Multiple-stopping time Sequential Detection for Energy Efficient Mining in Blockchain-Enabled IoT

Anurag Gupta, and Vikram Krishnamurthy, *Fellow, IEEE*

***Abstract*—What are the optimal times for an Internet of Things (IoT) device to act as a blockchain miner? The aim is to minimize the energy consumed by low-power IoT devices that log their data into a secure (tamper-proof) distributed ledger. We formulate a multiple stopping time Bayesian sequential detection problem to address energy-efficient blockchain mining for IoT devices. The objective is to identify L optimal stops for mining, thereby maximizing the probability of successfully adding a block to the blockchain; we also present a model to optimize the number of stops (mining instants). The formulation is equivalent to a multiple stopping time POMDP. Since POMDPs are in general computationally intractable to solve, we show mathematically using submodularity arguments that the optimal mining policy has a useful structure: 1) it is monotone in belief space, and 2) it exhibits a threshold structure, which divides the belief space into two connected sets. Exploiting the structural results, we formulate a computationally-efficient linear mining policy for the blockchain-enabled IoT device. We present a policy gradient technique to optimize the parameters of the linear mining policy. Finally, we use synthetic and real Bitcoin datasets to study the performance of our proposed mining policy. We demonstrate the energy efficiency achieved by the optimal linear mining policy in contrast to other heuristic strategies.**

***Index Terms*—Internet of Things (IoT), blockchain, optimal mining, partially observed Markov decision process (POMDP), multiple stopping time, maximum likelihood estimator (MLE), monotone likelihood ratio (MLR), total positivity of order 2 (TP2), value iteration, stochastic gradient descent, Bellman equation, submodularity.**

I. Introduction

Blockchain is a decentralized distributed ledger technology [1]. Each block in the chain contains a set of transactions and a cryptographic hash of the previous block. This creates a chain of blocks that are secure: it is difficult for a single entity to take control of the network or to alter past transactions. An important element of blockchain is Proof of Work (PoW) [2]. PoW is a consensus algorithm used in blockchain to add new blocks to the chain; this requires miners to solve a cryptographic puzzle. The first miner to solve the puzzle is rewarded monetarily. However, PoW in blockchain requires a large amount of computational power and leads to high energy consumption. This is detrimental to using blockchain in an IoT application.

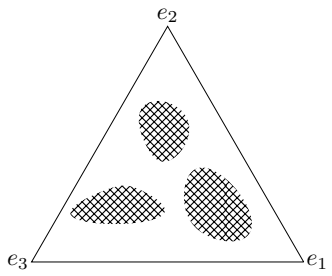
We focus on blockchain-enabled IoT applications wherein IoT devices are resource-constrained [3], e.g. wireless sensor networks. The combination of blockchain and IoT can create a secure and decentralized network of devices, enabling efficient and transparent data sharing and transactions [4]. For example, in a sensor network, blockchain can provide an immutable and tamper-proof[1] data storage platform for storing sensor readings; the decentralized storage of data also makes it immune to a single point of failure [6]. An IoT network consists of heterogeneous devices: some have low power and low energy requirements, like Raspberry Pi, while others have high processing power and energy requirements, like PCs and servers. Low-power IoT devices are typically deployed for data collection, whereas high-power devices are used for time-critical applications. Irrespective of their processing power, the devices are capable of mining in a blockchain. For a typical IoT application [3], CPU usage without mining is around 3%-6%. However, mining in blockchain increases the CPU usage of IoT devices to 30%-50%; this is roughly a ten times increase in power consumption. This is detrimental for IoT devices as they have limited energy resources. Additionally, blockchain mining has an adverse impact on the environment due to its energy consumption [7]. This motivates the study of energy-efficient mining strategy in blockchain for an IoT device: the IoT device wants to optimize its mining time instants so as to maximize its probability of adding a new block in the blockchain. By doing so, the IoT device prevents the waste of energy on mining when there is high competition to add a new block to the blockchain.

Main Idea. Multiple Stopping Time POMDP

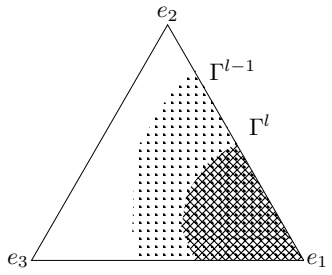
The problem we address is: *What are the optimal times for an IoT device to act as a blockchain miner?* The aim is to minimize the energy consumed by low-power IoT devices that register their data into a secure (tamper-proof) distributed ledger. In IoT applications, IoT devices have to log their data in the blockchain multiple times, depending on the data rate. Moreover, energy constraints may limit the lifespan of IoT devices in wireless sensor networks. These factors motivate the study of multiple mining time selections for IoT devices. We formulate a multiple stopping time Bayesian sequential detection problem as a partially observed Markov decsion process (POMDP) [8] to address energy-efficient blockchain mining for IoT devices. The objective is to identify L optimal stops for mining, thereby maximizing the probability of successfully adding a block to the blockchain. We assume that the dynamics of the POMDP are not affected by the mining activity of the IoT device. This assumption is reasonable since the computing power of the IoT device is too small to affect the overall rate of new blocks in the blockchain. Multiple-stopping time POMDP has not been studied much in the literature. Compared to the single-stopping time problem, the multiple-stopping

Anurag Gupta and Vikram Krishnamurthy are with the School of Electrical & Computer Engineering, Cornell University, Ithaca NY, 14853, USA. (e-mail: ag2589@cornell.edu; vikramk@cornell.edu).

[1]Sensor networks are used in the field of agriculture to log the farming practices like the use of pesticides. An immutable and tamper-proof database will improve trust between farmers, manufacturers, retailers and consumers [5].



(a) Mining policy with no special structure.



(b) Mining policy with a threshold and nested structure.

Fig. 1: Visual illustration of the structure of a mining policy. The triangle represents the two-dimensional belief space for the POMDP. The shaded regions indicate the belief state where it is optimal to mine in the blockchain. In general, the optimal mining states are arbitrary, as in (a) and computing the shaded regions is intractable. The main contribution of this paper is to propose sufficient conditions for submodularity so that the optimal policy has the threshold and nested structure as in (b). Here, Γ_l denotes the threshold for the l^{th} mining instant (see Sec.III for details). This structure is then exploited to develop policy gradient algorithms.

time problem is a more complex generalization because using the single-stopping policy repeatedly results in a suboptimal solution.

Using the optimal policy of the resulting POMDP ensures that the IoT device does not waste energy on mining when there are several miners competing to mine a new block. We also formulate an optimization problem to optimize the number of mining instants for a blockchain-enabled IoT device. This is important in IoT applications such as wireless sensor networks, as each IoT device senses data at a different rate. Hence, the amount of data that needs to be logged in the blockchain varies with IoT devices.

Submodular Structure of the Energy-Efficient Mining Problem

In general, solving a multiple-stopping time POMDP is P-SPACE hard [9] as the optimal mining policy may not have a special structure as shown (Fig. 1a). Hence, often POMDPs are solved via heuristics. In this paper, we show mathematically that the optimal policy for the multiple stopping time POMDP has a special structure: 1) optimal mining policy is monotone in belief space, 2) optimal mining policy has a threshold structure, thereby partitioning the belief space into two connected sets (Fig.1b). These structural results are proved via submodularity arguments on the stochastic dynamic programming equation of the POMDP. The important practical consequence is that this structure can be exploited to design policies which are linear in belief state and efficiently implementable on IoT devices. The structure also facilitates the development of computationally-efficient policy gradient algorithms that can be implemented on IoT devices. Firstly, we establish both the necessary and sufficient conditions that the parameters of the linear mining policy must meet in order to satisfy the structural results. Subsequently, we convert these parameters into spherical coordinates, enabling us to formulate an unconstrained optimization problem for optimizing the parameters.

Related works

The benefits of integrating blockchain and IoT is discussed in [6]. [4] describes an architecture for integrating IoT and blockchain, and [10] proposes a medium access control (MAC) protocol for IoT-blockchain network setup. [3] conducts simulations, and [11] uses a prototype implementation to study the overall system performance while integrating blockchain with IoT.

Related to the modeling of the blockchain system, [12] and [13] employ a Markov process model to study performance and network security in a distributed ledger. The evolution of cryptocurrency as a Hidden Markov Model is explored in [12], [14].

Regarding optimal mining strategies in blockchain, [15] formulates the mining problem with resource cost as a dynamic game over an infinite horizon and shows that it is optimal to mine together. [16], [17] uses reinforcement learning to optimize selfish mining strategy in the blockchain. [18] utilizes a game theoretic approach to study Nash equilibrium in blockchain mining when miners can hide their newly mined nodes. [19] formulates various selfish mining strategy in blockchain as a Markov decision process and solves it to obtain a lower bound on their performance. [20] analyzes the optimal mining time for pooled mining reward systems. The author argues that the optimal mining time depends on the miners' incentives and the network's transaction volume.

[21] surveys the problem of energy overhead in integrating IoT and blockchain both from computational and communication viewpoints. [22] presents a clustering method to improve energy efficiency for blockchain mining in an IoT application.

Our energy-efficient mining problem for a blockchain-enabled IoT device utilizes tools from [23], which provides useful structural results on the optimal policy for multiple stopping time POMDP. The structural result allows solving the POMDP for large state space. Another approach to reducing the complexity of MDP with large state space is presented in [24]: the authors approximate the value function by projecting it into a lower-dimensional subspace. Also, one can use stochastic dominance to compute bounds on HMM filter with reduced computational complexity [25]. Multiple stopping time POMDP has been used for targeting ads, intrusion prevention [26], active sensing [27], sensor scheduling [28] and detecting line outage in power systems [29].

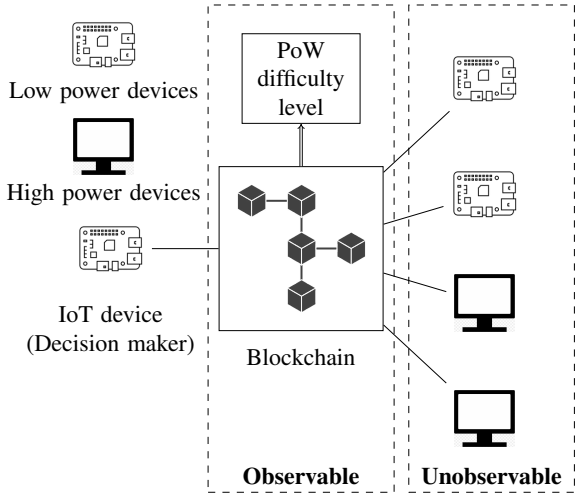


Fig. 2: Schematic of the energy-efficient mining problem for blockchain-enabled IoT applications with heterogeneous devices. The IoT device (decision maker) wants to decide when to mine in the blockchain to maximize its probability of adding a new block. The IoT device cannot observe the computing power of other miners, nor can it observe whether they are mining or not. It can only observe the PoW difficulty level of the blockchain, a noisy observation of the other miners' actions. Hence, the IoT device has to decide its action based only on the available information to maximize its probability of adding a new block to the blockchain.

To the best of our knowledge, the multiple stopping time POMDP approach to study energy-efficient mining strategy in blockchain-enabled IoT devices has not been explored in the literature. The approach provides a computationally-efficient way to maximize energy efficiency for a blockchain-enabled IoT device.

Organization and Main Results

Sec.II describes the energy-efficient mining problem in blockchain for IoT applications; it is formulated as a multiple stopping time POMDP in Sec.II-A. We explore the comparison between blockchain-enabled IoT devices and multiple stopping time POMDP in Sec.II-D. Sec.III-A discusses our model assumptions, and in Sec.III-B, we derive structural results on the optimal policy for the energy-efficient mining problem in blockchain. Finally, using the structural results, Sec.IV solves for an optimal linear mining policy for a blockchain-enabled IoT device to maximize energy efficiency on a synthetic and real Bitcoin dataset. We also compare the optimal mining policy with other heuristic mining strategies.

II. Energy-Efficient Mining in Blockchain

Recall that our aim is to determine the optimal mining times for an IoT device to act as a blockchain miner. The purpose is to minimize the energy consumed by low-power IoT devices that log their data into a secure (tamper-proof) distributed ledger.

We consider a blockchain-based distributed ledger for IoT applications, where multiple miners compete to mine the next block (see Fig. 2). Mining involves solving a cryptographic puzzle to satisfy PoW: a consensus algorithm for blockchain. An important parameter associated with PoW is the PoW difficulty level; it determines the complexity of the cryptographic puzzle to be solved to add subsequent blocks to the blockchain. The blockchain protocol adjusts the PoW difficulty level to regulate the rate of new blocks: when many miners participate in mining, the total computing power invested in the blockchain is significant, hence, decreasing the expected mining time for the next block; this ensues increase in the PoW difficulty level. As each miner has different computing power and incentives, they individually decide whether or not to invest their computing power for mining. Deciding the mining time instants is crucial for integrating blockchain in an IoT application, as IoT devices have limited energy resources.

We consider the energy-efficient mining problem from the perspective of a single miner. We model the evolution of blockchain and decisions made by the miner as a discrete-time POMDP. Specifically, the energy-efficient mining problem in the blockchain is formulated as a multiple-stopping time POMDP. The reward function encodes the miner's probability of solving the PoW puzzle faster than all other miners. The miner aims to optimize its mining policy to maximize its probability of adding the next block to the blockchain, thereby minimizing the wastage of energy resources.

Sec.II-A formulates the energy-efficient mining in blockchain for IoT devices as a discrete-time multiple stopping time POMDP. We explore the comparison between blockchain-enabled IoT and optimal stopping time POMDP in Sec.II-D.

A. POMDP model for the energy-efficient mining problem in blockchain

In this section, we formulate the energy-efficient mining problem in blockchain as a multiple-stopping time POMDP. We assume that the dynamics of the POMDP are not affected by the mining activity of the IoT device. This assumption is reasonable since the computing power of the IoT device is too small to affect the overall rate of new blocks in the blockchain. We discuss this assumption in detail in (A3). Let $t = 0, 1, 2, \dots$ denote discrete time.

1) System state X_t and the initial state distribution π_0: The system state $X_t \in \mathcal{X}$ denotes the total computing power invested by all the miners in the blockchain at time t with the initial distribution denoted by the pmf $\pi_0 \in \mathbb{R}^{|\mathcal{X}|}$. Here, $\mathcal{X} = \{1, 2, \dots, |\mathcal{X}|\}$ denotes the set of all possible system states. When the total computing power X_t is large, then the mining activity in the blockchain is also large. As discussed in (2), an individual device can only observe the total computing power X_t in noise. Hence, determining the optimal mining time is non-trivial.

2) Transition matrix P: We model the evolution of the total computing power in the blockchain as a time-homogeneous Markov chain with transition matrix P. This Markov assumption is widely used [14]. We will justify the Markov assumption in Sec.IV using a real Bitcoin dataset.

For $i, j \in \mathcal{X}$, elements of the transition matrix $P \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|}$ are

$$P(i,j) = \mathbb{P}\left(X_{t+1} = j \mid X_t = i\right) \tag{1}$$

3) Observation Y_t: An individual IoT device is unaware of the total mining activity at each time instant. Therefore, the IoT device cannot observe the total computing power X_t invested in the blockchain at time t. Instead, the IoT device observes the PoW difficulty level $Y_t \in \mathcal{Y} = \{1, 2, \ldots, |\mathcal{Y}|\}$, which can be viewed as a noisy measurement of the total computing power X_t. The relationship between X_t and Y_t is described by the distribution B.

$$B(i,y) = \mathbb{P}(Y_t = y \mid X = i),\ i \in \mathcal{X}, y \in \mathcal{Y} \tag{2}$$

4) Policy μ: The IoT device decides its mining instants in the blockchain using the policy μ. The IoT device decides whether or not to mine at time t as a function of $Z_t = \{\pi_0, u_0, Y_1, \ldots, u_{t-1}, Y_t\} \in \mathcal{Z}_t$. Here, Z_t denotes the history of information available at time t, and $\mathcal{Z}_t$ denotes the set of all possible history of information at time t. Due to limited power, the IoT device can mine at L time instants over the infinite time horizon. Let $l \in \{1, 2, \ldots, L\}$ index the mining instants. The mining policy of the IoT device is modelled as $\mu : \mathcal{Z} \times \{1, 2, \ldots, L\} \to \mathcal{U}$. Here $\mathcal{U} = \{1, 2\}$ denotes the action space of the IoT device. To decide the l^{th} mining time, the IoT device chooses an action at time t as $u_t = \mu(Z_t, l)$. At time t, $u_t = 1$ corresponds to don't mine, and $u_t = 2$ corresponds to mine.

5) Reward $r : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$: The reward function incentivizes the IoT device to choose its mining time judiciously. It encodes the IoT device's probability of adding a block to the blockchain. An advantage of our submodularity-based mathematical formulation is that we only require the reward function to satisfy the following general structure[2]:

$$r(X,1) = 0,\ r(X,2) \text{ is decreasing in } X \tag{3}$$

Therefore, if the total computing power invested in the blockchain is X and the IoT device decides not to mine in the blockchain, it receives a reward of $r(X,1) = 0$. If the IoT device decides to mine, it earns a reward $r(X,2)$, which encodes the probability of adding a new block in the blockchain.

We now justify (3) for blockchain-enabled IoT devices. Typically, the transaction fee is considered in the literature as the reward function [30], [31]. The expected transaction fee earned depends on the probability of adding a block to the blockchain. As typical in blockchain-enabled IoT devices [21], the devices are responsible for both generating the data and mining a new block, making it appropriate to consider the probability of adding a block as the reward. This probability is directly proportional to the computing power of the IoT device and inversely proportional to the total computing power of the blockchain[3] X. The assumption (3) that the reward is decreasing emerges naturally with the above justification of the transaction fee. The assumption also enables us to characterize mathematically the optimal mining policy in Sec.III. In the absence of this assumption, one would have to rely on heuristic mining policies, several of which are discussed in Sec.IV.

6) Cumulative reward J: For an IoT device, choosing the mining instants myopically to maximize the reward r is not satisfactory as that does not exploit knowledge of the Markov evolution of the state (total computing power X_t). In this paper, we choose the mining times by maximizing the cumulative reward over a finite but random horizon resulting in a multiple-stopping time stochastic control problem. From a practical point of view, the cumulative reward represents the overall energy efficiency of the IoT device. The goal of the IoT device is to maximize its cumulative reward over L mining instants.

$$\begin{aligned} J_\mu(\pi_0) = \mathbb{E}_\mu \Bigg[&\sum_{t=0}^{\tau_1-1} \rho^t r(X_t, 1) + \rho^{\tau_1} r(X_{\tau_1}, 2) + \ldots \\ &\ldots + \sum_{t=\tau_{L-1}+1}^{\tau_L-1} \rho^t r(X_t, 1) + \rho^{\tau_L} r(X_{\tau_L}, 2) \mid \pi_0 \Bigg] \\ \tau_{i+1} = \min\{t > \tau_i : u_t = 2\},\ &i \in \{1, \ldots, L\},\ \tau_0 = 0 \end{aligned} \tag{4}$$

Here, τ_i represents the i^{th} mining instant in the blockchain. The discount factor ρ represents the IoT device's decreased value assigned to rewards obtained in the future. One can also consider risk-averse cost function as in [32]. The optimal policy ensures that the IoT device maximizes the cumulative reward for adding a new block to the blockchain, thereby maximizing energy efficiency.

It is important to emphasize that the total computing power X_t is not observed by the IoT device. It only observes the PoW difficulty level Y_t, which is a noisy measurement of the total computing power. Put simply, by observing a noisy Markov chain, what are the optimal L mining instants? Therefore, (4) is a multiple-stopping time POMDP.

B. Belief State Representation

The POMDP for the energy-efficient mining problem in blockchain (Sec.II-A) can be formulated as the standard Markov decision process (MDP) by introducing a belief state. The belief state π_t is the posterior probability distribution of the underlying state given the observations until the present time. It is updated recursively using the Bayesian update [8] as new observations are received.

$$\begin{aligned} \pi_{t+1} &= T(\pi_t, Y_t) \\ T(\pi, y) &= \frac{B_y P^\intercal \pi}{\sigma(\pi, y)}, \quad \sigma(\pi, y) = \mathbb{1}_{|\mathcal{X}|}^\intercal B_y P^\intercal \pi \end{aligned} \tag{5}$$

[2]Our structural results can be generalized to the case when the difference in reward $r(X,2) - r(X,1)$ is decreasing in X.

[3]Solving PoW involves an exhaustive search over all possible values of nonce [2]. The search continues until a desired number of zeros is found at the beginning of the hash code. Therefore, a miner with higher computing power can search more nonce per unit of time, leading to a higher probability of successfully adding a block to the blockchain. Thus, the probability that an IoT device will add a block to the blockchain can be modelled as a Bernoulli random variable with $p = \frac{1}{X}$. This assumes that the computing power of the IoT device is normalized to 1, and it is small compared to the total computing power X.

Here, $B_y = \text{diag}(B(1,y), B(2,y), \ldots, B(|\mathcal{X}|,y))$ where B is the observation matrix (2); and $\mathbb{1}_{|\mathcal{X}|}$ represents the $|\mathcal{X}|$-dimensional column vector of ones and its transpose is denoted as $\mathbb{1}^{\mathsf{T}}_{|\mathcal{X}|}$.

In the MDP formulation, one designs the policy μ as a function of belief state $\pi_t \in \Pi$. Here, Π is a simplex, also known as belief space. It is well-known that the belief state is a sufficient statistic [8], and designing a policy as a function of the belief state yields the same optimal solution. Using belief state facilitates analysis, but because the belief space is a simplex, it yields an MDP with continuous state-space Π. In this paper, we use the equivalent MDP formulation of the energy-efficient mining problem in blockchain using belief state to derive our structural results in Sec.III. The latter is used to design a linear mining policy as a function of the belief state. Algorithm 1 provides the steps to compute the cumulative reward for the energy-efficient mining problem for a given mining policy μ.

Algorithm 1 Simulating energy-efficient mining problem in blockchain using belief state π_t given the mining policy μ

Require: $\pi_0, \mathcal{X}, \mathcal{Y}, P, B, r, \rho, L$ and an upper limit on the horizon length T

Initialize $l \leftarrow 1,\ J \leftarrow 0$
for $t = 0, 1, 2, \ldots, T$ **do**
  Compute $u_t \leftarrow \mu(\pi_t, l)$
  if $(u_t = 2)$ **then**
    $J \leftarrow J + \rho^t r(\pi_t, 2)$, $l \leftarrow l + 1$.
    if $(l = L)$ **then**
      return J
    end if
  else
    $J \leftarrow J + \rho^t r(\pi_t, 1)$
  end if
  Generate a new observation y_t and compute π_{t+1} using (5).
end for

C. Equivalent formulation as a discounted-cost POMDP

The multiple-stopping time POMDP for the energy-efficient mining in blockchain can be formulated as an infinite-horizon POMDP. This is achieved by augmenting a fictitious absorbing state $|\mathcal{X}|+1$ with the continue reward $r(|\mathcal{X}|+1, 1) = 0$. When the last stop is made, the belief state π_t (ref. Sec.II-B) transitions to $e_{|\mathcal{X}|+1}$. Here, $e_{|\mathcal{X}|+1} = (0, \ldots, 0, 1) \in \mathbb{R}^{|\mathcal{X}|+1}$. The cumulative reward (4) for the multiple stopping POMDP is equivalent to $J_\mu(\pi_0) = \mathbb{E}_\mu \Big[\sum_{t=0}^{\tau_1 - 1} \rho^t r(X_t, 1) + \rho^{\tau_1} r(X_{\tau_1}, 2) + \ldots + \sum_{t=\tau_{L-1}+1}^{\tau_L - 1} \rho^t r(X_t, 1) + \rho^{\tau_L} r(X_{\tau_L}, 2) + \sum_{t=\tau_L+1}^{\infty} \rho^t r(|\mathcal{X}|+1, 1) \mid \pi_0\Big]$. In the standard form of POMDP, the transition matrix depends on the input. To obtain an input-dependent transition matrix, one can use the modified state $(l, X), X \in \mathcal{X}, l \in \{1, 2, \ldots, L\}$. Here, l denotes the stop number and X denotes the original state (ref. Sec.II-A). To specify the new transition matrix, we order the modified state as:

$$((1,1), \ldots, (1,|\mathcal{X}|), (2,1), \ldots, (2,|\mathcal{X}|), \ldots, (L,1), \ldots, (L,|\mathcal{X}|), |\mathcal{X}|{+}1))$$

For this ordering, the transition matrix with $u = 1$ (ref. (3)) is given by $\widetilde{P}_{u=1} = \text{diag}\{P, \ldots, P, 1\}$. Here, operator diag is used to construct a block diagonal matrix, and P is defined in (1). The transition matrix with $u = 2$ (ref. (3)) is given by

$$\widetilde{P}_{u=2} = \begin{bmatrix} 0 & P & & & \\ 0 & 0 & P & & \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & 0 & P & 0 \\ 0 & \cdots & 0 & 0 & 1_{|\mathcal{X}|\times 1} \\ 0 & \cdots & 0 & 0 & 1 \end{bmatrix}$$

D. Discussion. Multiple Stopping Time Model for Blockchain-Enabled IoT

We now discuss the POMDP model in the context of blockchain for IoT. The combination of IoT and blockchain enables secure, transparent and scalable data sharing amongst a large number of users [6] [33]. We consider the example of a sensor network consisting of heterogeneous IoT devices (see Fig.2): low-power devices like Raspberry Pi and high-power devices like PCs. Low-power devices are typically used for data collection at remote locations, while high-power devices monitor time-critical applications. In our setup, the devices in the sensor network use a blockchain platform, like Etherium [34], to log their data. The IoT devices have to compete to solve a PoW faster than other miners to add a new block in the blockchain. This improves the security of the blockchain: it is difficult to tamper with transactions in the blockchain. However, PoW involves solving a cryptographic puzzle which is energy-intensive. As low-power IoT devices have limited energy resources, they need to optimize their mining time instants to maximize their probability of adding a new block to the blockchain. This would minimize the wastage of energy by resource-constrained IoT devices. We described our POMDP formulation for the energy-efficient mining problem in blockchain for IoT applications in Sec.II-A. We now discuss the model parameters in the context of IoT and blockchain.

1) Markovian system dynamics for mining in blockchain: The probability of a miner adding a new block to the blockchain is determined by the total computing power invested in it. Therefore, it is crucial for an IoT device to keep track of the total computing power. Our approach involves modeling the total computing power invested in the blockchain as a Markov chain with a transition matrix P. This matrix captures how the computing power changes over time as individual miners make decisions based on their own trade-offs between mining cost and reward.

Remarks. Estimating the transition matrix. Even though the actual total computing power in the blockchain is not directly observable, it can still be estimated based on the rate of new blocks and the PoW difficulty level. [35] includes a record of the estimated total computing power in the past, which can

be used to estimate the transition matrix. This can be done by grouping the historical data into a specified number of states and applying a maximum likelihood estimator (MLE) to obtain the transition matrix.

2) PoW difficulty level as a noisy observation of the system state: Recall that the observations are the PoW difficulty level, a noisy observation of the system state. With a larger total computing power, new blocks are mined faster on average, and thus, the blockchain protocol adjusts the PoW difficulty level to maintain a constant rate of new blocks. As the IoT device cannot observe the total computing power invested in the blockchain, it uses the PoW difficulty level to update the belief about the total computing power.

Remarks. Estimating the observation distribution. [35] provides a historical record of the estimated total computing power and the PoW difficulty level in the blockchain. One can use an MLE to estimate the observation distribution B from the data.

3) Probability of adding a new block in the blockchain: The IoT device wants to mine the blockchain to log its sensor readings while maximizing its energy efficiency, which is defined as the probability of adding the next block to the blockchain. This preference is modelled as the reward in the optimal stopping time problem within the framework of POMDP. The reward function is proportional to the computing power of the IoT device and inversely proportional to the total computing power invested in the blockchain if the IoT device decides to mine. Otherwise, it receives no reward.

4) Mining policy and the total number of mining instants: We model a single IoT device as a decision maker and optimize its mining time instants so as to maximize its energy efficiency. Due to the energy constraint imposed on the IoT device, it can only engage in mining for a finite number of time instants. Consequently, the IoT device seeks to increase its likelihood of adding a new block to the blockchain in order to minimize energy wastage.

E. Optimizing the number of mining instants in blockchain

In Sec.II-A, we presented our model for the energy-efficient mining problem in the blockchain. Our model assumed that the number of mining instants L was fixed and known to the IoT device. In a realistic scenario, the IoT device also has to optimize the number of mining instants L. This is because different sensors (IoT devices) in an IoT application record data at different rates based on their task. The amount of data that needs to be logged in the blockchain is proportional to the data rate. The optimization problem to optimize the number of mining instants L is:

$$\max_{L} J_{\mu^*,L}(\pi_0) - \Omega(L) \tag{6}$$

Here, $J_{\mu^*,L}(\pi_0)$ denotes the optimal cumulative reward (4) when number of mining instants is L; Ω is the cost function for choosing a particular L. We assume that Ω is an increasing and convex function of L. This is because the energy consumed increases with the number of mining instants L. Also, as the energy consumption increases, the size of the battery increases and incurs additional cost; this motivates the convexity of Ω. It can be empirically verified that $J_{\mu^*,L}$ is concave in L. Therefore, (6) is a convex optimization problem. Although L is discrete-valued, we can solve (6) in continuous domain and compare the nearest integer solutions to obtain the optimal number of mining instants.

To summarize, we formulated the energy-efficient mining problem in blockchain as a multiple-stopping time POMDP. Due to the curse of dimensionality, it is difficult to solve the optimal mining policy. So, in Sec.III, we derive structural results, which would be exploited in Sec.IV for obtaining an optimal linear mining policy.

III. Structural Results for Energy-Efficient Mining in Blockchain

This section presents structural results for the energy-efficient mining problem in blockchain (ref. Sec.II). Sec.III-A discussed the model assumptions to derive the structural results. In Sec.III-B, we first show that the optimal mining policy in blockchain has a threshold structure (Theorem 1). Sec.III-C discusses the significance of the structural results for blockchain-enabled IoT devices. Sec.III-D exploits Theorem 1 to design a linear mining policy. This is followed by necessary and sufficient conditions on the parameters (Theorem 2) of the linear mining policy to satisfy the structural results. Optimizing the parameters of the linear policy corresponds to solving a constrained optimization problem which is difficult. Hence, Sec.III-E describes the parameters of the linear mining policy in spherical coordinates. The spherical coordinates simplify the problem of optimizing the parameters to an unconstrained optimization problem. Sec.III-F discusses the policy gradient algorithm to optimize the parameters of the linear mining policy in spherical coordinates. The main outcome of this section is to construct a computationally efficient, optimal linear mining policy for the blockchain-enabled IoT device. This is achieved by exploiting the structural results for the selection of the policy parameters.

A. Assumptions on multiple-stopping time POMDP model

We now discuss our assumptions on the model for the energy-efficient mining problem in blockchain for IoT applications. To understand our assumptions, we need to define the property of total positivity of order 2 (TP2).

Definition 1 (Total positivity of order 2 (TP2) [8])**.** *A stochastic matrix A is TP2 if all the second-order minors are non-negative, i.e., the determinants $\begin{vmatrix} A_{i_1 j_1} & A_{i_1 j_2} \\ A_{i_2 j_1} & A_{i_2 j_2} \end{vmatrix} \geq 0,\ \forall i_1 < i_2, j_1 < j_2$ .Here, A_{ij} denotes the $(i,j)^{th}$ element of the matrix A.*

The assumptions (A1)-(A3) will serve as the basis for deriving the structural properties of the optimal policy μ^* in Sec.III.

(A1) The transition matrix P is totally positive of order 2 (TP2) (Definition 1). To satisfy the TP2 assumption, we impose two conditions on the Markov chain X_t: (1) it varies slowly with time, i.e., diagonal terms are dominant, (2) the transition matrix P has a tri-diagonal structure.

Justification: If there are no collusions among the miners, the mining activity changes slowly with time. Hence, the tri-diagonal assumption is valid, and it can be satisfied by using a small enough sampling time and/or appropriately binning the states of the Markov chain X_t.

(A2) Observation distribution B (defined in (2)) is totally positive of order 2 (TP2).
Justification: Since the observations are the PoW difficulty level, they are non-negative integers. We can employ empirical methods to approximate the observation distribution within the class of TP2 distributions. This is not restrictive as several well-known distributions over non-negative integers satisfy this property [36] such as binomial, Poisson, geometric distribution, etc. In the numerical section (Sec.IV) involving a real Bitcoin dataset, we fit the data to the nearest TP2 distribution to the observation distribution.

(A3) The dynamics of X_t are not affected by the miner's (decision maker) action u_t at time t.
Justification: This assumption is realistic for an IoT device with small computing power compared to the total computing power ([37] [21] provides a comparative study of the typical computing power and energy used by an IoT device and a PC). This assumption is further justified because the IoT device's computing power is negligible to make a significant impact on the rate at which new blocks are added in the blockchain[4].

B. Structural results for the optimal mining policy

Before discussing our structural results, we need to define the maximum likelihood ratio (MLR) ordering on the belief space Π (see Sec.II-B). The MLR ordering is preserved under conditional expectations [8], making it suitable for Bayesian problems. The MLR ordering defines a partial order on a simplex, and we use it to show that the mining policy is monotone with respect to the belief state.

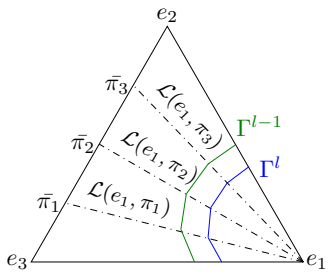
Definition 2 (MLR ordering)**.** *Let $\pi_1, \pi_2 \in \Pi$ be two belief states. Then, π_1 is greater than π_2 with respect to MLR ordering, denoted as $\pi_1 \geq_r \pi_2$, if $\pi_1(j)\pi_2(i) \geq \pi_2(j)\pi_1(i), \forall i < j$*

To understand the threshold property, let us define two families of sets: (1) *mine set* $M^l,\ l \in \{1, 2, \ldots, L\}$ containing the belief states where it is optimal to mine, (2) *don't mine set* $D^l,\ l \in \{1, 2, \ldots, L\}$ containing the belief states where it is optimal to not mine.

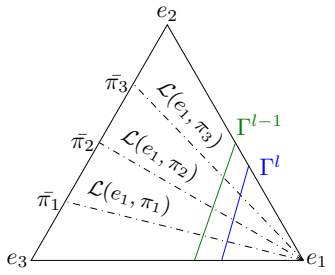
$$D^l = \{\pi : \mu^*(\pi, l) = 1\}, \quad M^l = \{\pi : \mu^*(\pi, l) = 2\} \tag{7}$$

Theorem 1 shows the existence of an optimal mining policy that partitions the belief space Π into two connected regions for each $l \in \{1, 2, \ldots, L\}$. Moreover, the family of sets M^l and D^l are nested. Theorem 1 also shows the monotonicity of the optimal mining policy μ^* in the belief space Π. Fig. 3a shows a visual illustration of the Theorem 1.

[4](A3) allows us to deploy our model for multiple low-power IoT devices as long as the total computing power of the IoT devices is significantly small compared to the total computing power invested in the blockchain.



(a) General mining policy with a threshold structure



(b) Linear mining policy with a threshold structure

Fig. 3: Visual illustration of Theorem 1 and Theorem 2 for $|\mathcal{X}|= 3$. Γ^l denotes the threshold for deciding the l^{th} mining instant. Γ^l partitions the belief space Π into two connected regions M^l (right of Γ^l) and D^l (left of Γ^l). The linear policy in Sec.III-D approximates the threshold Γ^l using a linear hyperplane. The dashed lines $\mathcal{L}(e_1, \pi)$ in the figure are used in the proof to show the existence of a threshold policy. From a practical point of view, we exploit the structure to estimate the optimal linear mining policy using a stochastic gradient algorithm.

The belief space Π consists of probability simplices. Hence, a total order can not be defined. To show the monotonicity of the optimal mining policy μ^*, we define a family of total order subsets $\mathcal{L}(e_i, \bar{\pi})$ of the belief space Π. Here, $i \in \{1, 2, \ldots, |\mathcal{X}|\}$ indexes the elements in $\mathcal{X}$ and e_i denotes the i^{th} standard basis vector in $\mathbb{R}^{|\mathcal{X}|}$.

$$\begin{aligned} \mathcal{H}_i &:= \{\bar{\pi} \in \Pi,\ \bar{\pi}_i = 0\} \\ \mathcal{L}(e_i, \bar{\pi}) &:= \{\pi \mid \pi = \gamma e_i + (1-\gamma)\bar{\pi},\ \gamma \in [0, 1]\},\ \bar{\pi} \in \mathcal{H}_i \end{aligned} \tag{8}$$

The set $\mathcal{L}(e_i, \bar{\pi}), i \in \{1, 2, \ldots, |\mathcal{X}|\}$ consists of line segments in the belief space Π; each element defines a totally ordered subset of the belief space Π with respect to the monotone likelihood ratio (MLR) ordering (Definition 2). In Theorem 1, we use the MLR order to show that the optimal mining strategy in blockchain is monotonically decreasing in the belief state on the lines $\mathcal{L}(e_1, \bar{\pi})$ and $\mathcal{L}(e_{|\mathcal{X}|}, \bar{\pi})$.

Theorem 1. *Under assumptions (A1)-(A3), for each $l \in \{1, 2, \ldots, L\}$,*

A) *There exists an optimal policy $\mu^*(\pi, l)$ that is decreasing on lines $\mathcal{L}(e_1, \bar{\pi})$, and $\mathcal{L}(e_S, \bar{\pi})$ (defined in* (8)*).*

B) *There optimal policy μ^* partitions the belief space* Π *into two individually connected sets* M^l *and* D^l *(defined in* (7)*).*

C) $M^{l-1} \supset M^l$

Proof. See Sec.VI of supplementary material. □

Theorem 1.A asserts that the optimal mining strategy $\mu^*(\pi, l),\ l \in \{1, 2, \ldots, L\}$ is monotonically decreasing on lines $\mathcal{L}(e_1, \bar{\pi})$, and $\mathcal{L}(e_S, \bar{\pi})$. This implies that there exists a threshold for the belief state π above which it is optimal to mine in the blockchain. Theorem 1.B shows that the threshold partitions the belief space into two connected sets. Theorem 1.C shows that the sets of belief state π, indexed by $l \in \{1, 2, \ldots, L\}$, such that $\mu^*(\pi, l) = 1$ are nested.

C. Implications for Energy-Efficient Mining

For modeling the energy-efficient mining problem, one needs to discretize the set of total computing power invested in the blockchain. This can lead to a large number of states for the POMDP formulation described in Sec.II-A. Therefore, the dynamic programming solution to POMDP is not practical for implementation on IoT devices. This is because the look-up table corresponding to the optimal mining policy grows exponentially with the number of states and requires search operations at each time instant. This is detrimental for IoT devices which have limited computational and energy resources. Theorem 1 enables a less demanding approach to store the optimal mining policy in blockchain, both in terms of memory requirements and computational complexity. Memory requirement is reduced by solving a parametrized policy; this also reduces computational complexity as search operations are avoided. Moreover, under the assumption that the blockchain is time-invariant, IoT devices can be pre-programmed with the optimal linear mining policy before their deployment. This alternate solution approach will be discussed in Sec.IV, where we describe our approach to compute an optimal linear policy for the energy-efficient mining problem in the blockchain.

D. Linear mining policy for a blockchain-enabled IoT device

This subsection focuses on the design of a linear mining policy for a blockchain-enabled IoT device which meets the structural results outlined in Theorem 1. Our main result is summarized in Theorem 2, which characterizes the conditions on the parameters of the linear mining policy (9).

Consider a linear mining policy of the form

$$\mu_\theta(\pi, l) = \begin{cases} 2, & \begin{bmatrix}\theta_l & 1 & 0\end{bmatrix} \begin{bmatrix}-1 \\ \pi \\ 1\end{bmatrix} \geq 0 \\ 1, & \text{otherwise} \end{cases} \tag{9}$$

Here, $\theta_l \in \mathbb{R}^{L-1}$ is the parameter for the linear mining policy to decide l^{th} mining instant. θ. We can restrict the search space for θ using structural results from Sec.III. Theorem 2 enumerates necessary and sufficient conditions on the parameter θ so that the linear mining policy (9) satisfies Theorem 1. The conditions guarantee that all MLR-decreasing linear policies are included and no non-MLR-decreasing linear policies are excluded. Fig. 3b shows a visual illustration of the linear policy (9) and Theorem 2.

Theorem 2. *Assuming the set M^l is non-empty, the necessary and sufficient conditions for the linear policy* (9) *to satisfy the structural results in Theorem 1 are: 1) $\theta_l(i) \geq 0,\ \forall i, \forall l$, 2) $\theta_l(2) \geq 1,\ \forall l$ and $\theta_l(i) \leq \theta_l(2),\ \forall i > 2, \forall l$, 3) $\theta_l(1) \leq \theta_{l+1}(1)$ and $\theta_l(i) \geq \theta_{l+1}(i),\ \forall i > 1, \forall l$*

Proof. See Sec.VII of supplementary material. □

E. Parameters of the linear mining policy in spherical coordinates

The optimization of the linear mining policy (9) subject to conditions in Theorem 2 can be formulated as a constrained optimization problem. In this subsection, we present a transformation of the policy parameters θ into spherical coordinates. This transformation will be exploited in Sec.III-F to formulate the optimization of the policy parameters as an unconstrained optimization problem.

The parameter θ in (9) has to satisfy the conditions described in Theorem 2 so as to satisfy the structural results in Theorem 1. We now define a relation between parameter $\theta \in \mathbb{R}^{L-1}$ in Euclidean coordinates and parameter $\phi \in \mathbb{R}^{L-1}$ in spherical coordinates:

$$\theta_l^\phi(i) = \begin{cases} \phi_1^2(1) \prod_{j=l}^{L-1} \sin^2\left(\phi_j(1)\right), & i = 1 \\ 1 + \phi_1^2(2) \prod_{j=2}^{l} \sin^2\left(\phi_j(2)\right), & i = 2 \\ \theta_l(2) \prod_{j=1}^{L} \sin^2\left(\phi_j(i)\right), & i > 2. \end{cases} \tag{10}$$

It can be easily verified that the θ obtained using (10) satisfies the conditions in Theorem 2. So, instead of optimizing the parameter θ using a constrained optimization problem, we can optimize the parameter ϕ as an unconstrained optimization problem.

F. Policy gradient reinforcement learning algorithm [38]

In this subsection, we describe the policy gradient algorithm to optimize the parameters of the linear mining policy (9) for a blockchain-enabled IoT device. As it is difficult to obtain a closed-form expression for the cumulative reward as a function of the mining policy, we utilize techniques from stochastic optimization to optimize the policy parameters.

Algorithm 2 Policy gradient algorithm

Require: $\pi_0, \mathcal{X}, \mathcal{Y}, P, B, r, \rho, L, \epsilon, \zeta, \kappa, \nu, \psi$

1: Initialize $\phi^{(0)}$ randomly.
2: **for** $n = 1$ to N **do**
3: Compute $\theta^{\phi^{(n)}}$ using (10) and compute a_n, c_n using (12).
4: Use Algorithm 1 to simulate the POMDP for the energy-efficient mining problem in Sec.II-A using the linear mining policy (9) with policy parameters $\theta^{\phi^{(n)}} + c_n\omega_n$ and $\theta^{\phi^{(n)}} + c_n\omega_n$. Update the parameter $\phi^{(n)}$ using (11).
5: **end for**
6: **return** $\theta^{\phi^{(N)}}$

The policy gradient algorithm to optimize the parameter θ in spherical coordinates is as follows: (1) initialize the parameter ϕ, (2) update ϕ using (11). Algorithm 2 summarizes the steps in the policy gradient algorithm.

$$\hat{\nabla}_{\phi} J\left(\theta^{\phi^{(n)}}\right) = \frac{J\left(\theta^{\phi^{(n)}} + c_n\omega_n\right) - J\left(\theta^{\phi^{(n)}} - c_n\omega_n\right)}{2c_n}\omega_n$$
$$\phi^{(n+1)} = \phi^{(n)} + a_n\hat{\nabla}_{\phi} J\left(\theta^{\phi^{(n)}}\right) \tag{11}$$

Here, $\phi^{(n)}$ is the value of parameter ϕ at n^{th} iteration. $\theta^{\phi^{(n)}}$ is the value of the parameter in Euclidean coordinates obtained using (10). The parameters a_n and c_n are typically chosen as:

$$a_n = \varepsilon(n+1+\varsigma)^{-\kappa}, \quad c_n = \psi(n+1)^{-v}, \\ 0.5 < \kappa \leq 1, \varepsilon, \quad \varsigma > 0, 0.5 < v \leq 1, \quad \psi > 0 \tag{12}$$

To summarize, we showed that the optimal mining policy has a threshold structure (Theorem 1), and it partitions the belief space into two connected sets. We exploited these results to design a linear mining policy (9) for the energy-efficient mining problem in the blockchain. We specified conditions on the parameters of the linear mining policy (Theorem 2) so that it satisfies the structural results in Theorem 1. This was followed by the transformation of the parameters of the linear mining policy to spherical coordinates (10). The latter facilitated us to formulate the optimization of the policy parameters as an unconstrained optimization problem. We also presented a policy gradient algorithm (11) to optimize the linear policy's parameters in the spherical coordinates.

IV. Numerical results and bitcoin dataset

In this section, we compute an optimal linear mining policy (9) for the energy-efficient mining problem in blockchain[5] using synthetic (Sec.IV-A and Sec.IV-B) and a real bitcoin dataset (Sec.IV-C). The optimal linear mining policy (9) satisfies the structural results in Theorem 1 and is suitable for IoT devices: the linear policy uses less memory, less computation and can be computed offline. To illustrate the performance of our proposed optimal linear policy (P2), we compare it with four other mining strategies:

(P1) **Optimal mining policy**: a mining policy obtained using the value iteration algorithm for the multiple stopping time problem. This qualifies as the ground truth since it is the optimal solution [8].

(P2) **Optimal linear mining policy**: a linear mining policy (9) obtained using the policy gradient algorithm (11).

(P3) **First L mining**: a policy that chooses the first L time instants for mining.

(P4) **Random policy**: a policy that decides to mine or not at each time instant by tossing a biased coin. For a random policy, the probability of heads for a biased coin can be adjusted based on the rate of data sensing. In our simulation, we consider a fair coin.

[5]All the numerical results are reproducible, and the codes are available on GitHub at https://github.com/anuraggin/blockchain˙pomdp.git

TABLE I: Model parameters for the energy-efficient mining problem in blockchain for an IoT device (Low-dimensional synthetic data)

Parameters	Eq.	Values
$\{\pi_0, \mathcal{X}\}$	(1)	$\{[0\ 0\ 1], \{1,2,3\}\}$
P	(1)	$\begin{bmatrix} 0.5 & 0.5 & 0 \\ 0.25 & 0.5 & 0.25 \\ 0 & 0.5 & 0.5 \end{bmatrix}$
$\mathcal{Y}$	(2)	$\{1,2,3,4,5\}$
$B^{\intercal}$	(2)	$\begin{bmatrix} 0.2384 & 0.1686 & 0.0221 \\ 0.3129 & 0.2580 & 0.0955 \\ 0.3951 & 0.3258 & 0.1207 \\ 0.0629 & 0.3 & 0.4546 \\ 0.0044 & 0.0669 & 0.1741 \end{bmatrix}$
$[r(1,2), r(2,2), r(3,2)]$	(3)	[0.1, 0.01, 0.001]
$[\rho, L]$	(4)	[0.9, 3]
$[\epsilon, \zeta, \kappa, \nu, \psi]$	(12)	$[0.7, 0.1, 0.6, 0.6, 0.1]$

TABLE II: Comparison of the optimal mining policy with other heuristic mining strategies

(a) Low-dimensional synthetic dataset

Policy	**Reward** (4)
(P1) Optimal mining policy	0.0579
(P2) Optimal linear mining policy	0.0549
(P3) First L mining	0.0204
(P4) Random policy	0.0348
(P5) Reinforcement learning	0.0452

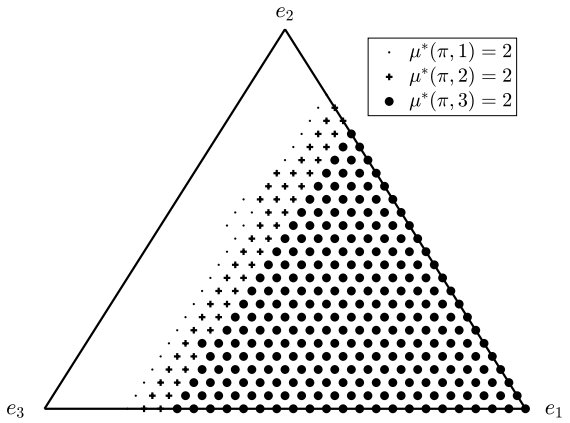
(b) Real Bitcoin dataset

Policy	**Reward** (4)
(P1) Optimal mining policy	0.2021
(P2) Optimal linear mining policy	0.1991
(P3) First L mining	0.1265
(P4) Random policy	0.1432
(P5) Reinforcement learning	0.1621

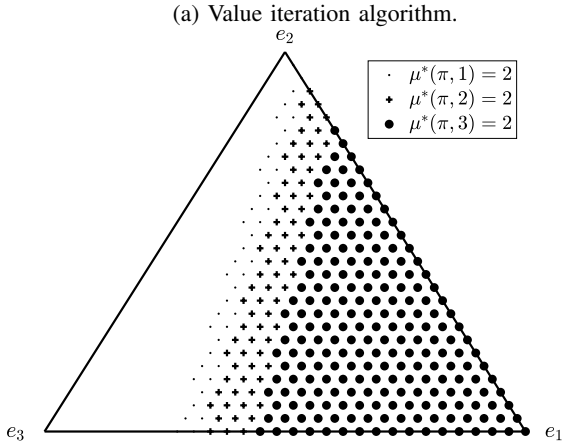
(P5) **Reinforcement learning based mining policy**: Reinforcement learning has been exploited in the literature to compute optimal mining policy under various settings [17] [39]; it uses softmax parametrization to model policy. In the reinforcement learning paradigm, the parameters of the MDP are unknown to the decision maker. Hence, the policy is designed as a function of the current and past observations. $\Pr(\mu(\pi,l) = u) = \frac{\exp\left(\theta_{l,u}^{\intercal} W_t\right)}{\sum_{u=1}^{2}\exp\left(\theta_{l,u}^{\intercal} W_t\right)}$ Here, $W_t := [y_t \quad y_{t-1} \quad \dots \quad y_{t-N+1}]^{\intercal}$ is the observation window,$\theta_{l,u} \in \mathbb{R}^N, \forall l, \forall u$ is the policy parameters; N denotes the size of observation window for designing a mining policy using the softmax parametrization. For simulation, we chose $N = 2$ so that the number of parameters in the softmax parametrization is similar to that of the linear mining policy.

A. Low-dimensional numerical examples using synthetic data

In this subsection, we use synthetic model parameters for the proposed energy-efficient mining problem in blockchain (Sec.II-A). Our model parameters are summarized in Table I. We chose $|\mathcal{X}| = 3$ to visualize the structure of the optimal mining policy. We solved the optimal mining policy for the energy-efficient mining problem in blockchain using the value iteration algorithm (Fig. 4a) and the optimal linear mining



(a) Value iteration algorithm.



(b) Policy gradient algorithm.

Fig. 4: Optimal mining policy for the energy-efficient mining problem in blockchain on a low-dimensional synthetic data. The triangle represents the belief space for the energy-efficient mining problem in blockchain. The belief space has been discretized into 30 equal parts along each axis. The markers indicate the belief states where the optimal action is to mine in the blockchain. The optimal mining policy has a threshold and nested structure (Theorem 1).

policy using the policy gradient algorithm (Fig. 4b). Fig. 5a shows the convergence of the policy gradient algorithm. The optimal mining policy gives an expected reward of 0.0579 whereas the optimal linear mining policy gives an expected reward of 0.0549. Therefore, there is a 5.5% loss in the expected reward for using a linear mining policy. On the positive side, the optimal linear mining policy uses much less memory and requires less computation compared to the solution of the value iteration algorithm. This is because the solution of value iteration corresponds to storing a look-up table, the size of which grows exponentially with the size of state space. Moreover, for the optimal mining policy, the IoT device has to perform a search operation on the look-up table at each time instant to obtain the optimal action. The optimal linear mining policy overcomes these two drawbacks making it suitable for resource-constrained IoT devices.

We also compared our proposed mining policy with other heuristic mining strategies. The results are summarized in Table IIa. We observe that the optimal linear mining policy

TABLE III: Model parameters for the energy-efficient mining problem in blockchain for an IoT device (High-dimensional synthetic data)

Parameters	Eq.	Values
$\{\pi_0, \mathcal{X}\}$	(1)	$\{[0\ 0\ \ldots\ 0\ 1], \{1, 2, \ldots, 10\}\}$
P	(1)	$P_{i,i} = 0.5,\ \forall i,\ P_{1,2} = P_{10,9} = 0.5,$ $P_{i,j-1} = P_{i,j+1} = 0.25, \forall i \neq \{1, 10\}$
$\{\mathcal{Y}, B(i,y)\}$	(2)	$\left\{\{1, 2, \ldots, 12\}, \frac{(10\,i)^y \exp(-10\,i)/y!}{\sum_y (10i)^y \exp(-10i)/y!}\right\}$
$r(X, 2)$	(3)	$1/X^3$
$[\rho, L]$	(4)	$[0.9, 3]$
$[\epsilon, \zeta, \kappa, \nu, \psi]$	(12)	$[0.7, 0.1, 0.6, 0.6, 0.1]$

provides a significant improvement over other heuristic policies: 69% improvement over the first L mining policy, 58% improvement over the random policy, and 22% improvement over reinforcement learning-based mining policy.

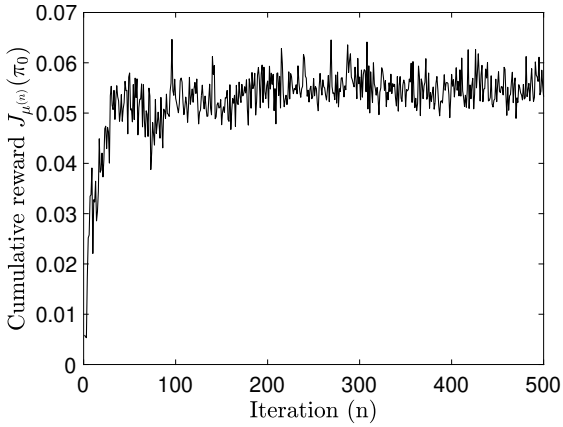
B. High-dimensional numerical example using synthetic data

In this subsection, we solve a higher dimensional energy-efficient mining problem in blockchain (Sec.II-A) using synthetic data. The model parameters are summarized in Table III. Fig.5b shows the convergence of the policy gradient algorithm. Even for high dimensional data, the policy gradient algorithm converges within 200 iterations using a suitable choice of parameters for the policy gradient algorithm. Therefore, if the total number of miners in an IoT application evolves slowly with time, the IoT device can also use the policy gradient algorithm to update its optimal linear mining policy.

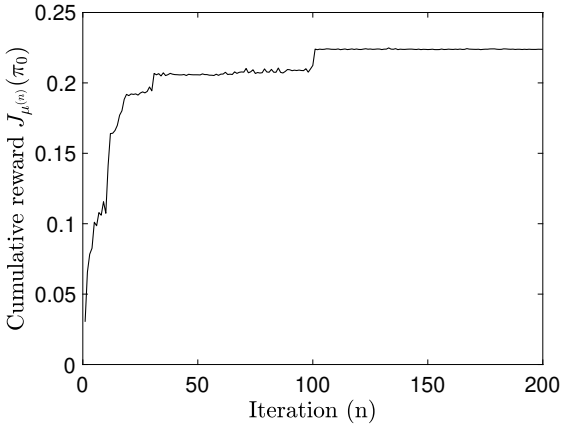
C. Numerical examples using real bitcoin dataset

Now, we use real Bitcoin data to estimate the model parameters for the energy-efficient mining problem in blockchain (Sec.II-A). The estimated model parameters are used to solve the optimal mining policy for a blockchain-enabled IoT device.

A record of historical data on bitcoin mining is available in [35]. We use their data on the estimated hash rate (total computing power) and the difficulty (PoW difficulty level) to estimate the transition matrix and the observation matrix. The dataset contains total computing power and the PoW difficulty sampled once per day. As the total number of miners in the Bitcoin network is growing with time, we use a small range of data to compute the model parameters. Fig. 6 shows the plot of the Bitcoin dataset between April 2022 - August 2022. To compute the model parameters, we first group the data into bins of uniform size as follows: (1) total computing power is grouped into three bins, i.e., $|\mathcal{X}|= 3$, (2) PoW difficulty level is grouped into five bins, i.e., $|\mathcal{Y}|= 5$. We used the binned data to estimate the transition matrix P using the MLE estimator (see Table IV). The estimated transition matrix P satisfies the tri-diagonal structure assumption, and the diagonal terms are dominant, thereby satisfying the assumption (A1). Hence, (A1) is easy to satisfy with a suitable choice of binning and sampling interval. However, in our case, the estimated observation matrix using the MLE did not yield a TP2 matrix. This could be due to external factors or the insufficient size of the dataset. Hence, to exploit our structural results, we estimate

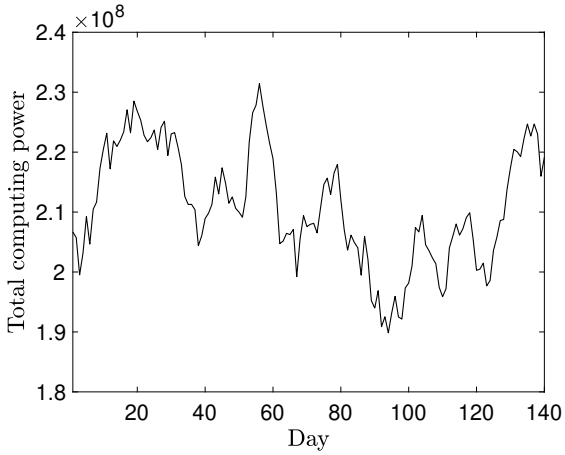


(a) Low-dimensional synthetic data (Sec.IV-A)

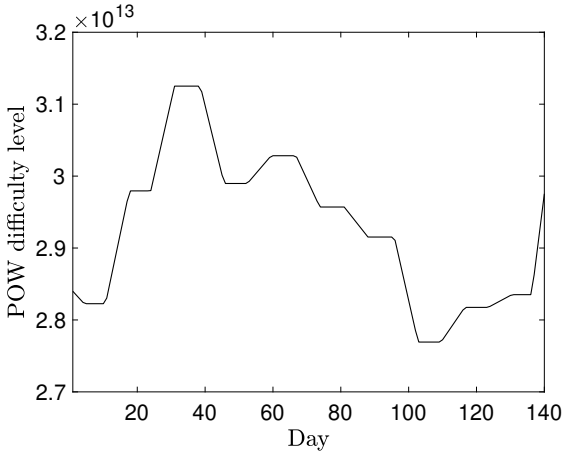


(b) High-dimensional synthetic data (Sec.IV-B).

Fig. 5: Cumulative reward for the energy-efficient mining problem at each iteration of the policy gradient algorithm. $\mu^{(n)}$ denotes the linear mining policy μ at iteration n. The policy gradient algorithm converges within 100 iterations. Therefore, if the total miners in an IoT application evolve slowly with time, the IoT device can also use the policy gradient algorithm to update its optimal mining policy.



(a) Total computing power vs. time. The total computing power was estimated using the rate of new blocks in the blockchain over 24 hours interval.



(b) PoW difficulty level vs. time. The PoW difficulty level is the average of the PoW difficulty level computed over 24-hour interval.

Fig. 6: Bitcoin mining dataset between April 2022 - August 2022 (Source: [35])

.

the observation matrix $\hat{B}$ within the class of TP2 distribution[6]. The model parameters are summarized in Table IV.

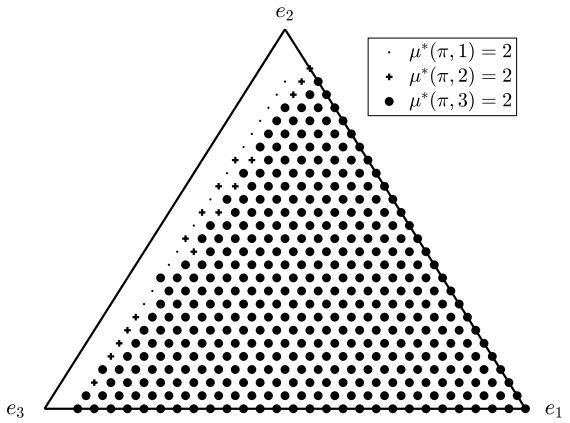
We solved the optimal mining policy for the energy-efficient mining problem in blockchain using the value iteration algorithm (Fig. 7a) and the optimal linear policy using the policy gradient algorithm (Fig. 7b). One can observe that the optimal linear mining policy provides a similar performance as that of the optimal mining policy. Furthermore, the linear policy is suitable for resource-constrained IoT devices: it uses less memory and less computation to compute the optimal action.

We also compared the optimal linear mining policy with other heuristic policies on the real Bitcoin dataset. The results are summarized in Table IIb. We observe that the optimal linear mining policy provides a significant improvement over

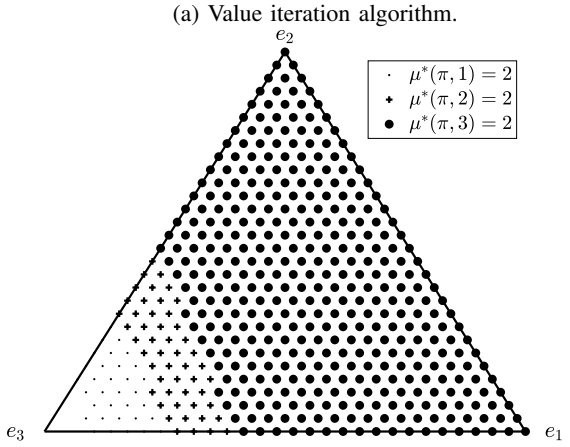
TABLE IV: Model parameters for the energy-efficient mining problem in blockchain for an IoT device (Real Bitcoin dataset)

Parameters	Eq.	Values
$\{\pi_0, \mathcal{X}\}$,	(1)	$\{[0\ 0\ 1], \{1, 2, 3\}\}$
P	(1)	$\begin{bmatrix} 0.8 & 0.2 & 0 \\ 0.038 & 0.8861 & 0.0759 \\ 0 & 0.1111 & 0.8889 \end{bmatrix}$
$\mathcal{Y}$	(2)	$\{1, 2, 3, 4, 5\}$
$B^\intercal$	(2)	$\begin{bmatrix} 0.2384 & 0.1686 & 0.0221 \\ 0.3129 & 0.258 & 0.0955 \\ 0.3951 & 0.3258 & 0.1207 \\ 0.0629 & 0.3 & 0.4546 \\ 0.0044 & 0.0669 & 0.1741 \end{bmatrix}$
$[r(1,2), r(2,2), r(3,2)]$	(3)	[1, 0.125, 0.037]
$[\rho, L]$	(4)	[0.9, 3]
$[\epsilon, \zeta, \kappa, \nu, \psi]$	(12)	[0.5, 0.1, 0.6, 0.6, 0.1]

[6]Although it is beyond the scope of this study, it would be worth studying the loss in optimality due to estimation of the model parameters within the class of TP2 distribution.



(a) Value iteration algorithm.



(b) Policy gradient algorithm.

Fig. 7: Optimal mining policy for energy-efficient mining problem in blockchain on a real Bitcoin dataset. The triangle represents the belief space for the energy-efficient mining problem in blockchain. The belief space has been discretized into 30 equal parts along each axis. The markers indicate the belief states where the optimal action is to mine in the blockchain. The optimal mining policy has a threshold and nested structure (Theorem 1).

other heuristic policies: 57% improvement over the first L mining policy, 39% improvement over the random policy, and 22% improvement over reinforcement learning-based mining policy.

To sum up, we solved the optimal mining policy for an energy-efficient mining problem in a blockchain device using real and synthetic model parameters. Additionally, we conducted numerical experiments to compare the optimal mining policy with other heuristic policies.

V. Conclusion

In this paper, we addressed: what are the optimal times for an IoT device to mine in a blockchain? We formulated the energy-efficient mining problem in blockchain as a multiple stopping time Bayesian sequential detection problem in POMDP. Computing the exact solution of the multiple stopping time POMDP is computationally intractable. So, using submodularity, we derived a useful mathematical structure that characterizes the optimal mining policy: the optimal policy is monotone in the belief state with respect to MLR order and, therefore, has a threshold structure. We exploited these structural results to derive necessary and sufficient conditions on the parameters of an optimal linear mining policy for the energy-efficient mining problem. This optimal linear policy can be computed offline and stored on the IoT device. This makes it suitable for IoT applications. Finally, we illustrated how the proposed multiple stopping time approach achieves energy-efficient mining in blockchain on synthetic data and a real Bitcoin dataset. We also studied the benefit of the optimal mining policy over other heuristic strategies for a blockchain-enabled IoT device.

Acknowledgement This research was supported in part by the U.S. Army Research Office grant W911NF-21-1-0093 and National Science Foundation grant CCF-2112457.

References

[1] A. Sunyaev. Distributed ledger technology. *Internet computing: Principles of distributed systems and emerging internet-based technologies*, pages 265–299, 2020.

[2] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Decentralized business review*, page 21260, 2008.

[3] S. Misra, A. Mukherjee, A. Roy, N. Saurabh, Y. Rahulamathavan, and M. Rajarajan. Blockchain at the edge: Performance of resource-constrained IoT networks. *IEEE Transactions on Parallel and Distributed Systems*, 32(1):174–183, 2021.

[4] O. Novo. Blockchain meets IoT: An architecture for scalable access management in IoT. *IEEE internet of things journal*, 5(2):1184–1195, 2018.

[5] J. Bradshaw. Using blockchain technology to increase transparency in agriculture, Jan 2023.

[6] E. Kovalenko. How can the blockchain secure IoT networks?, Feb 2023.

[7] L. Badea and M. C. Mungiu-Pupazan. The economic and environmental impact of Bitcoin. *IEEE Access*, 9:48091–48104, 2021.

[8] V. Krishnamurthy. *Partially Observed Markov Decision Processes: From Filtering to Controlled Sensing*. Cambridge University Press, 2016.

[9] J. N. T. Dimitri P. Bertsekas. *Neuro-Dynamic Programming*. Optimization and neural computation series. Athena Scientific, 1 edition, 1996.

[10] A. Z. Abyaneh, N. Zorba, and B. Hamdaoui. Empowering next-generation IoT wlans through blockchain and 802.11ax technologies. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–10, 2022.

[11] N. Kullig, P. Lämmel, and N. Tcholtchev. Prototype implementation and evaluation of a blockchain component on IoT devices. *Procedia Computer Science*, 175:379–386, 2020. The 17th International Conference on Mobile Systems and Pervasive Computing (MobiSPC),The 15th International Conference on Future Networks and Communications (FNC),The 10th International Conference on Sustainable Energy Information Technology.

[12] X.-S. Song, Q.-L. Li, Y.-X. Chang, and C. Zhan. A Markov process theory for network growth processes of DAG-based blockchain systems. *arXiv preprint arXiv:2209.01458*, 2022.

[13] Y. Liu, S. Zhang, X. Chen, X. Zhou, and X. Zheng. *Blockchain Security Analysis: A POMDP-Based Approach for Analyzing Blockchain System Security Against the Long Delay Attack*. Eliva Press, 2020.

[14] K. Kim, S.-Y. T. Lee, and S. Assar. The dynamics of cryptocurrency market behavior: sentiment analysis using Markov chains. *Industrial Management & Data Systems*, 122(2):365–395, 2022.

[15] R. Singh, A. D. Dwivedi, G. Srivastava, A. Wiszniewska-Matyszkiel, and X. Cheng. A game theoretic analysis of resource mining in blockchain. *Cluster Computing*, 23:2035–2046, 2020.

[16] G. Yang, Y. Wang, Z. Wang, Y. Tian, X. Yu, and S. Li. Ipbsm: An optimal bribery selfish mining in the presence of intelligent and pure attackers. *International Journal of Intelligent Systems*, 35(11):1735–1748, 2020.

[17] T. Wang, S. C. Liew, and S. Zhang. When blockchain meets AI: Optimal mining strategy achieved by machine learning. *International Journal of Intelligent Systems*, 36(5):2183–2207, 2021.

[18] A. Kiayias, E. Koutsoupias, M. Kyropoulou, and Y. Tselekounis. Blockchain mining games. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 365–382, 2016.

[19] Y. Zhang, M. Liu, J. Guo, Z. Wang, Y. Wang, T. Liang, and S. K. Singh. Optimal revenue analysis of the stubborn mining based on Markov decision process. In *International Conference on Machine Learning for Cyber Security*, pages 299–308. Springer, 2023.

[20] M. Rosenfeld. Analysis of Bitcoin pooled mining reward systems. *arXiv preprint arXiv:1112.4980*, 2011.

[21] W. Mao, Z. Zhao, Z. Chang, G. Min, and W. Gao. Energy-efficient industrial internet of things: Overview and open issues. *IEEE Transactions on Industrial Informatics*, 17(11):7225–7237, 2021.

[22] C. Savaglio, P. Gerace, G. Di Fatta, and G. Fortino. Data mining at the IoT edge. In *2019 28th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–6, 2019.

[23] V. Krishnamurthy, A. Aprem, and S. Bhatt. Multiple stopping time POMDPs: Structural results & application in interactive advertising on social media. *Automatica*, 95:385–398, 2018.

[24] T. Bozkus and U. Mitra. Link analysis for solving multiple-access MDPs with large state spaces. *IEEE Transactions on Signal Processing*, 71:947–962, 2023.

[25] V. Krishnamurthy and C. Rojas. Reduced complexity HMM filtering with stochastic dominance bounds: A convex optimization approach. *IEEE Transactions on Signal Processing*, 62(23):6309–6322, 2014.

[26] K. Hammar and R. Stadler. Intrusion prevention through optimal stopping. *IEEE Transactions on Network and Service Management*, 19(3):2333–2348, 2022.

[27] D.-S. Zois and U. Mitra. Active state tracking with sensing costs: Analysis of two-states and methods for n-states. *IEEE Transactions on Signal Processing*, 65(11):2828–2843, 2017.

[28] G. Atia, V. Veeravalli, and J. Fuemmeler. Sensor scheduling for energy-efficient target tracking in sensor networks. *IEEE Transactions on Signal Processing*, 59(10):4923–4937, 2011.

[29] G. Rovatsos, X. Jiang, A. D. Domínguez-García, and V. V. Veeravalli. Statistical power system line outage detection under transient dynamics. *IEEE Transactions on Signal Processing*, 65(11):2787–2797, 2017.

[30] N. Houy. The bitcoin mining game. *Available at SSRN 2407834*, 2014.

[31] N. Dimitri. Bitcoin mining as a contest. *Ledger*, 2:31–37, 2017.

[32] N. A. Urpi, S. Curi, and A. Krause. Risk-averse offline reinforcement learning. *arXiv preprint arXiv:2102.05371*, 2021.

[33] D. Fakhri and K. Mutijarsa. Secure IoT communication using blockchain technology. In *2018 international symposium on electronics and smart devices (ISESD)*, pages 1–6. IEEE, 2018.

[34] C. Dannen. *Introducing Ethereum and solidity*, volume 1. Springer, 2017.

[35] Blockchain charts, 2023.

[36] A. Muller and D. Stoyan. *Comparison Methods for Stochastic Models and Risk*. Wiley, 2002.

[37] W. Anwaar and M. A. Shah. Energy efficient computing: A comparison of Raspberry pi with modern devices. *Energy*, 4(02), 2015.

[38] J. C. Spall. *Introduction to stochastic search and optimization: estimation, simulation, and control*. John Wiley & Sons, 2005.

[39] J. Soria, J. Moya, and A. Mohazab. Optimal mining in proof-of-work blockchain protocols. *Finance Research Letters*, 53:103610, 2023.

[40] D. M. Topkis. *Supermodularity and Complementarity*. Princeton University Press, 2011.

SUPPLEMENTARY MATERIAL

VI. PROOF OF THEOREM 1 IN SEC.III

The value iteration algorithm is an iterative approach to solve Bellman's equation. However, in this paper, we use the value iteration algorithm to prove by mathematical induction that the optimal policy has a threshold structure. The value iteration algorithm is as follows:

$$\begin{aligned}
V_{k+1}(\pi,l) &= \max_{u\in\{1,2\}} Q_{k+1}(\pi,l,u)\\
\mu_{k+1}(\pi,l) &= \arg\max_{u\in\{1,2\}} Q_{k+1}(\pi,l,u),\\
Q_{k+1}(\pi,l,2) &= r^{\intercal}\pi + \rho\sum_{y} V_k(T(\pi,y),l+1)\sigma(\pi,y)\\
Q_{k+1}(\pi,l,1) &= \rho\sum_{y} V_k(T(\pi,y),l)\sigma(\pi,y)
\end{aligned}$$

Here, $r := [r(x_1,2)\ \ r(x_2,2)\ \ \dots\ \ r(x_{|\mathcal{X}|},2)]$ is the reward vector when the IoT device decides to mine. Define $W_k(\pi,l) := V_k(\pi,l) - V_k(\pi,l+1)$. The mine set M_k^l and don't mine set D_k^l to decide l^{th} mining time instant at iteration k of the value iteration algorithm is defined as:

$$\begin{aligned}
M_{k+1}^l &= \{\pi | r^{\intercal}\pi \geq \rho\sum_{y} W_k(T(\pi,y),l)\sigma(\pi,y)\}\\
D_{k+1}^l &= \{\pi | r^{\intercal}\pi < \rho\sum_{y} W_k(T(\pi,y),l)\sigma(\pi,y)\}
\end{aligned} \tag{13}$$

Our main result uses the following Lemmas from [23].

Lemma 1. *$W_k(\pi,l) \leq W_k(\pi,l+1)$, and $M_k^l \supset M_k^{l+1}$*

Proof. The proof is by forward-induction over k and backward-induction over l, i.e., we assume that the lemma holds for $l+1,\dots,L$ for all values of k, and upto k for l. The base case for mathematical induction is $W_0(\cdot,\cdot) = 0$, $M_k^{L+1} = \phi$, $\forall k$. The induction step is as follows. Note that

$$\begin{aligned}
W_k(\pi,l) = &\left[\rho\sum_{y} W_{k-1}(T(\pi,y),l)\sigma(\pi,y)\right]\mathcal{I}_{D_k^l}(\pi)\\
&+ r^{\intercal}\pi\,\mathcal{I}_{D_k^{l+1}\cap M_k^l}(\pi)\\
&+ \left[\rho\sum_{y} W_{k-1}(T(\pi,y),l+1)\sigma(\pi,y)\right]\mathcal{I}_{M_k^{l+1}}(\pi)
\end{aligned}$$

Consider the following cases:
(a) $\pi \in M_k^{l+2}$. This implies $\pi \in M_k^{l+1}, M_k^l$ and $\pi \notin D_k^{l+2}, D_k^{l+1}, D_k^l$.

$$W_{k+1}(\pi,l) - W_{k+1}(\pi,l+1) =$$
$$\rho\sum_{y}\left(W_k(T(\pi,y),l+1) - W_k(T(\pi,y),l+1)\right)\sigma(\pi,y) \leq 0$$

(b) $\pi \in M_k^{l+1}\cap D_k^{l+2}$. This implies $\pi \in M_k^l$ and $\pi \notin M_k^{l+2}, D_k^{l+1}, D_k^l$.

$$W_{k+1}(\pi,l) - W_{k+1}(\pi,l+1) =$$
$$\rho\sum_{y}\left(W_k(T(\pi,y),l+1)\right)\sigma(\pi,y) - r^{\intercal}\pi \leq 0$$

Last inequality holds using $\pi \in M_k^{l+1}$ and (13)
(c) $\pi \in M_k^l \cap D_k^{l+1}$. This implies $\pi \in D_k^{l+2}$ and $\pi \notin M_k^{l+2}, M_k^{l+1}, D_k^l$.

$$W_{k+1}(\pi,l) - W_{k+1}(\pi,l+1) =$$
$$\rho\sum_{y}\left(W_k(T(\pi,y),l)\right)\sigma(\pi,y) - r^{\intercal}\pi \leq 0$$

Last inequality holds using $\pi \in D_k^{l+1}$ and (13)
(d) $\pi \in D_k^l$. This implies $\pi \in D_k^{l+1}, D_k^{l+2}$ and $\pi \notin M_k^{l+2}, M_k^{l+1}, M_k^l$.

$$W_{k+1}(\pi,l) - W_{k+1}(\pi,l+1) =$$
$$\rho\sum_{y}\left(W_k(T(\pi,y),l) - W_k(T(\pi,y),l)\right)\sigma(\pi,y) \leq 0$$

Thus we have showed that $W_{k+1}(\pi,l) \leq W_{k+1}(\pi,l+1)$. From the definition of the continue and stopping sets in (13), it then follows that $M_{k+1}^l \supset M_{k+1}^{l+1}$.

□

Next, we define a submodular function. This is required to prove that the mining policy is monotone.

Definition 3 (Submodular function)**.** *A function $f(\pi,u)$ is submodular if $f(\pi,u) - f(\pi,\bar{u}) \geq f(\bar{\pi},u) - f(\bar{\pi},\bar{u})$ for $u \geq \bar{u}, \pi \geq_r \bar{\pi}$.*

To show the existence of an optimal policy which is decreasing in $\pi \in \mathcal{L}(e_i,\bar{\pi}),\ i \in \{1,L\}$ (Theorem 1.A), we use the following result on submodular functions from [40]:

Theorem 3. *If $f(\pi,u)$ is submodular, then there exists an optimal policy $u^*(\pi) = \arg\max_{u\in\mathcal{U}}(\pi,u)$ that is MLR decreasing in π.*

We use the MLR order for π in the set $\mathcal{L}(e_i,\bar{\pi}),\ i \in \{1,L\}$. Hence, all we need to show is that $Q(\pi,l,u)$ is a submodular function. Before deriving the main result, we present a property of the Bayesian update from [8]: MLR ordering is preserved under Bayes' rule.

Theorem 4. *If the transition matrix P and the observation matrix B are TP2. Then, for $\pi_1 \geq_r \pi_2$, $T(\pi_1,\cdot) \geq T(\pi_2,\cdot)$ and $\sigma(\pi_1,\cdot) \geq \sigma(\pi_2,\cdot)$.*

The following result guarantees the existence of a monotonically decreasing mining policy.

Theorem 5. *$Q(\pi,l,u)$ is a submodular function $\forall l$.*

Proof. Since, $Q_{k+1}(\pi,l,2) - Q_{k+1}(\pi,l,1) = -\rho\sum_y W_k(T(\pi,y),l)\sigma(\pi,y) + r^{\intercal}\pi$. We need to show that $-\rho\sum_y W_k(T(\pi,y),l)\sigma(\pi,y) + r^{\intercal}\pi$ is MLR decreasing in π.

$$\begin{aligned}
&-\rho\textstyle\sum_y W_k(T(\pi,y),l)\sigma(\pi,y) + r^{\intercal}\pi\\
&= -\textstyle\sum_y\left(\left(\rho W_k(T(\pi,y),l) + \rho r^{\intercal}T(\pi,y)\right)\right.\\
&\left.+\left(r^{\intercal}\pi - \rho r^{\intercal}T(\pi,y)\right)\right)\sigma(\pi,y)\\
&= -\rho\textstyle\sum_y\left(W_k(T(\pi,y),l) - r^{\intercal}T(\pi,y)\right)\sigma(\pi,y)\\
&+ r^{\intercal}\left(I - \rho P^{\intercal}\right)\pi
\end{aligned}$$

The term $r^{\intercal}\left(I-\rho P^{\intercal}\right)\pi$ in is MLR decreasing in π due to our assumptions in Sec.III-A. Hence, to show that

$-\rho\sum_y W_k(T(\pi,y),l)\sigma(\pi,y)+r^\intercal\pi$ is MLR decreasing in π it is sufficient to show that $W_k(\pi,l)-r^\intercal\pi$ is MLR decreasing in π. Define, $\overline{W}_k(\pi,l):=W_k(\pi,l)-r^\intercal\pi$.

$$V_k(\pi,l)=\left(r^\intercal\pi+\rho\sum_y V_{k-1}(T(\pi,y),l-1)\sigma(\pi,y)\right)\mathcal{I}_{M_k^l}+\left(\rho\sum_y V_{k-1}(T(\pi,y),l)\sigma(\pi,y)\right)\mathcal{I}_{D_k^l}$$

where $\mathcal{I}_{D_k^l}$ and $\mathcal{I}_{M_k^l}$ are indicator functions on the don't mine and mine sets, respectively, for each iteration k. As $M_k^{l+1}\subset M_k^l$,

$$\begin{aligned}W_k(\pi,l)&=\left(\rho\sum_y W_{k-1}(T(\pi,y),l)\sigma(\pi,y)\right)\mathcal{I}_{D_k^l}(\pi)\\&\quad+r^\intercal\pi\mathcal{I}_{D_k^{l+1}\cap M_k^l}(\pi)\\&\quad+\left(\rho\sum_y W_{k-1}(T(\pi,y),l+1)\sigma(\pi,y)\right)\mathcal{I}_{M_k^{l+1}(\pi)}\\\Rightarrow\overline{W}_k(\pi,l)&=\sum_y\widetilde{W}_{k-1}(T(\pi,y),l)\mathcal{I}_{D_k^l}(\pi)\\&\quad+\sum_y\widetilde{W}_{k-1}(T(\pi,y),l+1)\mathcal{I}_{M_k^{l+1}}(\pi)\\\widetilde{W}_k(\pi,l)&:=\rho\overline{W}_k(\pi,l)\sigma(\pi,y)-r^\intercal(I-\rho P)^\intercal\pi\end{aligned}$$

We prove using induction that $\overline{W}_k(\pi,l)$ is MLR increasing in π, using the recursive relation over k. For $k=0,\overline{W}_0(\pi,l)=W_0(\pi,l)-r^\intercal\pi=V_0(\pi,l)-V_0(\pi,l+1)-r^\intercal\pi$. The initial conditions of the value iteration algorithm can be chosen such that $\overline{W}_0(\pi,l)$ is increasing in π.

Next, we show that $\overline{W}_k(\pi,l)$ is MLR increasing in π, if $\overline{W}_{k-1}(\pi,l)$ is MLR increasing in π. For $\pi_1\geq_r\pi_2$, consider the following cases: (a) $\pi_1,\pi_2\in M_k^{l+1}$, (b) $\pi_2\in M_k^{l+1},\pi_1\in D_k^{l+1}$, (c) $\pi_1,\pi_2\in D_k^{l+1}\cap M_k^l$, (d) $\pi_2\in D_k^{l+1}\cap M_k^l,\pi_1\in D_k^l$, (e) $\pi_1,\pi_2\in D_k^l$. For cases (a), (e), $\overline{W}_k(\pi_1,l)\geq\overline{W}_k(\pi_2,l)$ by the induction assumption. For case (b), $\overline{W}_k(\pi_1,l)\geq\overline{W}_k(\pi_2,l)$ by definition of M_k^{l+1} and D_k^{l+1} (13). For case (c), $\overline{W}_k(\pi_1,l)=\overline{W}_k(\pi_2,l)=0$. For case (d), $\overline{W}_k(\pi_1,l)\geq\overline{W}_k(\pi_2,l)$ by definition of M_k^l and D_k^l. □

Theorem 1.B follows from the monotonicity property. Suppose M^l is a disconnected set. We can find a line $\mathcal{L}(e_1,\bar{\pi}),\ i\in\{1,L\}$ that passes through the two disconnected components of M^l. Existence of disconnected set would imply that $u^*(\pi)$ is not monotonic in the set $\mathcal{L}(e_i,\bar{\pi}),\ i\in\{1,L\}$ with respect to the MLR order. This is a contradiction.

The proof of Theorem 1.C follows from Lemma 1.

VII. Proof of Theorem 2 in Sec.IV

D^l is non-empty implies $e_{|\mathcal{X}|-1}$ should be in D^l. This implies $\theta_l(1)\geq 0,\forall l$. We first derive conditions on $\begin{bmatrix}\theta_l & 1 & 0\end{bmatrix}\begin{bmatrix}-1\\ \pi\end{bmatrix}$ to be MLR decreasing on $\mathcal{L}(e_i,\bar{\pi}),i\in\{1,|\mathcal{X}|\}$. For $\pi_1\geq_r\pi_2$, $\begin{bmatrix}\theta_l & 1 & 0\end{bmatrix}\begin{bmatrix}-1\\ \pi_1\end{bmatrix}\leq\begin{bmatrix}\theta_l & 1 & 0\end{bmatrix}\begin{bmatrix}-1\\ \pi_2\end{bmatrix}\Leftrightarrow[\theta(2)\ \dots\ \theta(|\mathcal{X}|-1)\ 1\ 0](\pi_1-\pi_2)\leq 0$. For $\pi_1,\pi_2\in\mathcal{L}(e_{|\mathcal{X}|},\bar{\pi})$, $[\theta(2)\ \dots\ \theta(|\mathcal{X}|-1)\ 1\ 0](e_{|\mathcal{X}|}-\bar{\pi})\leq 0\Leftrightarrow[0\ \theta(2)\ \dots\ \theta(|\mathcal{X}|-1)\ 1\ 0]\,\bar{\pi}\geq 0$. For $\pi_1,\pi_2\in\mathcal{L}(e_1,\bar{\pi})$, $[\theta(2)\ \dots\ \theta(|\mathcal{X}|-1)\ 1\ 0](e_1-\bar{\pi})\geq 0\Leftrightarrow\theta(2)-[\theta(2)\ \dots\ \theta(|\mathcal{X}|-1)\ 1\ 0]\,\bar{\pi}\geq 0$

The last condition on the parameters can be derived by using the nested property of the mining set M^l, $\mu_\theta(\pi,l)\geq\mu_\theta(\pi,l+1)\Leftrightarrow\begin{bmatrix}\theta_l-\theta_{l+1} & 1 & 0\end{bmatrix}\begin{bmatrix}-1\\ \pi\end{bmatrix}\geq 0$