i^j}{j!} e^{-\lambda_i}, & k = m \\ 0, & \text{otherwise} \end{cases}.$$

In particular, the average number of attacks on each

node $i \in \mathbb{N}$, denoted by μ_i , is

$$\mu_i = \sum_{k=1}^{m-1} k \cdot \frac{\lambda_i^k}{k!} e^{-\lambda_i} + m \sum_{k=m}^{\infty} \frac{\lambda_i^k}{k!} e^{-\lambda_i}.$$

Let $\lambda = (\lambda_1, \dots, \lambda_N)$ and assume a foresight belief on λ in terms of a prior distribution $\mathcal{F} = (\mathcal{F}_1, \dots, \mathcal{F}_N)$ over λ , where $\lambda_i \sim \mathcal{F}_i$ for all $i \in \mathbb{N}$. In particular, it is assumed that each \mathcal{F}_i is a gamma distribution with parameters (α_i, β_i) and all \mathcal{F}_i 's are mutually independent with each other. Whenever k cyber attacks are observed at node $i \in \mathbb{N}$, the parameters (α_i, β_i) are updated through Bayesian posterior as $(\alpha_i, \beta_i) \rightarrow (\alpha_i + k, \beta_i + 1)$.

At time $t = 1, 2, \dots$, a cost $c_{i,t}$ is calculated from an optimization process for all $i \in \mathbb{N}$. Without loss of generality, it is assumed that all $c_{i,t}$'s are normalized such that $c_{i,t} \in [0, 1/m]$ for all $i \in \mathbb{N}$ and $t = 1, 2, \dots$. The process uses some inputs (*e.g.*, external factors) that are decided by an adversary (environment). Without ambiguity, the cost vector $c_t = (c_{1,t}, \dots, c_{N,t})$ is assumed to be decided by the adversary at each time $t = 1, 2, \dots$. If node $i \in \mathbb{N}$ is probed, the operator avoids incurring a total cost of $X(i, t) = K_{i,t} \cdot c_{i,t}$, $X(i, t) \in [0, 1]$. Equivalently, a reward of $K_{i,t} \cdot c_{i,t}$ is achieved by probing node i at time t . An admissible policy π is a sequence of mappings $\pi = \{\pi_1, \pi_2, \dots\}$, where each π_t is a mapping from historical observations to the set of all probability distributions on \mathbb{N} . To distinguish, let $\{i_1, i_2, \dots\}$ represent the sequence of chosen nodes. The axioms of the problem are formulated in the problem protocol below.

Problem protocol:

For each time $t = 1, 2, \dots$

- 1: The adversary decides an cost $c_{i,t}$ for all $i \in \mathbb{N}$.
 - 2: The operator probes a node $i_t \in \mathbb{N}$ according to π_t and observes the number of cyber attacks on this node to be $k_{i_t,t}$.
 - 3: All $c_{i,t}$'s are revealed to the operator.
 - 4: A reward $k_{i_t,t} \cdot c_{i_t,t}$ is achieved by the operator.
-

For notational convenience, let $\theta = (\alpha_1, \beta_1, \dots, \alpha_N, \beta_N)$ denote the initial parameter vector and $\vec{c}_T = \{c_1, \dots, c_T\}$ the cost sequence up to time T . To proceed, the optimal policy and the regret function is firstly formulated given the parameter vector θ and cost sequence \vec{c}_T . The mean reward of node $i \in \mathbb{N}$ at time t is $\mathbf{E}(K_{i,t} \cdot c_{i,t}) = \mu_i \cdot c_{i,t}$. Since the reward sequence from each node is not stationary due to $c_{i,t}$, for all $t = 1, 2, \dots$, the optimal node is defined as $i_t^* \triangleq \arg \max_{i \in \mathbb{N}} \mu_i \cdot c_{i,t}$ and a non-stationary optimal policy is supposed to choose node i_t^* at time $t = 1, 2, \dots$. Thus, for any admissible policy π , given θ

and \vec{c}_T , the regret function up to time T is defined as

$$R^\pi(\lambda, \vec{c}_T, T) \triangleq \sum_{t=1}^T \mu_{i_t^*} \cdot c_{i_t^*, t} - \mathbf{E}^\pi \left(\sum_{t=1}^T \mu_{\pi_t} \cdot c_{\pi_t, t} \mid \lambda, \vec{c}_T \right), \quad (1)$$

where \mathbf{E}^π means the expectation taken with respect to the (random) sequence $\{i_1, \dots, i_T\}$ generated by π .

A constraint is imposed upon the adversary by introducing a sequence of sets $\{\mathbb{C}_t \subset [0, 1]^t\}_{t \geq 2}$, that for all time $t = 1, 2, \dots$, $\vec{c}_T \in \mathbb{C}_t$. Then, given θ , the regret function with respect to the worst case up to time $T > 0$ is

$$\sup_{\vec{c}_T \in \mathbb{C}_T} R^\pi(\lambda, \vec{c}_T, T). \quad (2)$$

Note that $\sup_{\vec{c}_T \in \mathbb{C}_T} R(\lambda, \vec{c}_T, T)$ is supposed to measure the performance with the action sequence uniformly on all possible cost sequences. Finally, incorporating the prior distribution \mathcal{F} on θ , the Bayesian regret function with respect to the worst case is defined as

$$\mathcal{R}^\pi(\mathbb{C}_T, T) \triangleq \mathbf{E}_{\lambda \sim \mathcal{F}} \left[\sup_{\vec{c}_T \in \mathbb{C}_T} R^\pi(\lambda, \vec{c}_T, T) \right]. \quad (3)$$

To distinguish between the regret functions in (1), (2) and (3), they are named as regret, sup regret and Bayesian sup regret, respectively, for brevity.

By analyzing some real databases (such as the Elia Grid in Section 4), the sequence of sets $\{\mathbb{C}_T \subset [0, 1]^T\}_{T \geq 2}$ is formulated with constraint on the cost. It is concluded from the statistical analysis of the real data that the temporal variations of the cost sequence of each node $i \in \mathbb{N}$, i.e., $\{c_{i, t+1} - c_{i, t}\}_{t \geq 1}$, are stationary and uniformly upper bounded by a value, which only relies on i and is small compared to the average value of $\{c_{i, t}\}_{t \geq 1}$. Therefore, it is assumed that for each node the cumulative temporal variation up to time $t \geq 1$ is upper bounded by a linear function of t . Based on this assumption, a uniform upper bound on the cumulative temporal variation for all nodes is introduced and $\{\mathbb{C}_T \subset [0, 1]^T\}_{T \geq 2}$ is constructed as follows.

$$\mathbb{C}_T \triangleq \{ \{c_{i, t}\}_{t=1}^T \in [0, 1]^T, i \in \mathbb{N} : \sum_{t=1}^T \max_{i \in \mathbb{N}} |c_{i, t+1} - c_{i, t}| \leq \mathcal{V}_T \}, \forall T \geq 2, \quad (4)$$

where $\{\mathcal{V}_T\}_{T \geq 1}$ is a sequence of positive numbers. Since $c_{i, t} \in [0, 1/m]$, $\forall i \in \mathbb{N}$ and $t \geq 1$, $\max_{i \in \mathbb{N}} |c_{i, t+1} - c_{i, t}| \ll 1/m$ holds for all $t \geq 1$. Thus, $\mathcal{V}_T \leq O(T)$ and $\mathcal{V}_T/T \ll 1/m$. Meanwhile, \mathcal{V}_T is monotonically increasing in T . Based on these considerations, the following assumption on \mathcal{V}_T is proposed.

Assumption 1 *There exists $T_0 \geq 1$, such that for all $T \geq T_0$, $1/m \leq \mathcal{V}_T \leq T/m$.*

The restriction of $T \geq T_0$ for some T_0 in Assumption 1 is necessary to further present our results. Actually the condition $1/m \leq \mathcal{V}_T$ cannot be directly concluded based on the previous discussions. Since \mathcal{V}_T is the accumulation of $\max_{i \in \mathbb{N}} |c_{i, t+1} - c_{i, t}|$ and we suppose that $\max_{i \in \mathbb{N}} |c_{i, t+1} - c_{i, t}| \ll 1/m$. Hence, when T is small, \mathcal{V}_T may not satisfy the condition $1/m \leq \mathcal{V}_T$. It is worth noting that a justification of the existence of T_0 is useful to guarantee the performance of our algorithm, while it is not necessary to know the exact value of T_0 . Prior knowledge can be incorporated to ensure the existence of T_0 . For example, if the difference term $\max_{i \in \mathbb{N}} |c_{i, t+1} - c_{i, t}|$ fails to rapidly become infinitely small (which is natural when the cost sequences are non-stationary), then \mathcal{V}_T will surely be larger than $1/m$ when T is large enough. Examples of securing an estimated value for T_0 can be found in the smart grid application of Section 4, where \mathcal{V}_T is estimated to grow linearly in T and a threshold time for \mathcal{V}_T to exceed $1/m$ can be easily calculated.

3 Thompson-Hedge algorithm

In this section, an online learning algorithm is developed to optimize the Bayesian sup regret for our problem. The algorithm has two layers. In the outer layer, at each time $t \geq 1$, the algorithm uses Thompson sampling (posterior sampling) method to sample a parameter vector $(\lambda_1, \dots, \lambda_N)$ from the posterior distributions of all λ_i 's based on the historical observations. Then, in the inner layer, a so-called sub-algorithm is fed with the sampled parameters. The sub-algorithm returns an action $\pi_t \in \mathbb{N}$. At the end of this loop, the algorithm chooses node π_t , observes the reward from π_t and updates the posterior distribution of λ_{π_t} . In the end of this section, a theorem is provided to characterize the convergence rate of the regret function.

Since Hedge algorithm is applied as the sub-algorithm in the inner layer, the proposed algorithm is named as Thompson-Hedge algorithm. Before formally presenting our algorithm, the Hedge algorithm is firstly introduced. Hedge algorithm is a classical algorithm designed for the adversarial MNB problem with full feedback (Freund and Schapire, 1999). It is a randomized algorithm that maintains a weight $\omega_t(i)$ for each $i \in \mathbb{N}$. Hedge algorithm then chooses node $i \in \mathbb{N}$ with a probability proportional to $\omega_t(i)$ at time t . Subsequently, the weight for each node is updated according to the observed reward of this node. The algorithm chooses a size Δ and restarts updating the weights of each node every Δ times. The set of intervals between two successive restarting epochs (including the first restarting time) is called a batch. Since Hedge algorithm is designed for non-Bayesian adversarial MNB, in order to use it under the Bayesian framework, one needs to modify the original Hedge algorithm and feed

it by the value of parameter λ . Denote $\text{Hedge}(\lambda)$ as the modified Hedge algorithm used in our Bayesian adversarial MNB fed by λ . The algorithm is summarized in Algorithm 1.

Algorithm 1: Hedge(λ) Algorithm

Initialization: Parameter vector λ , $\varepsilon = (1 - \sqrt{\ln N/2T})^{-1}$, calculate μ_i using λ_i .

- 1: **for** $b = 1, 2, \dots, \lceil T/\Delta \rceil$ **do**
- 2: For all $i \in \mathbb{N}$, $\omega_1^i = 1$.
- 3: **while** $t \leq \min\{T, b \cdot \Delta\}$ **do**
- 4: For each $i \in \mathbb{N}$, set

$$p_t^i = \frac{\omega_t^i}{\sum_{j \in \mathbb{N}} \omega_t^j}.$$

- 5: Choose a node π_t according to the probability distribution $\{p_t^i\}_{i \in \mathbb{N}}$ and receive a reward $k_{\pi_t, t} \cdot c_{\pi_t, t}$.
- 6: For each node $i \in \mathbb{N}$, update

$$\omega_{t+1}^i = \omega_t^i \varepsilon^{\mu_i \cdot c_{\pi_t, t}}.$$

- 7: Set $t \rightarrow t + 1$.
- 8: **end while**
- 9: **end for**

Hedge(λ) follows the paradigm of the classical Hedge algorithm. The following lemma shows that if the value of parameter λ fed to Hedge(λ) is the true parameter of the Bayesian adversarial MNB model, then Hedge(λ) retains a convergent sup regret with a known upper bound.

Lemma 1 *If the input λ in the Hedge algorithm is the true model parameter and the batch size is chosen to be $\Delta = \lceil (\ln N)^{1/3} (T/m\mathcal{V}_T)^{2/3} \rceil$, then for all $T \geq N$, the sup regret by the Hedge(λ) algorithm is upper bounded by*

$$\sup_{\vec{c}_T \in \mathcal{C}_T} R^\pi(\lambda, \vec{c}_T, T) \leq (8 + 2\sqrt{2}) (m\mathcal{V}_T \ln N)^{1/3} (T)^{2/3}. \quad (5)$$

Detailed proof of Lemma 1 is given in the Appendix. In the proof of our main result of Theorem 1, the conclusion of Lemma 1 is used as an intermediate benchmark result.

In the proposed Thompson-Hedge algorithm, parameter λ is sampled from its posterior distribution in each epoch. Then, the sampled parameter is fed to the inner algorithm Hedge(λ) and a node is chosen correspondingly. In particular, the algorithm chooses a node according to the probability weight of each node. Then, the algorithm observes the costs and updates the posterior distribution of λ . Next, a sampled parameter $\hat{\lambda}$ is drawn from the posterior distribution of λ . Finally, the algorithm updates the weight of each node for next epoch

using both the observed costs and $\hat{\lambda}$. Gamma(α, β) is used to represent Gamma distribution with parameters (α, β). The proposed Thompson-Hedge algorithm is summarized in Algorithm 2.

Algorithm 2: Thompson-Hedge Algorithm

Initialization: Parameter θ , batch size Δ , $\varepsilon = (1 - \sqrt{\ln N/2T})^{-1}$, $t=1$.

- 1: **for** $b = 1, 2, \dots, \lceil T/\Delta \rceil$ **do**
- 2: For all $i \in \mathbb{N}$, $\omega_1^i = 1$.
- 3: **while** $t \leq \min\{T, b \cdot \Delta\}$ **do**
- 4: For each $i \in \mathbb{N}$, set

$$p_t^i = \frac{\omega_t^i}{\sum_{j \in \mathbb{N}} \omega_t^j}.$$

- 5: Choose a node i_t according to the probability distribution $\{p_t^i\}_{i \in \mathbb{N}}$ and receive a reward $k_{i_t, t} \cdot c_{i_t, t}$.
- 6: For node i_t , update $(\alpha_{i_t}, \beta_{i_t}) \rightarrow (\alpha_{i_t} + k_{i_t, t}, \beta_{i_t} + 1)$.
- 7: For each node $i \in \mathbb{N}$, sample $\hat{\lambda}_i \sim \text{Gamma}(\alpha_i, \beta_i)$, calculate $\hat{\mu}_i$ using $\hat{\lambda}_i$, and update

$$\omega_{t+1}^i = \omega_t^i \cdot \varepsilon^{\hat{\mu}_i \cdot c_{i_t, t}}.$$

- 8: Set $t \rightarrow t + 1$.
- 9: **end while**
- 10: **end for**

Before the main result (Theorem 1) is presented, the following lemma is introduced that will be used for the proof of the main result.

Lemma 2 *[(Osband et al., 2013), Lemma 2] If F is the true distribution of λ , and $\hat{\lambda}$ is the sampled parameter in epoch t , then for any $\sigma(\mathbb{H}_t)$ -measurable function g , it follows*

$$\mathbf{E}_\lambda [g(\lambda) | \mathbb{H}_t] = \mathbf{E}_{\hat{\lambda}} [g(\hat{\lambda}) | \mathbb{H}_t].$$

Lemma 2 shows a central result in Bayesian learning area. The sampled parameter from the posterior distribution each time can be “considered” as the true parameter in the sense that any deterministic function using it as an argument independently has the same expectation. On the basis of Lemma 1 and Lemma 2, the main theorem is presented as below, which establishes an upper bound on the Bayesian sup regret for our Thompson-Hedge algorithm and thus indicates its convergence rate.

Theorem 1 *Let π be the Thompson-Hedge algorithm with batch size $\Delta = \lceil (\ln N)^{1/3} (T/m\mathcal{V}_T)^{2/3} \rceil$, \mathcal{C}_T be defined in (4), and T_0 be the same value as defined in Assumption 1, then for all $T \geq T_0$, the Bayesian sup regret*

$\mathcal{R}^\pi(\mathbb{C}_T, T)$ can be upper bounded by

$$\mathcal{R}^\pi(\mathbb{C}_T, T) \leq (8 + 2\sqrt{2}) (m\mathcal{V}_T \ln N)^{1/3} (T)^{2/3}. \quad (6)$$

PROOF. The basic idea for the proof is sketched as follows. The Bayesian sup regret by Thompson-Hedge algorithm is firstly decomposed into two parts: the regret of the introduced Hedge(λ) algorithm and the difference of performance between Thompson-Hedge and Hedge algorithms. Since the sup regret by Hedge algorithm is bounded by Lemma 1, Theorem 1 can then be proved by upper bounding the performance difference between the two algorithms. To distinguish between two algorithms in the work, π^{TH} and π^H are used for Thompson-Hedge and Hedge algorithms correspondingly. Moreover, R^{TH} and R^H are denoted as the regret function of π^{TH} and π^H , respectively.

a) Separation of the target regret \mathcal{R}^{TH} .

In the first step, the Bayesian sup regret \mathcal{R}^{TH} is separated into a combination of two terms. In Step b) and Step c), the upper bound of these two terms are obtained separately and therefore, the upper bound of \mathcal{R}^{TH} can be readily obtained. The separation of \mathcal{R}^{TH} is given as below.

$$\begin{aligned} \mathcal{R}^{TH}(\mathbb{C}_T, T) &= \mathbf{E}_{\lambda \sim \mathcal{F}} \left[\sup_{\vec{c}_T \in \mathbb{C}_T} \left\{ \sum_{t=1}^T \mu_{i_t^*} \cdot c_{i_t^*, t} \right. \right. \\ &\quad \left. \left. - \mathbf{E}^{TH} \left(\sum_{t=1}^T \mu_{i_t} \cdot c_{i_t, t} \mid \lambda, \vec{c}_T \right) \right\} \right] \\ &\leq \mathbf{E}_{\lambda \sim \mathcal{F}} \left[\sup_{\vec{c}_T \in \mathbb{C}_T} \left\{ \sum_{t=1}^T \mu_{i_t^*} \cdot c_{i_t^*, t} \right. \right. \\ &\quad \left. \left. - \mathbf{E}^H \left(\sum_{t=1}^T \mu_{i_t} \cdot c_{i_t, t} \mid \lambda, \vec{c}_T \right) \right\} \right] \\ &\quad + \mathbf{E}_{\lambda \sim \mathcal{F}} \left[\sup_{\vec{c}_T \in \mathbb{C}_T} \left\{ \mathbf{E}^H \left(\sum_{t=1}^T \mu_{i_t} \cdot c_{i_t, t} \mid \lambda, \vec{c}_T \right) \right. \right. \\ &\quad \left. \left. - \mathbf{E}^{TH} \left(\sum_{t=1}^T \mu_{i_t} \cdot c_{i_t, t} \mid \lambda, \vec{c}_T \right) \right\} \right]. \quad (7) \end{aligned}$$

For notational convenience, denote the above two terms in the two square brackets by Λ_1 and Λ_2 , respectively. The relation in (7) can be rewritten as

$$\mathcal{R}^{TH} = \Lambda_1 + \Lambda_2$$

Note that Λ_1 is closely associated with the sup regret of π^H and can be bounded based on the result in Lemma 1. Λ_2 is the difference between the return of π^{TH} and π^H that will be bounded in the following step.

b) Upper bound of Λ_1

Λ_1 is rewritten as

$$\Lambda_1 = \mathbf{E}_{\lambda \sim \mathcal{F}} \left[\sup_{\vec{c}_T \in \mathbb{C}_T} R^H(\lambda, \vec{c}_T, T) \right].$$

Note that the upper bound in Lemma 1 holds for any true model parameter λ . Thus the upper bound still holds after taking expectation on λ on both sides of (5). Therefore, it holds that

$$\Lambda_1 \leq (8 + 2\sqrt{2}) (m\mathcal{V}_T \ln N)^{1/3} (T)^{2/3}.$$

c) Upper bound Λ_2

The following clarification is made for notational convenience. λ_t is used for the sampled parameter by Thompson-Hedge algorithm at time t and λ for the true model parameter which is the input of Hedge algorithm. To bound the Bayesian sup regret by Thompson-Hedge algorithm, the difference between the Bayesian sup regret functions by Thompson-Hedge and Hedge algorithms, respectively, is firstly bounded, namely,

$$\begin{aligned} \mathbf{E}_{\lambda \sim \mathcal{F}} \left[\sup_{\vec{c}_T \in \mathbb{C}_T} \left\{ \mathbf{E}^{TH} \left(\sum_{t=1}^T \mu_{i_t} \cdot c_{i_t, t} \mid \lambda, \vec{c}_T \right) \right. \right. \\ \left. \left. - \mathbf{E}^H \left(\sum_{t=1}^T \mu_{i_t} \cdot c_{i_t, t} \mid \lambda, \vec{c}_T \right) \right\} \right], \quad (8) \end{aligned}$$

where \mathbf{E}^{TH} and \mathbf{E}^H have the same meaning as \mathbf{E}^π given π , for Thompson-Hedge algorithm and Hedge algorithm, respectively. Denote the observation history before time $t \geq 2$ as $\mathbb{H}_t = \{i_1, k_{i_1, 1}, c_1, \dots, i_{t-1}, k_{i_{t-1}, t-1}, c_{t-1}\}$. Conditioned on the observation history, the probability weight function $p_t(\cdot)$ in both algorithms is functions of λ and $\lambda(t)$ correspondingly. Moreover, let $p_t(\cdot)$ and $\tilde{p}_t^i(\cdot)$ be the probability functions of Thompson-Hedge and Hedge algorithms, respectively. For any fixed $\vec{c}_T \in \mathbb{C}_T$,

the following relation holds,

$$\begin{aligned}
& \mathbf{E}^{TH} \left(\sum_{t=1}^T \mu_{i_t} \cdot c_{i_t, t} \mid \lambda, \vec{c}_T \right) \\
& - \mathbf{E}^H \left(\sum_{t=1}^T \mu_{i_t} \cdot c_{i_t, t} \mid \lambda, \vec{c}_T \right) \\
& = \mathbf{E}^{TH} \left(\sum_{t=1}^T \mu_{i_t} \cdot c_{i_t, t} \mid \lambda, \vec{c}_T, \mathbb{H}_T \right) \\
& - \mathbf{E}^H \left(\sum_{t=1}^T \mu_{i_t} \cdot c_{i_t, t} \mid \lambda, \vec{c}_T, \mathbb{H}_T \right) \\
& = \sum_{t=1}^T \mathbf{E}_{\lambda_t} [p_t^i(\lambda_t) - \tilde{p}_t^i(\lambda) | \mathbb{H}_t] \mu_i \cdot c_{i, t}. \quad (9)
\end{aligned}$$

At any time $t \geq 1$, note that λ_t is the sampled parameter from the same posterior distribution as the true λ . Meanwhile, $p_t^i(\cdot)$ and $\tilde{p}_t^i(\cdot)$ are the same deterministic function based on \mathbb{H}_t . According to Lemma 2, it follows

$$\mathbf{E}_{\lambda_t, \lambda} [p_t^i(\lambda_t) - \tilde{p}_t^i(\lambda) | \mathbb{H}_t] \mu_i \cdot c_{i, t} = 0.$$

Therefore, for any fixed $\vec{c}_T \in \mathbb{C}_T$, it holds that

$$\begin{aligned}
& \mathbf{E}_{\lambda \sim \mathcal{F}} \left[\left\{ \mathbf{E}^{TH} \left(\sum_{t=1}^T \mu_{i_t} \cdot c_{i_t, t} \mid \lambda, \vec{c}_T \right) \right. \right. \\
& \left. \left. - \mathbf{E}^H \left(\sum_{t=1}^T \mu_{i_t} \cdot c_{i_t, t} \mid \lambda, \vec{c}_T \right) \right\} \right] \\
& = \mathbf{E} \left(\mathbf{E}_{\lambda \sim \mathcal{F}} \left[\left\{ \mathbf{E}^{TH} \left(\sum_{t=1}^T \mu_{i_t} \cdot c_{i_t, t} \mid \lambda, \vec{c}_T, \mathbb{H}_T \right) \right. \right. \right. \right. \\
& \left. \left. \left. - \mathbf{E}^H \left(\sum_{t=1}^T \mu_{i_t} \cdot c_{i_t, t} \mid \lambda, \vec{c}_T, \mathbb{H}_T \right) \right\} \right] \right) \\
& = \mathbf{E} \left(\mathbf{E}_{\lambda_t, \lambda} [p_t^i(\lambda_t) - \tilde{p}_t^i(\lambda) | \mathbb{H}_t] \mu_i \cdot c_{i, t} \right) \\
& = 0.
\end{aligned}$$

Note that the relation above holds for any $\vec{c}_T \in \mathbb{C}_T$, which leads to

$$\begin{aligned}
& \mathbf{E}_{\lambda \sim \mathcal{F}} \left[\sup_{\vec{c}_T \in \mathbb{C}_T} \left\{ \mathbf{E}^{TH} \left(\sum_{t=1}^T \mu_{i_t} \cdot c_{i_t, t} \mid \lambda, \vec{c}_T \right) \right. \right. \\
& \left. \left. - \mathbf{E}^H \left(\sum_{t=1}^T \mu_{i_t} \cdot c_{i_t, t} \mid \lambda, \vec{c}_T \right) \right\} \right] \equiv 0. \quad (10)
\end{aligned}$$

d) *Upper bound of the regret \mathcal{R}^{TH} .*

Finally, the upper bound of the Bayesian sup regret of Thompson-Hedge algorithm is obtained by combining

the results in a), b) and c). In particular, it holds that

$$\begin{aligned}
\mathcal{R}^{TH}(\mathbb{C}_T, T) & \leq \Lambda_1 + \Lambda_2 = \Lambda_1 + 0 \\
& \leq (8 + 2\sqrt{2}) (m\mathcal{V}_T \ln N)^{1/3} (T)^{2/3}.
\end{aligned}$$

Therefore, the proof is concluded.

Remark 1 *Relevant research that considers a similar problem can be found in Besbes et al. (2019). In Besbes et al. (2019), the classical EXP3 type algorithm was used and an upper bound of the order $O((m\mathcal{V}_T N \ln N)^{1/3} (T)^{2/3})$ was obtained for the sup regret. Since the upper bound holds uniformly on the parameter space, the same upper bound also holds for the Bayesian sup regret. If the constant $(8 + 2\sqrt{2})$ in (6) is neglected, our bound outperforms the bound in Besbes et al. (2019) by a term of $O(N^{1/3})$, which implies that the performance of our Thompson-Hedge algorithm is less sensitive to the number of nodes N . It indicates that when considering problems with large scales, our proposed algorithm is supposed to retain a Bayesian sup regret that converges relatively faster. Meanwhile, Besbes et al. (2019) constructs a lower bound of the order $O(\mathcal{V}_T^{1/3} T^{2/3})$ on the regret by any algorithm. Similarly, the lower bound also holds uniformly on all the parameters and adapts to our problem. Therefore, the lower bound shows that our algorithm achieves the order optimality.*

4 Applicability in smart grids

The model is formulated following the learning context in Section 2 and an online learning algorithm is developed in Section 3. This section presents an application case to demonstrate the applicability of the proposed model and method in practical smart grids. In particular, Assumption 1 plays a central role in our model. A real data set is used to verify that Assumption 1 may hold in reality, and thus the proposed method is practical. Section 4.1 introduces the procedure for calculating the reward $c_{i, t}$ in the smart grid. In Section 4.2, linear regression method is used to show a linear growth rate of the critical quantity \mathcal{V}_T in Assumption 1, and thus it is concluded that Assumption 1 holds for the selected data set.

4.1 Operation cost of the smart grid

To facilitate reading, meanings of the variables used in calculating the operation cost of the smart grid, are displayed in different categories as follows. For the operational variables, ϑ_t is the operation state of the smart grid at time t , $P_{i, j}^t$ [kW] is the output of type j power source at Node i , $\bar{P}_{i, j}$ is the rate power of type j power source at Node i , L_i^t is the power load at Node i , \bar{L}_i^t is

the load shedding, *i.e.*, power demand not supplied, at Node i , φ is the state of charge of the energy storage system (ESS) and \bar{C}_{ESS} is the power capacity of the ESS. For the configuration parameters of the smart grid, $B_{i,i'}$ [$1/\Omega$] is the susceptance of the pairs of Nodes (i, i') , δ_i is the voltage angle at Node i , $\Delta\delta$ is the voltage angle difference $(\delta_i - \delta_{i'})$ of two nearby nodes, $A_{i,i'}$ [A] is the ampacity of the pairs of Nodes (i, i') and V [kW] is the nominal voltage of the smart grid. For the cost and price coefficients, Co^{ϑ_t} is the operation cost of the smart grid subject to operation state ϑ_t , Cs_j is the variable operation cost for power source j , $Cf_{i,i'}$ is the variable operation cost for feeder (i, i') , Cp is the penalty cost for power demand not supplied, and Ep^{ϑ_t} is the energy price associated with operation state ϑ_t . T_s is the duration of ϑ_t , \mathbb{P} denotes a subset of power sources, \mathbb{N} denotes the set of all nodes, and \mathbb{F} denotes the set of node pairs with transmission line between them. $s_{i,i'} = 1$ indicates that power cannot be transmitted between Node i and Node i' due to the successful cyber attack on Node i .

The linear Direct Current (DC) power flow model (Mo and Sansavini, 2019; Sahraei-Ardakani and Hedman, 2016) is introduced to interpret the physical meaning of $c_{i,t}$ in the smart grid. At time t , operation state of the distributed generation system (DGS) is denoted by the following vector:

$$\vartheta_t = [P_{i,j}^t L_i^t], \quad (11)$$

where the power sources considered in this work consists of natural gas plant, biomass plant, wind farm, Photovoltaic farm and ESS. These data are all sampled from the historical database provided by the Elia Grid, Belgium¹.

Before calculating $c_{i,t}$, the first objective is to achieve the minimal cost in the presence of load shedding, denoted by Co^{ϑ_t} , by solving the following linear optimization problem

$$\begin{aligned} \min Co^{\vartheta_t} = & \sum_{i \in \mathbb{N}} \sum_{j \in \mathbb{P}} (Cs_j - Ep^{\vartheta_t}) P_{i,j}^t \\ & + \sum_{(i,i') \in \mathbb{F}} Cf_{i,i'} |B_{i,i'} (\delta_i - \delta_{i'})| \\ & + (Cp + Ep^{\vartheta_t}) \sum_{i=1}^N \bar{L}_i^t \end{aligned} \quad (12)$$

subject to

$$L_i^t - \bar{L}_i^t - \sum_{j \in \mathbb{P}} P_{i,j}^t - \sum_{i=1}^N B_{i,i'} (\delta_i - \delta_{i'}) = 0, \quad (13)$$

$$0 \leq P_{i,j} \leq \bar{P}_{i,j}, j \neq ESS, \quad (14)$$

$$|P_{i,j}^t \cdot T_s| \leq \min \left\{ (1 - \varphi) \cdot \bar{C}_{ESS}, \bar{P}_i^j \cdot T_s \right\}, j = ESS, \quad (15)$$

$$|B_{i,i'} (\delta_i - \delta_{i'})| \leq (1 - s_{i,i'}) V A_{i,i'}, \quad (16)$$

where constraint (13) requires that the power generated and consumed is balancing at each node of DGS, constraint (14) requires that the power generated should not be larger than the rated power, constraint (15) indicates that the charging or discharging of ESS should not be larger than the remaining capacity or nominal rate, and constraint (16) indicates that the power flow between two nodes should not be larger than the capacity of the transmission line. Eq. (12) is the objective function of a typical linear DC Optimal Power Flow model, which aims to minimize the total operation cost including the cost of generating power, the cost of running feeders and the penalty cost of demand not supplied, when satisfying physical constraints (13)-(16) of DGS.

The linear DC optimal power flow model is configured in the Matlab and can be solved using the well-known Simplex method, where the values of configuration parameters have been given in Mo and Sansavini (2019) and the computation time of operation cost for each practical operation state is around 0.006 second in Gurobi. This computation time contributes most to the total simulation time and therefore our proposed algorithm can be implemented in real time.

The physical meaning of $c_{i,t}$ in the smart grid can be defined as the difference between the operation cost of the DGS without cyber attacks and the operation cost of the DGS given the Node i^* is temporarily unavailable caused by a successful cyber attack. Therefore, when probing Node i at time t , the reward function $c_{i,i}$ is defined as

$$\begin{aligned} c_{i,t} \triangleq & \left| Co^{\vartheta_t} \left(P_{i,j}^t, L_i^t, \Delta\delta; s_{i,i'} = 0 \right) \right. \\ & \left. - Co^{\vartheta_t} \left(P_{i,j}^t, L_i^t, \Delta\delta, s_{i,i'}; s_{i,i'} = 1 \right) \right|, \end{aligned} \quad (17)$$

where $Co^{\vartheta_t} \left(P_{i,j}^t, L_i^t, \Delta\delta; s_{i,i'} = 0 \right)$ is the operation cost of the DGS without cyber attacks, and $Co^{\vartheta_t} \left(P_{i,j}^t, L_i^t, \Delta\delta; s_{i,i'} = 1 \right)$ is the operation cost of the DGS given Node i is unavailable caused by the successful cyber attack on Node i . That is to say, the reward

¹ (URL: <https://www.elia.be/en/grid-data>)

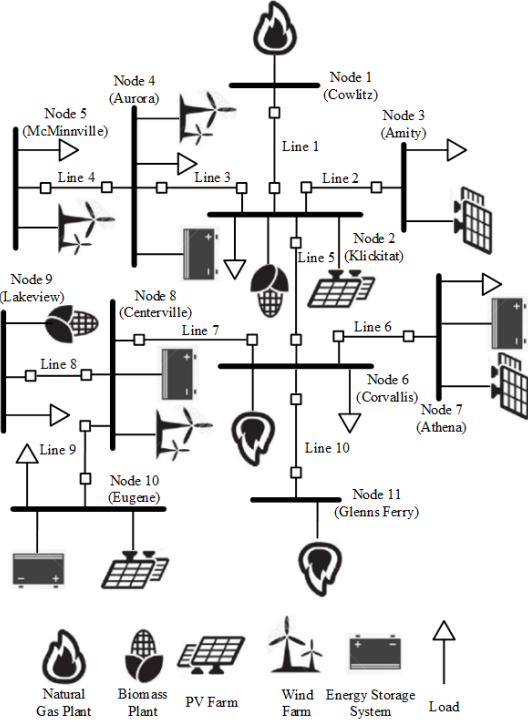


Fig. 1. The 11-node radial DGS originated from the IEEE 13 node test feeder.

$c_{i,t}$ can be calculated via solving the linear optimization problem, defined by (12) through (16) (with/without cyber attacks), where the input ϑ_t is drawn from the dataset of Elia Grid, Belgium.

4.2 Numerical analysis of cost sequences

In this section, numerical analysis is presented based on a real data set to verify that Assumption 1 holds in reality. Note that Assumption 1 implies a linear or sub-linear upper bound in terms of T on \mathcal{V}_T . Therefore, if \mathcal{V}_T has a linear or sub-linear growth rate in T , then Assumption 1 is supposed to hold by choosing a proper value for m . Thus, linear regression is performed on the sequence of \mathcal{V}_T against time T . The realistic grid data from the Elia is used, which provides data from the Belgian electricity market system. The underlying electricity network (a subgrid of Elia Grid) is shown in Figure 1. In the DGS, the dataset of Elia Grid, Belgium is recorded every 15 minutes, which means that the optimal power flow model will be performed and generate one attack cost $c_{i,t}$ every 15 minutes. The DGS under cyber attacks is investigated over one week, which indicates that a specific dataset with 672 successive observations of the attack cost is selected for illustration. m is chosen to be 4 and the data has been normalized such that $c_{i,t} \in [0, 1/4]$. The scatter plot is presented in Figure 2.

From Figure 2, a significant linear relation between \mathcal{V}_T and T can be observed. For further justification, the nu-

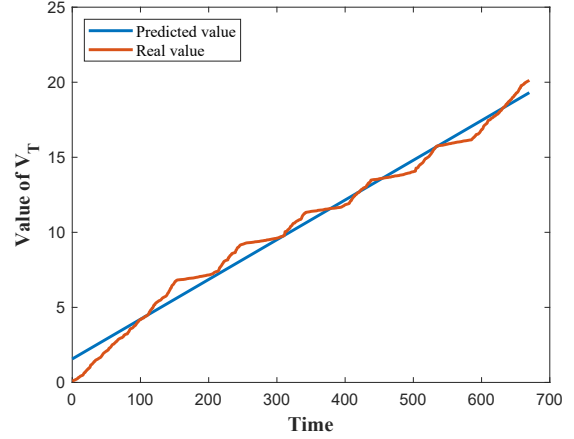


Fig. 2. Data scatter plot and linear prediction of \mathcal{V}_T .

merical results of the linear regression are presented in Table 1.

Table 1

Regression results.

| | Coefficients | Standard error | t Stat | P-value |
|-----------|--------------|----------------|--------|---------|
| Intercept | 1.560 | 0.04711 | 32.04 | < 0.01 |
| T | 0.02648 | 0.0001212 | 218.5 | 0 |

According to the results in Table 1, the coefficient for T is about 0.03, which means that Assumption 1 holds with T_0 chosen to be 9. As a significance test for the regression model, results of ANOVA for the regression model is shown in Table 2.

Table 2

ANOVA results.

| | df | SS | MS | F | Significance |
|------------|-----|---------|--------|-------|--------------|
| Regression | 1 | 17655 | 17655 | 47734 | ≤ 0.01 |
| Residual | 669 | 247.4 | 0.3699 | | |
| Total | 670 | 17902.4 | | | |

From the results in Table 2, the significance level is below 0.01, which means that the linear relation between \mathcal{V}_T and T for this data set is significant and the model is justified.

5 Numerical results

In this section, the performance of our Thompson-Hedge algorithm is illustrated through simulation studies. In Section 5.1, a comparative study is conducted between the performances of our Thompson-Hedge algorithm and the R.EXP3 algorithm which was recently proposed in Besbes et al. (2019). Subsequently, in Section 5.2, sensitivity analysis is conducted to investigate the influence of constraint \mathcal{V}_T on the cost sequence.

5.1 Comparative study

This section compares the performance of the proposed Thompson-Hedge algorithm with the R.EXP3 algorithm that was also designed for the same adversarial problem with constrained variation on cost sequence. Both algorithms use batch methods from the original EXP3 algorithm, *i.e.*, the time horizon is divided into small batches and the algorithm “restarts” at the beginning of each batch. Meanwhile, the R.EXP3 algorithm is also a randomized algorithm, retains a weight function for each node and updates the weights each time. Different from our Thompson-Hedge algorithm, in the R.EXP3 algorithm, if node $i \in \mathbb{N}$ is chosen at any time, the total reward $K_{i,t} \cdot c_{i,t}$ is considered as a whole adversarial reward. Since the attack number $k_{i',t}$ is unknown if $i \neq i'$, the R.EXP3 is supposed to ignore $c_{i',t}$ and treat the problem as a typical MNB model with bandit feedback.

The simulation is executed under $N = 10$ and $N = 20$. To initialize, the parameter $\theta = (\alpha_1, \beta_1, \dots, \alpha_N, \beta_N)$ is set for all $i \in \mathbb{N}$, and two large numbers for Q and L . The preset parameter θ is used to generate $\lambda = (\lambda_1, \dots, \lambda_N)$ for Q times in total. Under each generated λ , λ is used to generate the sequence $\{K_{i,t}\}_{t=1}^T$, and the adversarial cost sequence $\{c_{i,t}\}_{t=1}^T$ is artificially generated. Then, both algorithms are run based on $\{K_{i,t}\}_{t=1}^T$ and $\{c_{i,t}\}_{t=1}^T$, $\forall i \in \mathbb{N}$. Finally, the Bayesian sup regret in the two algorithms are calculated and compared. The simulation process is summarized using pseudo codes in Algorithm 3. Experience from the previous work Smith and Paté-Cornell (2018) and the real database from our numerical example are integrated to determine the parameter vector θ . In Smith and Paté-Cornell (2018), the prior distribution is set to be Gamma(2, 2); in our numerical example. It is estimated that $\lambda \approx 0.25$, which may correspond to the prior distribution Gamma(1, 4). Therefore, two values for θ are selected to implement the simulation: $\theta_1 : (\alpha_1, \beta_1) = \dots = (\alpha_N, \beta_N) = (2, 2)$ and $\theta_2 : (\alpha_1, \beta_1) = \dots = (\alpha_N, \beta_N) = (1, 4)$. According to the choices of θ , the truncation parameter m is set as 3, which ensures that the probability $Pr(K_{i,t} > m)$ is small.

Note that in the simulation, the empirical estimation of the sup regret and Bayesian sup regret functions is used in the two algorithms. Therefore, it is necessary to set the values of Q and L large enough to make the estimation with good precision. In the simulation, it is set as $Q = L = 10^5$. To simulate the real situations, the procedures of sketching the cost sequence are given as below:

Step 1: For all $i \in \mathbb{N}$, generate $c_{i,1}$ uniformly on $(0, 1/m)$ independently.

Step 2: For $t = 2, \dots, T$, generate $c_{i,t}$ independently

Algorithm 3: Simulation for comparison

Initialization: Preset parameter $\theta = (\alpha_1, \beta_1, \dots, \alpha_N, \beta_N)$; numbers of simulation trials Q, L ; time horizon T .

- 1: **for** $q = 1 : Q$ **do**
- 2: For all $i \in \mathbb{N}$, generate λ_i from $Gamma(\alpha_i, \beta_i)$.
- 3: **for** $l = 1 : L$ **do**
- 4: **for** $t = 1 : T$ **do**
- 5: **for** $i \in \mathbb{N}$ **do**
- 6: Generate $K_{i,t}$ from $Poisson(\lambda_i)$.
- 7: Generate $c_{i,t}$ artificially.
- 8: **end for**
- 9: **end for**
- 10: Run Thompson-Hedge algorithm and R.EXP3 algorithm independently based on the sequences $\{K_{i,t}\}_{t=1}^T$ and $\{c_{i,t}\}_{t=1}^T$, $\forall i \in \mathbb{N}$.
- 11: Calculate the regret function $R_l^{TH}(\lambda, \vec{c}_T, T)$ for Thompson-Hedge algorithm and $R_l^{R3}(\lambda, \vec{c}_T, T)$ for R.EXP3 algorithm.
- 12: **end for**
- 13: Calculate the sup regret $\tilde{R}_q^{TH} = \max_{l=1:L} R_l^{TH}$ for Thompson-Hedge algorithm and $\tilde{R}_q^{R3} = \max_{l=1:L} R_l^{R3}$ for R.EXP3 algorithm.
- 14: **end for**
- 15: Calculate the Bayesian sup regret

$$\mathcal{R}^{TH}(\mathcal{C}_T, T) = 1/Q \sum_{q=1}^Q \tilde{R}_q^{TH},$$

for Thompson-Hedge algorithm and,

$$\mathcal{R}^{R3}(\mathcal{C}_T, T) = 1/Q \sum_{q=1}^Q \tilde{R}_q^{R3},$$

for R.EXP3 algorithm.

from the uniform distribution on the overlapping interval between $(c_{i,t-1} - 1/100m, c_{i,t-1} + 1/100m)$ and $(0, 1/m)$.

By *Step 1*, simulate the random initial value of each cost sequence. Then, by *Step 2*, set $|c_{i,t} - c_{i,t-1}| \leq 1/50m$. It can be verified that any cost sequence generated by *Step 1* and *Step 2* satisfies Assumption 1 with $T_0 = 50$. The simulation results under $(\alpha_1, \beta_1) = \dots = (\alpha_N, \beta_N) = (2, 2)$ are given in Figures 3 and 4.

Using the same process, simulation is implemented for $(\alpha_1, \beta_1) = \dots = (\alpha_N, \beta_N) = (1, 4)$. The results are given in Figures 5 and 6.

In Figures 3-6, our Thompson-Hedge algorithm outperforms the existing R.EXP3 algorithm in terms of Bayesian sup regret. In particular, Thompson-Hedge

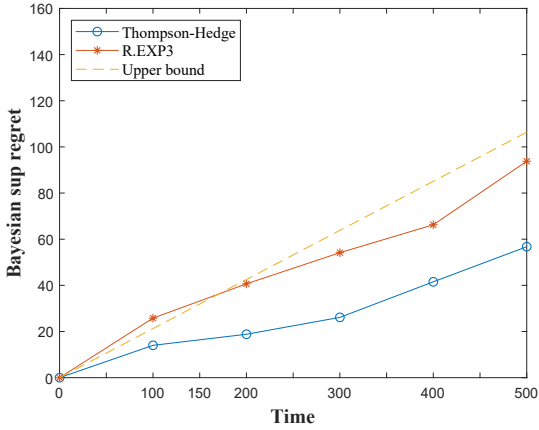


Fig. 3. Comparison of regrets ($\theta = \theta_1, N = 10$).

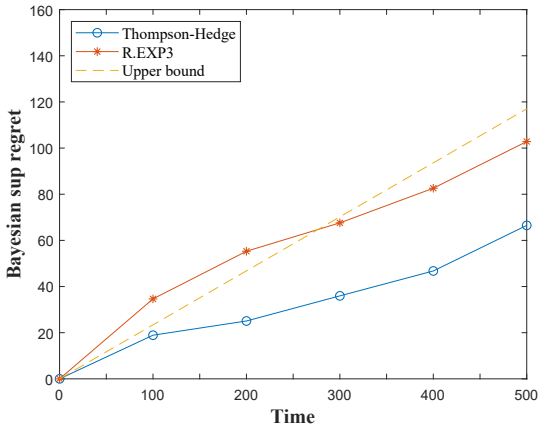


Fig. 4. Comparison of regrets ($\theta = \theta_1, N = 20$).

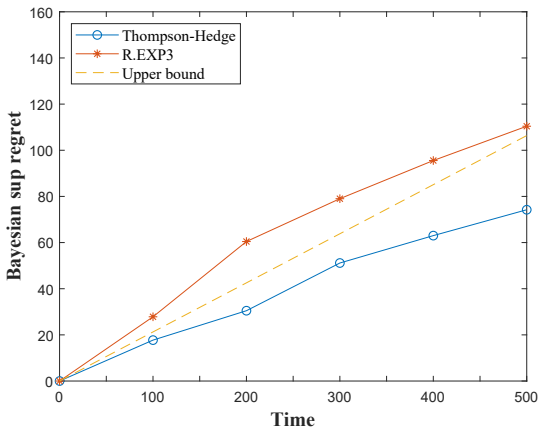


Fig. 5. Comparison of regrets ($\theta = \theta_2, N = 10$).

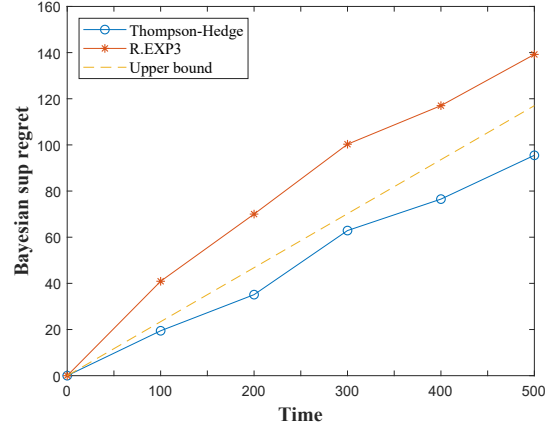


Fig. 6. Comparison of regrets ($\theta = \theta_2, N = 20$).

algorithm is less sensitive to the problem scale N by comparing the regret curves under $N = 10$ and $N = 20$, which is consistent with our discussion in Remark 1. Our algorithm has advantages over a typical algorithm designed for MNB with bandit feedback, such as the EXP3 or R.EXP3 algorithm. The usual convergence order of the regret function by a typical algorithm is $O(N \log N)^{1/3}$ for MNB with bandit feedback in terms of the problem scale N . Our Thompson-Hedge algorithm feeds a set of sampled parameters to Hedge algorithm. In the proof of Theorem 1, it is shown that using the sampled parameters is “as good as” using the true parameters under the Bayesian framework. Moreover, if the true parameters are known, then Hedge algorithm can solve the problem, which has a convergence order of $O(\log N)$. Therefore, our Thompson-Hedge algorithm is supposed to retain a convergence rate of $O(\log N)$ for the Bayesian sup regret in the special case.

Meanwhile, upper bound in Theorem 1 is also drawn in Figures 3-6. It is worth noting that the upper bound in Theorem 1 holds uniformly on all possible values of θ . Hence, it is concluded that when θ varies, the deviation of the regret from the upper bound also varies. In particular, as shown in the figures, the difference between the upper bound and the regret is comparatively large under $\theta = \theta_1, N = 20$ while small or moderate under other three parameters. However, it can be observed from the figures that our upper bound is obviously sharper than that for the R.EXP3 algorithm in the sense that our upper bound is below the regret under R.EXP3 at some time points in all the four figures.

5.2 Sensitivity to the variation

This section presents the sensitivity analysis of our Thompson-Hedge algorithm to the variation constraint \mathcal{V}_T . In reality, according to the specific environment and workload under which the network functions, the adversary may generate the cost sequence subject to various

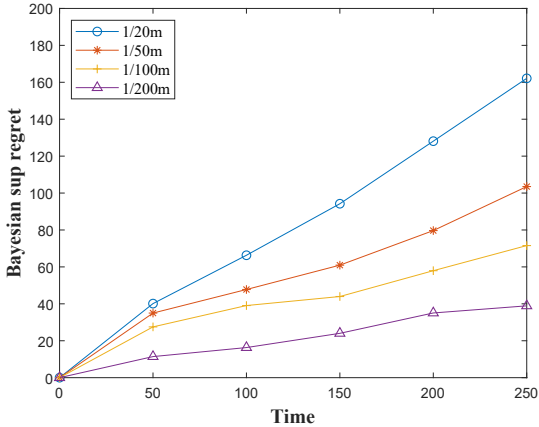


Fig. 7. Regrets under different variations ($\theta = \theta_1, N = 20$).

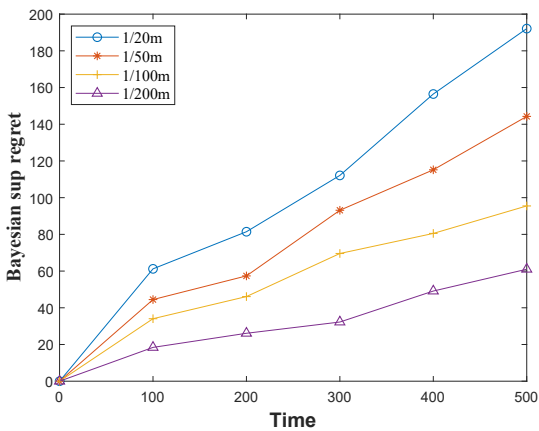


Fig. 8. Regrets under different variations ($\theta = \theta_2, N = 20$).

rules. In our model, the variation constraint \mathcal{V}_T is used to characterize the cost sequence. Thus, numerical results for the proposed Thompson-Hedge algorithm are presented under four levels of variation. In particular, the same steps given in Section 5.1 are employed, but 4 levels are selected for the variation scale in *Step 2*: $1/20m$, $1/50m$, $1/100m$ and $1/200m$. Other parameters are set to be identical as those in Section 5.1. The simulation results are illustrated under both θ_1 and θ_2 as shown in Figures 7 and 8.

The results in Figures 7 and 8 illustrate the different performances of our Thompson-Hedge algorithm under different variation scales. The Bayesian regret monotonically increases with the variation. This is because larger variation leads to more uncertainties of the cost sequence and makes the experience from historical observations unreliable. Thus, as an adaptive online learning algorithm, the Bayesian sup regret in our Thompson-Hedge algorithm may increase and the performance may have fluctuations. However, it should be noted that the incremental difference between the regret curves under $1/100m$ and $1/200m$ are not so significant compared to

that between the curves under $1/50m$ and $1/20m$. This implies that when the variation is within $\mathcal{V}_T \leq T/50m$, the regret function is less sensitive to the variation than the situation that \mathcal{V}_T is around $1/10m$. In the numerical example, the real data set (from Elia Grid, Belgium) is examined and it appears that the corresponding \mathcal{V}_T is around $T/100m$. Therefore, it can be concluded that the sensitivity of our Thompson-Hedge algorithm to the variation may not be high in practice.

6 Concluding remarks

This study investigates the sequential control problem in modern energy and power systems with smart grids. The existing work is extended by introducing an adversarial cost sequence with a variation constraint. A Bayesian MNB model is constructed to cope with the problem and an online learning algorithm named Thompson-Hedge algorithm is proposed to retain a converging regret function. In addition, it is proved that the convergence rate of the regret in the proposed algorithm is superior to an existing algorithm that can be used for this problem.

In developing our algorithm, a basic idea is to firstly obtain sampled parameters from the posterior distribution when the reinforcement learning model is a partly parameterized Bayesian model. Subsequently, the sampled parameters are used instead of the true parameters in the model. It is worth pointing out that although specific models are used in this work, the paradigm of our algorithm framework can also be used for other classical models.

From a conceptual perspective, future research will explore the applications of the proposed algorithms to more situations in reality. For example, the proposed algorithm can be used to mitigate the impact of failure in tracking goods, or obtain suppliers to exchange inventory information due to successful cyber attacks. Other potential applications can be found in the domains of multi-robot systems (Liu et al., 2017), Internet of Things (IOT) (Perera et al., 2015), *e.g.*, traffic management, water distribution, waste management, online manufacturing and smart supply chain. It is also worth investigating the impact of other types of cyber attacks such as the botnet command and control, data exfiltration, data tampering, data destruction or even physical, destruction via alternation of critical software/data (Pillitteri and Brewer, 2014) on the smart grid by properly modeling their influences on the Optimal Power Flow model according to their attack mechanisms. For example, the replay attacks maliciously repeat DR messages or dispatch commands, and the false data injection attacks purposely alter the integrity of a smart grid by compromising a subset of transmitted data packages and sending out inaccurate DR messages or dispatch commands.

Appendix A Proof of Lemma 1

In this section, detailed proof of Lemma 1 is provided, which is used in the proof of Theorem 1. Before reaching the proof of Lemma 1, a preliminary result is presented in the following lemma Slivkins et al. (2019).

Lemma 3 [Slivkins et al. (2019), Theorem 5.16] *If all per-time costs are in $[0, 1]$ and ε is chosen to be $(1 - \sqrt{\ln N/2T})^{-1}$, then the Hedge algorithm satisfies*

$$\begin{aligned} & \max_{i \in \mathbb{N}} \left\{ \sum_{t=\tau_{b-1}}^{\tau_b-1} K_{i,t} \cdot c_{i,t} \right\} \\ & - \mathbf{E}^H \left(\sum_{t=\tau_{b-1}}^{\tau_b-1} K_{i_t,t} \cdot c_{i_t,t} \mid \lambda, \vec{c}_T \right) \\ & \leq 2\sqrt{2} \cdot \sqrt{\Delta \ln N}. \end{aligned}$$

Note that the original theorem in Slivkins et al. (2019) concerns a reward-based problem and the parameter ε is chosen to be $\sqrt{\ln N/2T}$. Actually, by using the transformation of $cost = 1 - reward$ between the per-time cost and reward, the result can be transferred in the original theorem into the conclusion in Lemma 3. Detailed proof of Lemma 1 is given as follows.

Proof of Lemma 1. The proof is divided into four steps. In the first step, the best single action policy is used to decompose the regret function into two parts. In the second and third steps, upper bounds on the two parts from the first step are constructed, respectively. Finally, the upper bound on the original regret function is presented. Denote the sub-sequence of times when Hedge algorithm restarts as $1 = \tau_0 \leq \tau_1 \leq \dots \leq \tau_{\lceil T/\Delta \rceil}$, which implies that $\tau_{b+1} - \tau_b = \Delta$ for all $b = 1, \dots, \lceil T/\Delta \rceil - 1$ and $\tau_{\lceil T/\Delta \rceil} - \tau_{\lceil T/\Delta \rceil - 1} \leq \Delta$. For all $b = 1, \dots, \lceil T/\Delta \rceil$, note that the set of times $\{\tau_{b-1}, \dots, \tau_b - 1\}$ exactly forms the b th batch.

Step 1 Regret decomposition.

For each $b = 1, \dots, \lceil T/\Delta \rceil$, the best single node I_b^* in b th batch is defined as $I_b^* \triangleq \arg \max_{i \in \mathbb{N}} \left\{ \sum_{t=\tau_{b-1}}^{\tau_b-1} \mu_i \cdot c_{i,t} \right\}$. A specific sequence $\{c_{i,t}\}_{t=1}^T \in \mathbb{C}_T$ is fixed and the regret

function within b th batch is decomposed as

$$\begin{aligned} & \sup_{\vec{c}_T \in \mathbb{C}_T} \sum_{t=\tau_{b-1}}^{\tau_b-1} \mu_{i_t}^* \cdot c_{i_t}^*, t - \mathbf{E}^H \left(\sum_{t=\tau_{b-1}}^{\tau_b-1} \mu_{i_t} \cdot c_{i_t,t} \mid \lambda, \vec{c}_T \right) \\ & \leq \sup_{\vec{c}_T \in \mathbb{C}_T} \left\{ \sum_{t=\tau_{b-1}}^{\tau_b-1} \mu_{i_t}^* \cdot c_{i_t}^*, t - \sum_{t=\tau_{b-1}}^{\tau_b-1} \mu_{I_b^*} \cdot c_{I_b^*}^*, t \right\} \\ & + \sup_{\vec{c}_T \in \mathbb{C}_T} \left\{ \sum_{t=\tau_{b-1}}^{\tau_b-1} \mu_{I_b^*} \cdot c_{I_b^*}^*, t \right. \\ & \left. - \mathbf{E}^H \left(\sum_{t=\tau_{b-1}}^{\tau_b-1} \mu_{i_t} \cdot c_{i_t,t} \mid \lambda, \vec{c}_T \right) \right\}. \end{aligned} \tag{A.1}$$

The two terms in the right hand side of (A.1) are defined as

$$R_{b,1} \triangleq \sup_{\vec{c}_T \in \mathbb{C}_T} \left\{ \sum_{t=\tau_{b-1}}^{\tau_b-1} \mu_{i_t}^* \cdot c_{i_t}^*, t - \sum_{t=\tau_{b-1}}^{\tau_b-1} \mu_{I_b^*} \cdot c_{I_b^*}^*, t \right\},$$

and

$$\begin{aligned} R_{b,2} \triangleq & \sup_{\vec{c}_T \in \mathbb{C}_T} \left\{ \sum_{t=\tau_{b-1}}^{\tau_b-1} \mu_{I_b^*} \cdot c_{I_b^*}^*, t \right. \\ & \left. - \mathbf{E}^H \left(\sum_{t=\tau_{b-1}}^{\tau_b-1} \mu_{i_t} \cdot c_{i_t,t} \mid \lambda, \vec{c}_T \right) \right\}. \end{aligned}$$

Step 2 Upper bounding $R_{b,1}$.

Let $\mathcal{V}_T^b \triangleq \sum_{t=\tau_{b-1}}^{\tau_b-2} |c_{i,t+1} - c_{i,t}|$ be the total variation of $c_{i,t}$ in b th batch. For all $\vec{c}_T \in \mathbb{C}_T$, the following relation holds

$$\max_{\tau_{b-1} \leq t \leq \tau_b - 1} \left\{ \mu_{i_t}^* \cdot c_{i_t}^*, t - \mu_{I_b^*} \cdot c_{I_b^*}^*, t \right\} \leq 2m\mathcal{V}_T^b. \tag{A.2}$$

By contradiction, if (A.2) does not hold, then there is at least one time $\tau_{b-1} \leq t_0 \leq \tau_b - 1$ such that $\mu_{i_{t_0}}^* \cdot c_{i_{t_0}}^*, t_0 - \mu_{I_b^*} \cdot c_{I_b^*}^*, t_0 \geq 2m\mathcal{V}_T^b$. Let $i_{t_0}^* = i_0$. Since $\mu_i \leq m, \forall i \in \mathbb{N}$, for all $\tau_{b-1} \leq t \leq \tau_b - 1$, it follows

$$\begin{aligned} \mu_{i_0} \cdot c_{i_0,t} & > \mu_{i_0} (c_{i_0,t_0} - \mathcal{V}_T^b) > \mu_{i_0} \cdot c_{i_0,t_0} - m\mathcal{V}_T^b \\ & > \mu_{I_b^*} \cdot c_{I_b^*}^*, t_0 + m\mathcal{V}_T^b > \mu_{I_b^*} \cdot c_{I_b^*}^*, t_0 + \mathcal{V}_T^b \\ & > \mu_{I_b^*} \cdot c_{I_b^*}^*, t. \end{aligned}$$

However, this contradicts with the fact that I_b^* is the optimal single node of b th batch. Thus, inequality (A.2)

holds. Therefore, it can be obtained that

$$\begin{aligned} R_{b,1} &= \sup_{\vec{c}_T \in \mathbb{C}_T} \sum_{t=\tau_{b-1}}^{\tau_b-1} \left(\mu_{i_t}^* \cdot c_{i_t}^*, t - \mu_{I_b}^* \cdot c_{I_b}^*, t \right) \\ &\leq \Delta \sup_{\vec{c}_T \in \mathbb{C}_T} \max_{\tau_{b-1} \leq t \leq \tau_b-1} \left\{ \mu_{i_t}^* \cdot c_{i_t}^*, t - \mu_{I_b}^* \cdot c_{I_b}^*, t \right\} \\ &\leq 2\Delta m \mathcal{V}_T^b. \end{aligned}$$

Step 3 Upper bounding $R_{b,2}$.

Note that

$$\begin{aligned} \sum_{t=\tau_{b-1}}^{\tau_b-1} \mu_{I_b}^* \cdot c_{I_b}^*, t &= \sum_{t=\tau_{b-1}}^{\tau_b-1} \mathbf{E} \left(K_{I_b}^*, t \cdot c_{I_b}^*, t \right) \\ &= \max_{i \in \mathbb{N}} \left\{ \sum_{t=\tau_{b-1}}^{\tau_b-1} \mathbf{E} (K_{i,t} \cdot c_{i,t}) \right\} \\ &\leq \mathbf{E} \left(\max_{i \in \mathbb{N}} \left\{ \sum_{t=\tau_{b-1}}^{\tau_b-1} K_{i,t} \cdot c_{i,t} \right\} \right). \end{aligned}$$

Therefore, it holds that

$$\begin{aligned} R_{b,2} &\leq \sup_{\vec{c}_T \in \mathbb{C}_T} \left\{ \mathbf{E} \left[\left(\max_{i \in \mathbb{N}} \left\{ \sum_{t=\tau_{b-1}}^{\tau_b-1} K_{i,t} \cdot c_{i,t} \right\} \right) \right. \right. \\ &\quad \left. \left. - \mathbf{E}^H \left(\sum_{t=\tau_{b-1}}^{\tau_b-1} \mu_{i_t} \cdot c_{i_t}, t \mid \lambda, \vec{c}_T \right) \right] \right\} \\ &\leq \sup_{\vec{c}_T \in \mathbb{C}_T} \left\{ \mathbf{E} \left[\left(\max_{i \in \mathbb{N}} \left\{ \sum_{t=\tau_{b-1}}^{\tau_b-1} K_{i,t} \cdot c_{i,t} \right\} \right) \right. \right. \\ &\quad \left. \left. - \mathbf{E}^H \left(\sum_{t=\tau_{b-1}}^{\tau_b-1} k_{i_t}, t \cdot c_{i_t}, t \mid \lambda, \vec{c}_T \right) \right] \right\}. \end{aligned}$$

According to Lemma 1 and noting that the relation in Lemma 1 holds for arbitrary \vec{c}_T satisfying $K_{i,t} \cdot c_{i,t} \leq 1$, $\forall i, t$, it follows

$$R_{b,2} \leq 2\sqrt{2} \cdot \sqrt{\Delta \ln N}.$$

Step 4 Upper bounding the sup regret.

In the final step, discussions in the previous two steps and the upper bound on the sup regret by Hedge algo-

rithm are summarized. Note that

$$\begin{aligned} \sup_{\vec{c}_T \in \mathbb{C}_T} R^H(\lambda, \vec{c}_T, T) &\leq \sum_{b=1}^{\lceil T/\Delta \rceil} (R_{b,1} + R_{b,2}) \\ &\leq \sum_{b=1}^{\lceil T/\Delta \rceil} \left(2\Delta m \mathcal{V}_T^b + 2\sqrt{2} \cdot \sqrt{\Delta \ln N} \right) \\ &\leq 2\Delta m \mathcal{V}_T + \left(\frac{T}{\Delta} + 1 \right) 2\sqrt{2} \cdot \sqrt{\Delta \ln N}, \quad (\text{A.3}) \end{aligned}$$

where the third inequality follows from the relation $\sum_{b=1}^{\lceil T/\Delta \rceil} \mathcal{V}_T^b = \mathcal{V}_T$.

Embedding $\Delta = \left\lceil (\ln N)^{1/3} (T/m\mathcal{V}_T)^{2/3} \right\rceil$ into (A.3), it follows

$$\begin{aligned} \sup_{\vec{c}_T \in \mathbb{C}_T} R^\pi(\lambda, \vec{c}_T, T) &\leq 2 \left[(\ln N)^{1/3} \left(\frac{T}{m\mathcal{V}_T} \right)^{2/3} + 1 \right] m\mathcal{V}_T \\ &\quad + 2\sqrt{2} \cdot T \cdot \sqrt{\ln N (\ln N)^{-1/3} \left(\frac{T}{m\mathcal{V}_T} \right)^{-2/3}} \\ &\quad + 2\sqrt{2} \sqrt{\ln N \left[(\ln N)^{1/3} \left(\frac{T}{m\mathcal{V}_T} \right)^{2/3} + 1 \right]}. \end{aligned}$$

Since $N > 2$, Assumption 1 leads to $(\ln N)^{1/3} (T/m\mathcal{V}_T)^{2/3} > 1$ and $\lceil \ln N/T(m\mathcal{V}_T)^2 \rceil < 1$. It follows

$$\begin{aligned} \sup_{\vec{c}_T \in \mathbb{C}_T} R^\pi(\lambda, \vec{c}_T, T) &\leq (4 + 2\sqrt{2}) (m\mathcal{V}_T \ln N)^{1/3} (T)^{2/3} \\ &\quad + 4 (\ln N)^{2/3} \left(\frac{T}{m\mathcal{V}_T} \right)^{1/3} \\ &\leq (4 + 2\sqrt{2}) (m\mathcal{V}_T \ln N)^{1/3} (T)^{2/3} \\ &\quad + 4 (m\mathcal{V}_T \ln N)^{1/3} (T)^{2/3} \left[\frac{\ln N}{T (m\mathcal{V}_T)^2} \right]^{1/3} \\ &\leq (4 + 2\sqrt{2}) (m\mathcal{V}_T \ln N)^{1/3} (T)^{2/3} \\ &\quad + 4 (m\mathcal{V}_T \ln N)^{1/3} (T)^{2/3} \\ &\leq (8 + 2\sqrt{2}) (m\mathcal{V}_T \ln N)^{1/3} (T)^{2/3}, \end{aligned}$$

which concludes the proof.

References

Abiri-Jahromi, A., Kemmeugne, A., Kundur, D. and Haddadi, A. (2020), ‘Cyber-physical attacks targeting communication-assisted protection schemes’,

- IEEE Transactions on Power Systems* **35**(1), 440–450.
- Amin, S., Schwartz, G. A. and Sastry, S. S. (2013), ‘Security of interdependent and identical networked control systems’, *Automatica* **49**(1), 186–192.
- Auer, P., Cesa-Bianchi, N., Freund, Y. and Schapire, R. E. (2002), ‘The nonstochastic multiarmed bandit problem’, *SIAM journal on computing* **32**(1), 48–77.
- Besbes, O., Gur, Y. and Zeevi, A. (2019), ‘Optimal exploration–exploitation in a multi-armed bandit problem with non-stationary rewards’, *Stochastic Systems* **9**(4), 319–337.
- Che, L., Liu, X., Shuai, Z., Li, Z. and Wen, Y. (2018), ‘Cyber cascades screening considering the impacts of false data injection attacks’, *IEEE Transactions on Power Systems* **33**(6), 6545–6556.
- Even-Dar, E., Kakade, S. M. and Mansour, Y. (2009), ‘Online markov decision processes’, *Mathematics of Operations Research* **34**(3), 726–736.
- Freund, Y. and Schapire, R. E. (1999), ‘Adaptive game playing using multiplicative weights’, *Games and Economic Behavior* **29**(1-2), 79–103.
- Gallo, A. J., Turan, M. S., Boem, F., Parisini, T. and Ferrari-Trecate, G. (2020), ‘A distributed cyber-attack detection scheme with application to dc microgrids’, *IEEE Transactions on Automatic Control* DOI: [10.1109/TAC.2020.2982577](https://doi.org/10.1109/TAC.2020.2982577).
- Heyman, D. and Sobel, M. (2004), ‘Superposition of renewal processes’, *Stochastic Models in Operations Research: Stochastic Processes and Operating Characteristics* p. 158.
- Khalili, M., Zhang, X., Cao, Y., Polycarpou, M. M. and Parisini, T. (2020), ‘Distributed fault-tolerant control of multiagent systems: An adaptive learning approach’, *IEEE Transactions on Neural Networks and Learning Systems* **31**(2), 420–432.
- Konstantelos, I., Giannelos, S. and Strbac, G. (2016), ‘Strategic valuation of smart grid technology options in distribution networks’, *IEEE Transactions on Power Systems* **32**(2), 1293–1303.
- Li, J.-A., Dong, D., Wei, Z., Liu, Y., Pan, Y., Nori, F. and Zhang, X. (2020), ‘Quantum reinforcement learning during human decision-making’, *Nature Human Behaviour* **4**(3), 294–307.
- Littman, M. L. (2015), ‘Reinforcement learning improves behaviour from evaluative feedback’, *Nature* **521**(7553), 445–451.
- Liu, Z., Wang, L., Wang, J., Dong, D. and Hu, X. (2017), ‘Distributed sampled-data control of nonholonomic multi-robot systems with proximity networks’, *Automatica* **77**, 170–179.
- Mo, H. and Sansavini, G. (2017), ‘Dynamic defense resource allocation for minimizing unsupplied demand in cyber-physical systems against uncertain attacks’, *IEEE Transactions on Reliability* **66**(4), 1253–1265.
- Mo, H. and Sansavini, G. (2019), ‘Impact of aging and performance degradation on the operational costs of distributed generation systems’, *Renewable energy* **143**, 426–439.
- Osband, I., Russo, D. and Van Roy, B. (2013), (more) efficient reinforcement learning via posterior sampling, in ‘Advances in Neural Information Processing Systems’, pp. 3003–3011.
- Patsakis, G., Rajan, D., Aravena, I., Rios, J. and Oren, S. (2018), ‘Optimal black start allocation for power system restoration’, *IEEE Transactions on Power Systems* **33**(6), 6766–6776.
- Perera, C., Liu, C. H. and Jayawardena, S. (2015), ‘The emerging internet of things marketplace from an industrial perspective: A survey’, *IEEE Transactions on Emerging Topics in Computing* **3**(4), 585–598.
- Pillitteri, V. Y. and Brewer, T. L. (2014), Guidelines for smart grid cybersecurity, Technical report.
- Pogaku, N., Prodanovic, M. and Green, T. C. (2007), ‘Modeling, analysis and testing of autonomous operation of an inverter-based microgrid’, *IEEE Transactions on Power Electronics* **22**(2), 613–625.
- Rana, M. M., Li, L. and Su, S. W. (2017), ‘Cyber attack protection and control of microgrids’, *IEEE/CAA Journal of Automatica Sinica* **5**(2), 602–609.
- Rana, M. M., Xiang, W. and Wang, E. (2018), ‘Smart grid state estimation and stabilisation’, *International Journal of Electrical Power & Energy Systems* **102**, 152–159.
- Russo, D. and Van Roy, B. (2014), ‘Learning to optimize via posterior sampling’, *Mathematics of Operations Research* **39**(4), 1221–1243.
- Sahraei-Ardakani, M. and Hedman, K. W. (2016), ‘Computationally efficient adjustment of facts set points in dc optimal power flow with shift factor structure’, *IEEE Transactions on Power Systems* **32**(3), 1733–1740.
- Slivkins, A. et al. (2019), ‘Introduction to multi-armed bandits’, *Foundations and Trends® in Machine Learning* **12**(1-2), 1–286.
- Smith, M. D. and Paté-Cornell, M. E. (2018), ‘Cyber risk analysis for a smart grid: how smart is smart enough? a multiarmed bandit approach to cyber security investment’, *IEEE Transactions on Engineering Management* **65**(3), 434–447.
- Sutton, R. S. and Barto, A. G. (2018), *Reinforcement learning: An introduction*, MIT press.
- Todescato, M., Bof, N., Cavararo, G., Carli, R. and Schenato, L. (2020), ‘Partition-based multi-agent optimization in the presence of lossy and asynchronous communication’, *Automatica* **111**, 108648.
- Yu, K., Ai, Q., Wang, S., Ni, J. and Lv, T. (2015), ‘Analysis and optimization of droop controller for microgrid system based on small-signal dynamic model’, *IEEE Transactions on Smart Grid* **7**(2), 695–705.
- Zhang, X. and Papachristodoulou, A. (2015), ‘A real-time control framework for smart power networks: Design methodology and stability’, *Automatica* **58**, 43–50.
- Zhu, M. and Martínez, S. (2013), ‘On distributed constrained formation control in operator–vehicle adversarial networks’, *Automatica* **49**(12), 3571–3582.