MDP Analysis for Blockchain

Roi Bar Zur Ittay Eyal Aviv Tamar

Abstract

TODO: Add abstract

1 Introduction

1.1

TODO: Intro

1.2 Contributions

The contributions of the paper are as follows:

1. Designing a novel method to solve Blockchain protocols modeled as MDPs
2. Proved the new method converges to the optimal solution and bounded the approximation error.
3. Recreated the approximately optimal results of [SSZ16] for Bitcoin.
4. Surpassed the results of [HZJ+19] for Ethereum.
5. Analyzed the new method's running time and showed it is more efficient than the current state of the art method ([SSZ16]).
6. Empirical analysis of the trade-off between running time and approximation error.

TODO: Add navigation paragraph

2 Related Work

Eyal and Sirer were the first to show that the Bitcoin protocol is not incentive compatible ([ES13]). They have demonstrated and analyzed a strategy (called selfish mining or SM1) which can be used by miners and yields higher rewards than acting honestly. Selfish mining involves withholding newly minted blocks and violating the longest chain rule. The existence of such a strategy rules out the honest protocol as a Nash equilibrium.

The analysis in [ES13] gives a closed form formula of the revenue a miner can get if she uses SM1 and the minimal threshold for the relative computation power an attacker need such that using SM1 is superior to following the protocol. However, the analysis does not guarantee that an attacker with a relative power lower than the threshold cannot deviate in some other way and profit. This means that the SM1 threshold is an upper bound on the security threshold of the protocol. Overall, this analysis implies that under the reasonable assumption of $\gamma = 0.5$ the security threshold of Bitcoin is 0.25 at most.

Sapirshtein et al. modeled the Bitcoin protocol as an MDP (Markov Decision Process, described in 3.2) and solved the MDP in order to obtain approximately optimal strategies ([SSZ16]). Their analysis yields an accurate approximation of the security threshold of Bitcoin. For example, they found that for some parameters, the threshold is as low as 0.231.

Other works have tried to generalize the SM1 strategy to Ethereum and analyze its revenue in order to get an upper bound on the security threshold of Ethereum ([RZ18], [FN19], [GPM19]). [RZ18] used a Monte-Carlo simulation in order to assess the revenue of their generalized SM1 strategy while [FN19] and [GPM19] used a theoretical analysis. Nevertheless, all of these works analyze a specific strategy and therefore can only strive to obtain an upper bound of the security threshold.

A recent work by Charlie Hou, Mingxun Zhou and others demonstrates a new technique (called SquirRL) to analyze blockchain protocols ([HZJ+19]). This method is based on the method of [SSZ16] but instead of using MDPs in they use Q-learning. This is a similar method which allows using neural networks. SquirRL was used for Bitcoin and Ethereum and obtained a better upper bound for the security threshold of Ethereum. By observing the results for bitcoin and comparing to [SSZ16], they deduce that their method typically finds solutions with revenues approximately 1-2% lower than the optimum. Due to the approximation in their method, the threshold they find is not guaranteed to be approximately correct but only an upper bound.

3 Name TBD

TODO: Think of a name for the method (Random stop etc.)
Random Stop
Slot Machine
Ratio Reformation
Denominator Ironing/Smoothing

3.1 Motivation

Sapirshtein et al. ([SSZ16]) modeled the Bitcoin protocol as an MDP with an objective function of the form:

$$REV = \mathbb{E}\left[\lim_{T\to\infty} \frac{\sum_{t=1}^{T} R_t^1}{\sum_{t=1}^{T} R_t^1 + R_t^2}\right]$$

Where R_t^1 is the attacker's reward at step t and R_t^2 is the reward of the rest of the network at step t.

This form is not standard since it is not linear in the reward at each stage. In order to overcome this issue they show a novel technique of transforming this MDP to a standard linear reward MDP of the form:

$$REV(\rho) = \mathbb{E}\left[\lim_{T\to\infty} \sum_{t=1}^{T} (1-\rho)\cdot R_t^1 + \rho \cdot R_t^2\right]$$

Their method involves a binary search over $\rho \in [0,1]$ and solving the transformed MDP for the a given ρ at every stage. This method yields a close approximation but is computationally expensive due to the binary search.

This paper describes an alternative way to transform the MDP to a linear form which does not require a binary search at all. Using the new method, solving the MDP only once suffices.

3.2 Markov Decision Process

A Markov decision process (MDP) is a type of a controlled stochastic process [Ber95]. The process is comprised of an agent (in this paper, also known as the attacker) who has a set of actions $\mathcal{A}$ and the environment which has a set of states $\mathcal{S}$. The game goes as follows: at every step t, the agent observes the environment's state $s \in \mathcal{S}$ and decides to take some action $a \in \mathcal{A}$. Then, the environment transitions stochastically to a new state $s' \in \mathcal{S}$, the agent is awarded some $R_t = R_a(s, s')$ and the next step is played.

The transitions between states are Markovian, meaning that the distribution of the next state depends only on the current state and the agent's action. Formally:

$$\Pr\left(s_{i+1} = s' | s_i = s, a_i = a\right) = P_a(s, s')$$

where s_i and a_i are the state and chosen action at step i respectively.

The agent's policy is a mapping from the state space to the action space which given a state of the process, gives the action the agent takes. An MDP with a specific policy induces a Markov chain since at each state the action taken is determined by the policy. The agent wishes to maximize an objective

function of some form linear in the step rewards. A relevant standard objective is the discounted reward criterion:

$$R = \mathbb{E}\left[\lim_{T\to\infty} \sum_{t=0}^{T} \beta^t R_t\right]$$

Note that $\mathcal{S}, \mathcal{A}$ and the functions $P_a(s, s'), R_a(s, s')$ are fully known to the agent.

There are standard algorithms for solving MDPs such as value Iteration and Q-learning. Both methods find an approximately optimal policy given an MDP and converge to the optimum as they are allowed to run more time. Value iteration requires $P_a(s, s')$ and $R_a(s, s')$ explicitly. Q-learning is somewhat similar to value iteration but instead of requiring $P_a(s, s')$ and $R_a(s, s')$ explicitly it learns them. In addition, Q-Learning can work with a large state space by using function approximation, e.g., neural networks.

3.3 Method

Given an MDP of a blockchain protocol, w.l.o.g, the agent's objective function which she wants to maximize can be defined as:

$$REV = \mathbb{E}\left[\lim_{T\to\infty} \frac{\sum_{t=1}^{T} R_t}{\sum_{t=1}^{T} D_t}\right]$$

Where R_t is the reward the miner receives in step t and D_t is the contribution of the step t towards the difficulty adjustment.

In Bitcoin for example, the difficulty contribution is the sum of rewards all parties receive. Thus REV_a is equivalent to the relative revenue of the miner. In Ethereum, the difficulty contribution in step t is the amount of the amount of blocks added to the main chain and as uncles. This is since in Ethereum uncle blocks are also considered for the difficulty adjustment. This target function is non-linear in the rewards and therefore standard MDP solution methods will not work.

The method of solving an MDP of this form consists of two stages:

1. Constructing an auxiliary MDP (marked by $\text{MDP}_{\text{RandStop}}$) based on the original MDP (marked by MDP_{Org}). $\text{MDP}_{\text{RandStop}}$ has the same state space MDP_{Org} and is essentially a duplicate of MDP_{Org}. But $\text{MDP}_{\text{RandStop}}$ is constructed such that every transition in it has some chance to end the process and move to a finite state. If this does not happen then the transition occurs as in MDP_{Org}. At every transition, the chance to end abruptly is $(1 - \frac{1}{D})^{D_t}$ where D is some chosen parameter.
   The objective function of $\text{MDP}_{\text{RandStop}}$ is:

$$REV_{\text{RandStop}} = \mathbb{E}\left[\frac{1}{D} \sum_{t=0}^{\text{Term}} R_t\right]$$

where Term is a random variable indicating the step in which the process terminates. This is practically a discounted reward with $\beta = 1$ and setting $R_t = 0$ for all $t >$ Term.

2. Solving $\text{MDP}_{\text{RandStop}}$ using standard MDP solver algorithms (e.g. Value Iteration) to obtain an approximately optimal policy.

Note that the chance to end the process abruptly depends on D_t and the higher D_t is, the higher the chance to end the process. Intuitively, this encourages consideration of how actions add to the difficulty contribution and finding a balance between the reward added and the difficulty contribution. D is called the expected horizon since this is the expected total difficulty contribution when the process ends (this will be proved formally later on).

Then, in order to analyze the revenue of the policy one has to calculate the steady-state distribution of the Markov chain induced by the policy in MDP_{Org}. Then, use the steady-state probability distribution to calculate the expected reward and expected difficulty contribution of each step. The revenue is then the ratio between the two values.

3.4 Proof of Optimality

Definition 1. *Denote the revenue of the attacker in MDP_{Org} as:*

$$REV_{Org} \triangleq \mathbb{E}\left[\lim_{T\to\infty} \frac{\sum_{t=1}^{T} R_t}{\sum_{t=1}^{T} D_t}\right]$$

Definition 2. *At every step t, X_t is a random variable indicating whether the process $MDP_{RandStop}$ terminated at this step. It is defined as:*

$$X_t \triangleq \begin{cases} 1 & w.p\ \left(1-\frac{1}{D}\right)^{D_t}\ \textit{(continue)} \\ 0 & w.p\ 1-\left(1-\frac{1}{D}\right)^{D_t}\ \textit{(stop)} \end{cases}$$

Definition 3. *The step in which $MDP_{RandStop}$ terminates is defined as:*

$$Term_{RandStop}(D) \triangleq \arg\min_{T}\{X_T = 0\}$$

Definition 4. *The revenue of the attacker in the linear reward MDP we defined is:*

$$REV_{RandStop}(D) \triangleq \lim_{D\to\infty} \mathbb{E}\left[\frac{1}{D} \sum_{t=1}^{Term_{RandStop}(D)} R_t\right]$$

Theorem 1. *Optimizing the original MDP is the same as optimizing the auxiliary MDP constructed by the method.*

$$REV_{Org} = \lim_{D\to\infty} REV_{RandStop}(D)$$

Using the notations of [Whi01], X_n denotes a Markov chain in general. X_n may also denote a random variable of the state of the Markov chain at time n. S denotes the state space of the Markov chain. P denotes the transition matrix.

An irreducible chain is a chain in which for every pair of states i and j there is a chance to transition from i to j after any number of steps.

Definition 5. *(Definition 25 p. 16 [Whi01])*
A hitting time of a state i by a process X_n is defined by:

$$\tau_i = \min\{n \geq 2 : X_1 = X_n = i\}$$

Intuitively, the hitting time of a state is a random variable of the number of steps it takes to enter the state when starting at said state.

A positive recurrent state is a state i for which the expected hitting time is finite.

A period of a state is the gcd of all possible values for its hitting time. A state is aperiodic if its period is 1.

For an irreducible chain, if one state is positive recurrent and aperiodic then all its states are. In addition, the chain is called positive recurrent and aperiodic.

Lemma 2. *(Proposition 69 p. 44 [Whi01])*
Let X_n be an irreducible positive recurrent Markov chain with stationary distribution π. Suppose V_n, $n \geq 1$, are real-valued random variables associated with the chain such that

$$\mathbb{E}\left[V_n | X_1, X_2, ..., X_n\right] = a_{X_n}, \quad n \geq 1$$

where a_j are constants. Then, for the hitting time τ_i of a fixed state i,

$$\mathbb{E}\left[\sum_{n=1}^{\tau_i - 1} V_n \middle| X_1 = i\right] = \frac{1}{\pi_i} \sum_{j \in S} a_j \pi_j$$

provided the last sum is absolutely convergent.

This lemma gives a way to calculate the expected cumulative sum of a random variable which depends on the current state of the chain until some chosen state is entered. For example, in Bitcoin, V_n can be the number of blocks the attacker adds to the chain at step n. This variable clearly depends on the state and its expected value is a function of the state.

Lemma 3. *(Corollary 79 p. 50 [Whi01])*
For a fixed integer ℓ, the process $\tilde{X}_n = (X_n, ..., X_{n+\ell})$ is an ergodic Markov chain on $S^{\ell+1}$ with stationary distribution

$$\pi(i) = \pi_{i_0} p_{i_0, i_1} \cdots p_{i_{\ell-1}, i_\ell}$$

Hence, for $f : S^{\ell+1} \to \mathbb{R}$,

$$\lim_{n\to\infty} n^{-1} \sum_{m=1}^{n} f(\tilde{X}_m) = \sum_{i\in S^{\ell+1}} f(i)\pi(i) \quad a.s.,$$

provided the sum is absolutely convergent.

This lemma is a generalization of Lemma 2. First, instead of a random variable which depends on a single state, it allows using any (deterministic) function of the last $(\ell + 1)$ states. This is a powerful notion since it depends on multiple states instead of one. Second, instead of cumulative sum until entering a said state, it provides the average of the function when the chain plays indefinitely.

Lemma 4. *(Theorem 111 p. 74 [Whi01])*
If X_n *is an ergodic Markov chain with stationary distribution* π*, then*

$$\sup_i |Pr(X_n = i) - \pi_i| \to 0, \quad as\ n \to \infty$$

Intuitively, this means that any row of P^n gets closer to π as $n \to \infty$. This is since for every i, j: $\Pr(X_n = i) = \sum_{j\in S} \Pr(X_1 = j) \cdot (P^n)_{ji} \to \pi_i$. And this is for any choice of distribution for the initial state X_1.

Using the L_1 norm instead of L_∞ and denoting the distribution of X_1 as x, the lemma can be rewritten:

Lemma 5. *If* X_n *is an ergodic Markov chain with stationary distribution* π*, then for any* $\eta \in [0, 1]^{|S|}$ *such that* $||\eta|| = 1$*:*

$$\lim_{n\to\infty} ||\eta P^n - \pi|| = 0$$

From now on, we will assume a fixed policy σ. This means that any MDP we have can be reduced to a Markov chain by applying σ to it.

Definition 6. *The absolute revenue of the attacker in the original MDP (under a specific policy) is:*

$$REV \triangleq \mathbb{E}\left[\lim_{T\to\infty} \frac{\sum_{t=1}^{T} R_t}{\sum_{t=1}^{T} D_t}\right]$$

R_t is the reward of the attacker at step t and D_t is the difficulty contribution at step t. For example, in Bitcoin R_t would be the number of blocks the attacker adds to the main chain at step t and D_t would be the total number of blocks added to the main chain at said step (both blocks of the attacker and blocks of the rest of the network).

We make the easing assumption that at every step the maximal contribution to the difficulty is upper bounded by some constant c. Formally:

Assumption 1. $\forall_t : 0 \leq D_t \leq c$

Reminder of the definition MDP': A replica of the original MDP where we also take every state and give it a chance to transition to the initial state and start over. This is denoted by X_t. The reward and difficulty contribution are the same as in the original. MDP' is determined by some chosen parameter D. Denote the initial state of MDP and MDP' as s_{init}.
Denote $P_\sigma(i,j)$ as the chance to transition to state j when the MDP is in state i under policy σ. Denote $P'_\sigma(i,j)$ similarly for MDP'. Denote their steady-state distributions as π_i and π'_i respectively.
Also denote $R_\sigma(i,j)$ as the reward in MDP under policy σ when transitioning from state i to state j and $D_\sigma(i,j)$ as the difficulty contribution in a similar manner. Note that $R_\sigma(i,j)$ and $D_\sigma(i,j)$ are equivalent in MDP and MDP'.

Definition 7. *At every step t, X_t is a random variable indicating whether MDP' terminated at this step. For some parameter $D \in \mathbb{N}$, it is defined as:*

$$X_t \triangleq \begin{cases} 1 & w.p\ \left(1-\frac{1}{D}\right)^{D_t}\ \textit{(continue)} \\ 0 & w.p\ 1-\left(1-\frac{1}{D}\right)^{D_t}\ \textit{(stop)} \end{cases}$$

Corollary 6. *The transition probability of MDP' is:*

$$P'_\sigma(i,j) = \begin{cases} \left(1-\frac{1}{D}\right)^{D_\sigma(i,j)} P_\sigma(i,j) + 1 - \left(1-\frac{1}{D}\right)^{D_\sigma(i,j)} & j = s_{init} \\ \left(1-\frac{1}{D}\right)^{D_\sigma(i,j)} P_\sigma(i,j) & o.w \end{cases}$$

Definition 8. *The step in which MDP' terminates is defined as:*

$$Term_{RandStop}(D) \triangleq \arg\min_T \{X_T = 0\}$$

This is a random variable which denotes the first step in which "stop" is drawn.

Definition 9. *The absolute revenue of the attacker in MDP' we defined is:*

$$REV_{RandStop}(D) \triangleq \frac{1}{D}\mathbb{E}\left[\sum_{t=1}^{Term_{RandStop}(D)} R_t\right]$$

MDP' is a function of D, any choice of D would give a slightly different MDP'. We are interested at the behavior for a large enough D.

Definition 10. *The limit of the absolute revenue as D approaches ∞ is defines as $REV_{RandStop}$. Formally:*

$$\triangleq \lim_{D\to\infty} REV_{RandStop}(D) = \lim_{D\to\infty} \frac{1}{D}\mathbb{E}\left[\sum_{t=1}^{Term_{RandStop}(D)} R_t\right]$$

Lemma 7. *The expected total contribution to the difficulty when MDP' terminates is asymptotically equivalent to D. Formally:*

$$\lim_{D\to\infty}\frac{1}{D}\mathbb{E}\left[\sum_{t=1}^{Term_{RandStop}(D)} D_t\right]=1$$

This lemma is the reason we call D the expected horizon. Intuitively, For a large enough D, the expected sum of the difficulty contribution from the start of the process until termination is close to D. This also explains the intuition behind the choice of AR' - instead of dividing by the actual difficulty contribution, we divide by the expected difficulty contribution.

Proof. Define:

$$\hat{X}_{ti}\triangleq\begin{cases}1 & \text{w.p } 1-\frac{1}{D}\text{ (continue)}\\ 0 & \text{w.p }\frac{1}{D}\text{ (stop)}\end{cases},t\in\mathbb{N},i\in[c]$$

Notice that from this definition and from definition 7, X_t can be thought of as: $X_t=1\iff\forall_{i=1}^{D_t}\hat{X}_{ti}=1$. Or in other words:

$$X_t=\prod_{i=1}^{D_t}\hat{X}_{ti}\tag{1}$$

This decomposes each X_t to be a function of a set of hidden i.i.d Bernoulli trials. Remember from definition 8:

$$Term_{\text{RandStop}}(D)=\arg\min_T\{X_T=0\}=\arg\min_T\left\{\prod_{t=1}^{T}X_t=0\right\}$$

The last equivalence follows because for all $t<Term_{\text{RandStop}}(D):X_t=1$.
By using (1):

$$Term_{\text{RandStop}}(D)=\arg\min_T\left\{\prod_{t=1}^{T}\prod_{i=1}^{D_t}\hat{X}_{ti}=0\right\}$$

Since all $\hat{X}_{ti}$ are i.i.d we can rename and renumber them as $\hat{Z}_j$ to get:

$$Term_{\text{RandStop}}(D)=\arg\min_T\left\{\prod_{j=1}^{\sum_{t=1}^{T}D_t}\hat{Z}_j=0\right\}$$

Define:

$$G=\arg\min_k\left\{\prod_{j=1}^{k}\hat{Z}_j=0\right\}$$

Notice since all $\hat{Z}_j$ are independent Bernoulli trials (continue means failure and stop means success). G is distributed geometrically with a chance of success $p = \frac{1}{D}$. This yields:

$$\mathbb{E}[G] = \frac{1}{p} = D \tag{2}$$

We will establish a relation between $Term_{\text{RandStop}}(D)$ and G. Notice that by definition of $Term_{\text{RandStop}}(D)$:

$$\prod_{j=1}^{\sum_{t=1}^{Term_{\text{RandStop}}(D)} D_t} \hat{Z}_j = 0 \tag{3}$$

$$\prod_{j=1}^{\sum_{t=1}^{Term_{\text{RandStop}}(D)-1} D_t} \hat{Z}_j = 1 \tag{4}$$

The definition of G and (3) yield:

$$G \leq \sum_{t=1}^{Term_{\text{RandStop}}(D)} D_t$$

And (4) yields:

$$G > \sum_{t=1}^{Term_{\text{RandStop}}(D)-1} D_t \geq \sum_{t=1}^{Term_{\text{RandStop}}(D)} D_t - c$$

The last inequality follows from assumption 1. Overall:

$$G \leq \sum_{t=1}^{Term_{\text{RandStop}}(D)} D_t < G + c$$

$$\mathbb{E}[G] \leq \mathbb{E}\left[\sum_{t=1}^{Term_{\text{RandStop}}(D)} D_t\right] \leq \mathbb{E}[G] + c$$

$$\frac{1}{D}\mathbb{E}[G] \leq \frac{1}{D}\mathbb{E}\left[\sum_{t=1}^{Term_{\text{RandStop}}(D)} D_t\right] \leq \frac{1}{D}(\mathbb{E}[G] + c)$$

$$\lim_{D\to\infty} \frac{1}{D}\mathbb{E}[G] \leq \lim_{D\to\infty} \frac{1}{D}\mathbb{E}\left[\sum_{t=1}^{Term_{\text{RandStop}}(D)} D_t\right] \leq \lim_{D\to\infty} \frac{1}{D}(\mathbb{E}[G] + c)$$

Swapping $\mathbb{E}[G]$ thanks to (2):

$$1 = \lim_{D\to\infty} \frac{1}{D} D \leq \lim_{D\to\infty} \frac{1}{D} \mathbb{E}\left[\sum_{t=1}^{Term_{\text{RandStop}}(D)} D_t\right] \leq \lim_{D\to\infty} \frac{1}{D}(D+c) = 1$$

$$\lim_{D\to\infty} \frac{1}{D} \mathbb{E}\left[\sum_{t=1}^{Term_{\text{RandStop}}(D)} D_t\right] = 1$$

□

The main theorem:

Theorem 8. *The absolute revenue of the MDP is equivalent to the absolute revenue of the MDP' (under the same policy):*

$$REV_{Org} = REV_{RandStop}$$

First we make a few observations on MDP and MDP':

Assumption 2. *MDP and MDP' are irreducible.*

This is true w.l.o.g because we can ignore all the states that are unreachable from s_{init} and any transient states at the beginning.

Lemma 9. *MDP and MDP' are positive recurrent.*

Proof. Since both are irreducible and s_{init} is positive recurrent - in every state there is a chance to return to s_{init} eventually. In MDP this is because $\alpha < 0.5$, the honest network will always catch up and the attacker cannot keep waiting forever, she must perform adopt at some point. In MDP' there is additionally a chance to go back to s_{init}.

□

Assumption 3. *MDP and MDP' are aperiodic.*

We can say that both MDP and MDP' are aperiodic w.l.o.g because we can take any positive recurrent state and change it to have a chance to transition to itself with no reward and no difficulty contribution. This does not change AR or AR' at all since this just means that the game may halt for a few steps and then carry on normally. But, this change ensures the state has a period of 1 and with irreducibly this implies aperiodicity.

Corollary 10. *MDP and MDP' are ergodic.*

This quickly follows from lemma 9 and assumptions 2 and 3.

Proof. (of the main theorem)
Assume policy σ.

Define the expected reward and difficulty contribution of every state, for MDP and MDP':

$$\hat{R}_i = \sum_{j \in S} R_\sigma(i,j) P_\sigma(i,j)$$

$$\hat{R}'_i = \sum_{j \in S} R_\sigma(i,j) P'_\sigma(i,j)$$

$$\hat{D}_i = \sum_{j \in S} D_\sigma(i,j) P_\sigma(i,j)$$

$$\hat{D}'_i = \sum_{j \in S} D_\sigma(i,j) P'_\sigma(i,j)$$

These are the expected reward/difficulty contribution in MDP/MDP' when transitioning from state i.
Remember from definition 6:

$$REV_{\text{Org}} = \mathbb{E}\left[\lim_{T\to\infty} \frac{\sum_{t=1}^{T} R_t}{\sum_{t=1}^{T} D_t}\right]$$

Let's analyze this expression first:

$$\lim_{T\to\infty} \frac{\sum_{t=1}^{T} R_t}{\sum_{t=1}^{T} D_t} = \frac{\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} R_t}{\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} D_t} = \frac{\sum_{(i,j)\in S^2} R_\sigma(i,j)\pi_i P_\sigma(i,j)}{\sum_{(i,j)\in S^2} D_\sigma(i,j)\pi_i P_\sigma(i,j)}$$

The first equality is true only if both limits are well defined and the denominator is not 0. We will see that this is indeed the case after fully developing the expression.
The last equality follows from lemma 3 with $\ell = 1$ when choosing
$f : (X_t, X_{t+1}) \mapsto R_\sigma(X_t, X_{t+1})$ for the nominator and
$f : (X_t, X_{t+1}) \mapsto D_\sigma(X_t, X_{t+1})$ for the denominator.
By taking the sums apart we get:

$$\frac{\sum_{(i,j)\in S^2} R_\sigma(i,j)\pi_i P_\sigma(i,j)}{\sum_{(i,j)\in S^2} D_\sigma(i,j)\pi_i P_\sigma(i,j)} = \frac{\sum_{i\in s}\sum_{j\in s} R_\sigma(i,j)\pi_i P_\sigma(i,j)}{\sum_{i\in s}\sum_{j\in s} D_\sigma(i,j)\pi_i P_\sigma(i,j)} = \frac{\sum_{i\in s} \hat{R}_i \pi_i}{\sum_{i\in s} \hat{D}_i \pi_i}$$

We get that the last expression is not random so its expectation is itself. Overall:

$$REV_{\text{Org}} = \mathbb{E}\left[\lim_{T\to\infty} \frac{\sum_{t=1}^{T} R_t}{\sum_{t=1}^{T} D_t}\right] = \mathbb{E}\left[\frac{\sum_{i\in s} \hat{R}_i \pi_i}{\sum_{i\in s} \hat{D}_i \pi_i}\right] = \frac{\sum_{i\in s} \hat{R}_i \pi_i}{\sum_{i\in s} \hat{D}_i \pi_i}$$

The last equality follows from the fact that the expression in the expectation is constant.
From definition 10 and from lemma 7:

$$REV_{\text{RandStop}} = \lim_{D\to\infty} \frac{1}{D}\mathbb{E}\left[\sum_{t=1}^{Term_{\text{RandStop}}(D)} R_t\right] = \frac{\lim_{D\to\infty} \frac{1}{D}\mathbb{E}\left[\sum_{t=1}^{Term_{\text{RandStop}}(D)} R_t\right]}{\lim_{D\to\infty} \frac{1}{D}\mathbb{E}\left[\sum_{t=1}^{Term_{\text{RandStop}}(D)} D_t\right]} =$$

$$= \lim_{D\to\infty} \frac{\mathbb{E}\left[\sum_{t=1}^{Term_{\text{RandStop}}(D)} R_t\right]}{\mathbb{E}\left[\sum_{t=1}^{Term_{\text{RandStop}}(D)} D_t\right]}$$

Note that:

1. $(Term_{\text{RandStop}}(D) + 1)$ is the hitting time of s_{init}.
2. In MDP', $\mathbb{E}\left[R_t | X_t = i\right] = \hat{R}'_i$ and $\mathbb{E}\left[D_t | X_t = i\right] = \hat{D}'_i$,

Therefore lemma 2 can be used (twice) to obtain:

$$REV_{\text{RandStop}} = \lim_{D\to\infty} \frac{\mathbb{E}\left[\sum_{t=1}^{Term_{\text{RandStop}}(D)} R_t\right]}{\mathbb{E}\left[\sum_{t=1}^{Term_{\text{RandStop}}(D)} D_t\right]} = \lim_{D\to\infty} \frac{(\pi'_{s_{\text{init}}})^{-1} \sum_{i\in S} \hat{R}'_i \pi'_i}{(\pi'_{s_{\text{init}}})^{-1} \sum_{i\in S} \hat{D}'_i \pi'_i} = \lim_{D\to\infty} \frac{\sum_{i\in S} \hat{R}'_i \pi'_i}{\sum_{i\in S} \hat{D}'_i \pi'_i}$$

Remember from corollary 6:

$$P'_\sigma(i,j) = \begin{cases} \left(1 - \frac{1}{D}\right)^{D_\sigma(i,j)} P_\sigma(i,j) + 1 - \left(1 - \frac{1}{D}\right)^{D_\sigma(i,j)} & j = s_{\text{init}} \\ \left(1 - \frac{1}{D}\right)^{D_\sigma(i,j)} P_\sigma(i,j) & \text{o.w} \end{cases}$$

Since $P_\sigma(i,j)$ and $D_\sigma(i,j)$ are independent of D it is trivial ot see that:

$$\forall i,j \in S : \lim_{D\to\infty} P'_\sigma(i,j) = P_\sigma(i,j) \tag{5}$$

It quickly follows that:

$$\forall i \in S : \lim_{D\to\infty} \hat{R}'_i = \hat{R}_i$$

$$\forall i \in S : \lim_{D\to\infty} \hat{D}'_i = \hat{D}_i$$

In addition, since π and π' are the stationary distributions of MDP and MDP' respectively, by lemma 5:

$$\forall \eta \in [0,1]^{|S|} \text{s.t } ||\eta|| = 1 : \lim_{n\to\infty} ||\eta P^n - \pi|| = 0$$
$$\lim_{n\to\infty} ||\eta (P')^n - \pi'|| = 0$$

(5) implies:

$$\lim_{D\to\infty} P' = P$$

It is easy to see that for any n, it holds that:

$$\lim_{D\to\infty} (P')^n = P^n$$

Or in another form:

$$\lim_{D\to\infty} ||(P')^n - P^n|| = 0$$

By using the triangle inequality, we get:

$$||\pi' - \pi|| = ||\pi' - \eta(P')^n + \eta(P')^n - \eta P^n + \eta P^n - \pi|| \leq$$
$$\leq ||\pi' - \eta(P')^n|| + ||\eta(P')^n - \eta P^n|| + ||\eta P^n - \pi||$$

Overall:

$$\lim_{D\to\infty} ||\pi' - \pi|| = \lim_{D\to\infty} \lim_{n\to\infty} ||\pi' - \pi|| \leq$$
$$\leq \lim_{D\to\infty} \lim_{n\to\infty} ||\pi' - \eta(P')^n|| + ||\eta(P')^n - \eta P^n|| + ||\eta P^n - \pi|| =$$
$$= \lim_{D\to\infty} \lim_{n\to\infty} ||\eta(P')^n - \eta P^n|| \leq \lim_{D\to\infty} ||\eta(P')^{n_0} - \eta P^{n_0}|| = 0$$

The first equality follows from the fact that π and π' are independent of n.

Using all limits and the fact that the sums are finite (because S is finite), we can write:

$$REV_{\text{RandStop}} = \lim_{D\to\infty} \frac{\sum_{i\in S} \hat{R}'_i \pi'_i}{\sum_{i\in S} \hat{D}'_i \pi'_i} = \frac{\sum_{i\in S} \left(\lim_{D\to\infty} \hat{R}'_i\right)\left(\lim_{D\to\infty} \pi'_i\right)}{\sum_{i\in S} \left(\lim_{D\to\infty} \hat{D}'_i\right)\left(\lim_{D\to\infty} \pi'_i\right)} =$$
$$= \frac{\sum_{i\in s} \hat{R}_i \pi_i}{\sum_{i\in s} \hat{D}_i \pi_i} = REV_{\text{Org}}$$

□

3.5 Approximation Error

Theorem 11. *The approximation error of the method is linear in* $\frac{1}{D}$.

$$|REV_{Org} - REV_{RandStop}(D)| = O(\frac{1}{D})$$

4 Results

4.1 Optimality results

4.1.1 Bitcoin

The analysis in [ES13] gives a closed form formula of the revenue a miner can get if she uses SM1. The formula depends on two parameters which model the attacker:

1. $\alpha \in [0, \frac{1}{2})$ - The attacker's relative computational power in respect to the entire network.
2. $\gamma \in [0, 1]$ - The attacker's network connectivity.

The Bitcoin protocol specifies that in the case of a tie in the longest chain rule, the tie is decided in favor of the first chain seen. This is the reason γ is important. In case the attacker releases a chain of her own, the more connected she is to other parties in the network, the more parties will favor her chain.

The modeling of Bitcoin in [SSZ16] uses these 2 parameters as well. An overview of their results and the results obtained by the new method is in figure 1.

TODO: Run for max fork=95 and update our results

Power (α)	Connectivity (γ)	Revenue (REV)	Revenue in [SSZ16]
0.35	0	0.370773	0.37077
0.4	0	0.488634	0.48863
0.45	0	0.668149	0.66891

Figure 1: The method's results for Bitcoin when limiting the maximum fork length to 95

4.1.2 Ethereum

A model of Ethereum as an MDP is described in [HZJ+19]. A few results obtained by them and by the new method appear in figure 2.

TODO: Run with same alphas as SquirRL with max fork 20 and fix max fork

Power (α)	Revenue (REV)	Revenue in [HZJ+19]
0.2469	0.246912	?
0.247	0.247033	?
0.25	0.250705	?
0.3	0.317798	?
0.35	0.407925	?
0.4	0.534359	?

Figure 2: The method's results for Ethereum when limiting the maximum fork length to 20

Notice that the new threshold found is 0.2469. This is lower than state of the art results which are approximately 0.26 ([RZ18], [FN19]). TODO: Add squirRL corrected threshold

4.2 Running Time Comparison

TODO: Compare method performance for a limited running time Sapirshtein vs. ours

5 Hyperparameters

5.1 Expected Horizon

TODO: Go over this, Graph for expected horizon.

Figure 3 shows the revenue the policy found by the method against the horizon length considered. As the expected horizon increases the revenue converges to the revenue of the optimal policy.

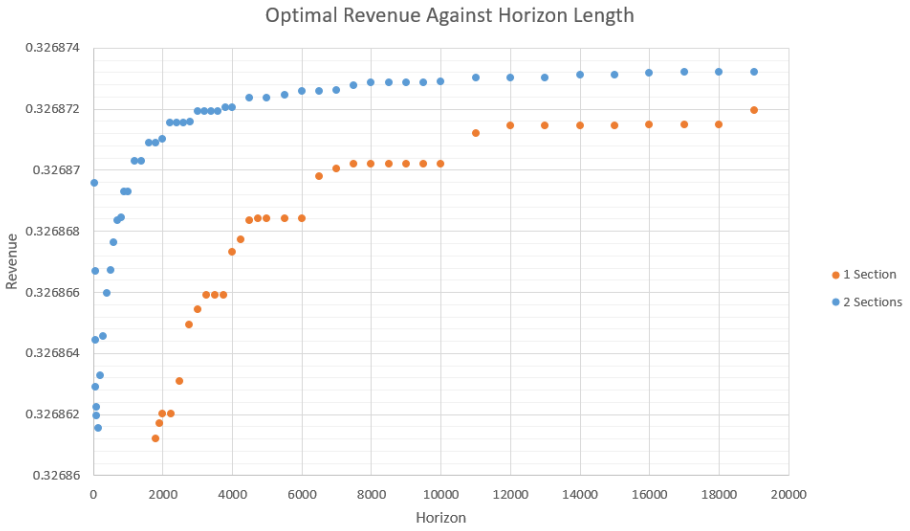

Figure 3: The method's results for Bitcoin with a $\alpha = 0.3, \gamma = 0.5$ and limiting the maximum fork length to 20

5.2 Maximal Fork Length

TODO: Graph for max fork Show that this is negligible for fork ¿ 20

6 Discussion and Future Work

TODO: Add to discussion In addition to finding an approximately optimal policy, Sapirshtein's analysis provided an upper bound to the optimal revenue for any possible policy which takes into consideration policies which allow a maximum fork length of more than what truncation they used for their MDP. However their upper bound is not tight so it does not provide much information since using it for bounding the security threshold wouldn't yield accurate results. In this paper the policy space is assumed to be the same for both the original MDP and the MDP obtained by the method and there is no reference to the actual upper bound when considering policies which allow very long forks. This is ignored since the probability of the attacker obtaining a very large fork declines exponentially with the fork length. Therefore, taking into consideration a large enough maximum fork length such as 100 is enough and increasing it fourther is negligible as shown in section **??**.

SquirRL's method ([HZJ+19]) is based on Sapirshtein's method ([SSZ16]). They use approximated Q-learning instead of using value iteration to solve the MDP at every stage of the binary search. Approximated Q-learning gives much worse approximation than value iteration and is also stochastic. This interferes with the binary search and gives limits the results to be less accurate. Using

the novel method with approximated Q-learning may reduce the error only to the approximation error of the function approximation used instead of both that and the stochastic nature of the method which interferes with the binary search.

7 Conclusion

TODO: Conclusion

References

[Ber95] Dimitri P Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena scientific Belmont, MA, 1995.

[ES13] Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. 2013.

[FN19] Chen Feng and Jianyu Niu. Selfish mining in ethereum. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 1306–1316. IEEE, 2019.

[GPM19] Cyril Grunspan and Ricardo Pérez-Marco. Selfish mining in ethereum. *arXiv preprint arXiv:1904.13330*, 2019.

[HZJ+19] Charlie Hou, Mingxun Zhou, Yan Ji, Phil Daian, Florian Tramer, Giulia Fanti, and Ari Juels. Squirrl: Automating attack discovery on blockchain incentive mechanisms with deep reinforcement learning. *arXiv preprint arXiv:1912.01798*, 2019.

[RZ18] Fabian Ritz and Alf Zugenmaier. The impact of uncle rewards on selfish mining in ethereum. In *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 50–57. IEEE, 2018.

[SSZ16] Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. Optimal selfish mining strategies in bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 515–532. Springer, 2016.

[Whi01] CC White. *Markov decision processes*. Springer, 2001.