The computational hardness of pricing compound options

Mark Braverman*
Princeton University (mbraverm@princeton.edu)

Kanika Pasricha
Princeton University (pasricha@alumni.princeton.edu)

January 9, 2014

Abstract

It is generally assumed that you can make a financial asset out of any underlying event or combination thereof, and then sell a security. We show that while this is theoretically true from the financial engineering perspective, compound securities might be intractable to price. Even given no information asymmetries, or adversarial sellers, it might be computationally intractable to put a value on these, and the associated computational complexity might afford an advantage to the party with more compute power. We prove that the problem of pricing an option *on a single security* with unbounded compounding is PSPACE hard, even when the behavior of the underlying security is computationally tractable. We also show that in the oracle model, even when compounding is limited to at most k layers, the complexity of pricing securities grows exponentially in k.

*Research supported in part by an Alfred P. Sloan Fellowship, an NSF CAREER award (CCF-1149888), NSF Award CCF-1215990, a Turing Centenary Fellowship, and a Packard Fellowships in Science and Engineering.

1 Introduction

A financial security is an instrument representing financial value. Examples of securities include currencies, shares of companies, debt obligations, and commodity contracts. Simple securities generally directly correspond to actual future cash flows, or events. For example, a stock of a company can be viewed as a contract entitling one to the company's dividend payments. A simple oil futures contract entitles one for a given amount of oil to be delivered at a pre-determined time. Simple securities therefore can be viewed as simple contracts that can be traded among agents.

Based on these simple securities one can construct more complicated ones, known as financial derivatives. The value of a derivative may depend on the value of one or multiple securities in a complicated fashion. For example, a *put option* may allow one to sell a stock at a later time at a predetermined price (e.g. the option to sell a share of GOOG for $1000 a year from now), thus realizing a piece-wise linear function on the price of the stock. Collateralized debt obligations (CDOs), which were much publicized during the 2008 financial crisis, represent step functions on bundles of debt obligations. Financial derivatives are created for many different reasons, including risk-hedging and giving market participants greater flexibility in expressing their beliefs through their holdings.

Assigning values to financial securities and derivatives is one of the primary functions of financial markets and institutions. This function is at the heart of the stock market, and the operations of hedge funds who extract surplus by identifying (and helping correct) mispriced financial products. The subject of this paper is the *computational complexity* aspects of the hardness of pricing financial derivatives. The process of valuing securities can be abstracted as the process of (1) obtaining information about the present and the future that is relevant to the present and future value of the security; and (2) performing computations to obtain a price for the security based on the information. In this paper we focus on the computational complexity of performing the second step, and thus abstract away as much of the first step as possible. A common way of representing complex information about the future is through Monte-Carlo simulations. We will assume therefore that we have a model of the relevant future world-states and prices, either as a program or as a black-box oracle, and focus on the hardness of turning this model into the relevant price. Much of economic theory deals with information and uncertainty of the agents, which leads to various interesting phenomena. Here we assume no uncertainty in that the model available to the pricer is exhaustive and fully describes the system. Thus, a risk-neutral pricer *should* be able to produce the correct prices by herself. In the case of a simple security this is, more or less, the case: by running enough simulations of future returns and averaging them out, the pricer can estimate the security's value. What about compound securities? In this paper we restrict ourselves to the simplest case of a single security (with multiple securities, the combinatorial relations between them may induce computational hardness — an issue not present in the case of a single security). The simplest kind of a single compound security is obtained by compounding European options, which will be the main subject of study in this paper.

A *European call option* is the option to buy an asset at a given price that can only be exercised at a given point of time. An example of a European option O_1 is "The option to

buy one GOOG share at \$900 on 07/01/2015"[1]. If the price of one GOOG share on that date is $\$X$, O_1 will be worth $\max(X-900, 0)$ on that date. A compound option is an option on an option. An example of a compound option O_2 is "The option to buy two O_1 options at \$50 each on 07/01/2014". The focus of this paper is the *computational* hardness of pricing such options even in the presence of full information. To illustrate why compound options cannot be straightforwardly priced using Monte Carlo simulations, note that the price of O_2 today depends not only on our belief about the distribution of the eventual price of GOOG on 07/01/2015, but also on our belief on the distribution of our belief on 07/01/2014 about the price of GOOG a year from then. Intuitively, this roughly squares the amount of simulations that are needed to price O_2 as compared to O_1. In this paper, we make this intuition precise.

1.1 Main results

To the best of our knowledge, we give the first computational-complexity-only hardness results for pricing option contracts on a single security. We obtain results both when the underlying security is modeled algorithmically, and when it can be queried through an oracle. The main results are as follows:

- With unbounded compounding, the problem of (risk-neutral) pricing compound options is **PSPACE**-hard, even in the presence of full information, where the distribution of the future world-states is just uniform, and the function mapping future states to payoffs is a poly-time computable function. Specifically, options considered are compound European put options.

- We analyze the bounded compounding case in the oracle model, where the distribution of the future states is still uniform, and the pricing algorithm may access the ultimate value of the asset via an oracle that maps states to values. For options that are such that pricing $Option(\Phi)$ requires knowing the value of the security Φ with error δ, we show that the query complexity of pricing k-layered options scales as $(\Omega(\delta))^{-2k}$. Thus for small constant k's, such as $k = 3$ or $k = 4$ and $\delta = 10^{-3}$, the query complexity is on the order of $\sim 10^{18}$ or $\sim 10^{24}$, and differences in computing capacity may give the computationally stronger player a real advantage in pricing the security. This suggests one possible explanation for the vast computing resources utilized by financial firms.

1.2 Related work and discussion

While a substantial body of work exists on the information-asymmetry aspects of security-pricing games, there has not been much work directly addressing computational hardness of pricing financial instruments. Arora et al. introduced notions from computational complexity into the study of financial derivatives [ABBG09]. They showed that securitization can amplify the lemon costs arising from asymmetric information between the buyer and seller

[1]The closely related *European put option* is the option to *sell* an asset at a given price that can only be exercised at a given point of time. For example "The option to sell one GOOG share at \$900 on 07/01/2015".

about the underlying assets, by modeling the detection of tampering with the composition of the securities as the problem of finding a dense subgraph. Lemon costs are the costs to efficiency arising from the asymmetry of information between the seller and the buyer, in the case of [ABBG09] the buyer expects that the seller has tampered with the composition of the CDO over the various assets, to allocate risk in a way that is beneficial to the seller. Zuckerman shows that this lemon placement can be remedied if the seller is required to construct derivatives of a certain form, and uses pseudorandom graphs constructed using expanders to demonstrate this [Zuc11]. We note that unlike our result, the main computational hardness result of [ABBG09] still requires some amount of *information asymmetry* to be present (i.e. there is information about the underlying assets that is known to the seller but not the buyer).

High-dimensional markets, such as combinatorial markets, provide a rich ground for hardness results. The high-dimensional nature of these markets allows for computationally complex combinatorics to be encoded into them. Chen, Goel, and Pennock investigate the pricing problem in combinatorial markets — specifically sports tournament results, which, with n teams and 2^{n-1} outcome possibilities, is in general $\#\mathbf{P}$-hard [CGP08]. The same work derives polynomial-time upper bound when restrictions are placed on the betting language (i.e. the types of securities over the n-dimensional outcomes one is allowed to trade in). Pennock also sets forth the idea of opening securities markets for hard problems, called "NP markets" [Pen01].

Closely related to the problem of pricing complex multi-dimensional securities is the problem of market-making over such securities. Bertsimas and Popescu [BP02] show that finding optimal bounds for multi-dimensional derivative prices is **NP**-hard. Fortnow et al. [FKPW05] develop and motivate the concept of a compound securities market (where participants can bet on any boolean statement involving future outcomes). They also analyze the matching problem of an intermediary to carry out optimal auctions in such a market, showing that, with n events, the problem is **coNP**-complete in the divisible case and $\mathbf{\Sigma_2^p}$-complete in the indivisible case. Maymin conjectures that market efficiency implies that $\mathbf{P} = \mathbf{NP}$, and that the market can be programmed to solve **NP**-complete problems [May11].

We see that there is significant work considering the hardness of pricing, or looking into information and sensitivity problems in high-dimensional securities. There has also been work examining how valuations for complex securities differ across investors, as well as examining the demand for such assets. Related fields, such as bounded rationality models for investors, and volatilities in the market have also been studied. However, there have not been prior complexity-theoretic results elucidating the difficulty of pricing an individual security from a purely computational perspective, which are what we initiate in this paper.

We show that hardness can "hide" not just in the n-dimensional space of possibilities spanned by a large number n of securities, but in the interplay between time periods within an n-layer compound security based on a single underlying asset.

The most important open problem raised by our work is understanding the applicability of our results to typical scenarios, as well as understanding both theoretically and empirically what makes *perfectly modeled* securities hard to price. One recurring theme in our examples

is the fact that multi-layered securitization leads to the amplification of the significance of low-probability events. As such, one can view the compound securities we construct as extremely leveraged instruments. What happens if leveraging is limited? In other words, does a bound on the worst-case (or best-case) payoff of the security translate into an upper bound on the complexity of pricing it? If the answer to this question is affirmative, it would imply that the main source of computational hardness in pricing compound securities is the need to account for tiny-probability events to which one is exposed through multi-layered leveraging. Together with the results of the present paper it would suggest that for complicated securities computational complexity is closely related to risk.

Acknowledgments

We would like to thank Itai Ashlagi, Jing Chen, and the anonymous ITCS'14 referees for their helpful comments on earlier versions of the manuscript.

2 Background and notation

2.1 Types of options

A simple European put option $put(\psi, s, t)$ is the right to sell stock ψ at a certain point in time t at a predetermined price s called a strike price (an American put is the right to do so until that point in time). The payoff from the put is the difference between the strike price, and the actual value of the stock at the time of exercise: you would want the stock value to be lower than the strike to make a payoff.

We will begin by focusing on a specific family of normalized (always priced between 0 and 1) multi step put option, which will be defined inductively as follows:

$$\Phi_i = 2 \times put\left(\Phi_{i-1}, \frac{1}{2}, i\right) \in [0,1]. \tag{1}$$

Time intervals are numbered backwards as $n, n-1, \ldots, 0$, so that time i happens before time $i-1$. Here Φ_0 represents the underlying event, Φ_1 will be the option to sell 2 shares of Φ_0 at time 1 (with a strike of 1/2), Φ_2 is the option to sell 2 shares of Φ_1 at time 2 (with a strike of 1/2), etc.

More generally, a sequence $\Phi(t, s, v, \phi)$ of compound European options of depth n is defined by the underlying security ϕ and by three vectors $t \in \{call, put\}^n$, $s \in \mathbb{R}^n$, and $v \in \mathbb{R}^n$. Here $\Phi_0(t, s, v, \phi) := \phi$, and the other layers are defined by induction "backwards": $\Phi_i(t, s, v, \phi)$ is a certain number of call or put options at a certain strike, *priced i time periods before time* 0. Specifically,

$$\Phi_i(t, s, v, \phi) := \begin{cases} v_i \times call\left(\Phi_{i-1}(t, s, v, \phi), s_i, i\right) & \text{if } t_i = call \\ v_i \times put\left(\Phi_{i-1}(t, s, v, \phi), s_i, i\right) & \text{if } t_i = put \end{cases} \tag{2}$$

In this language, the chain of compound securities $\{\phi_i\}$ from (1) corresponds to $\{\Phi_i(t, s, v, \phi)\}$ with $\phi = \phi_0$, $t = (put, put, \ldots, put)$, $s = (1/2, \ldots, 1/2)$, and $v = (2, \ldots, 2)$.

By a slight abuse of notation, we will sometimes identify a security with its risk-neutral value.

3 PSPACE-hardness of pricing multi-layered securities

In this section we show that if the depth of securitization is unbounded, then even pricing a chain of simple put options on a poly-time computable base-security is **PSPACE**-hard. We also show that pricing in any system of polynomial size where all conditional distributions can be computed in polynomial time can be accomplished in **PSPACE**. Thus the pricing problem is **PSPACE**-complete, and the **PSPACE**-hardness result is tight.

3.1 Warm-up: amplifying an exponentially small event

As a warm-up, we will show that pricing a sequence of call options is **NP**-hard. More specifically, we will show how a compound chain of n options, each priced between 0 and 1 expresses the existence of an unsatisfying assignment of a boolean formula $\psi(x_1, \ldots, x_n)$. In other words, the price of the security Ψ_n will be \$1 if ψ is a tautology (always satisfied), and \$0 otherwise. We consider an environment where $x_i \in_U \{0, 1\}$ is revealed at time period i. Thus x_n is revealed first and x_1 is revealed last. The price of Ψ_0 will be just $\psi(x_1, \ldots, x_n)$. We will have

$$\Psi_i = 2 \times call\left(\Psi_{i-1}, \frac{1}{2}, i\right) \in [0, 1]. \tag{3}$$

The price of $\Psi_0 = \psi$ is calculated after all the $x_1, \ldots, x_n$ have been revealed, and is just equal to $\Psi_0(x_1, \ldots, x_n) = \psi(x_1, \ldots, x_n)$. The price of Ψ_1 is calculated when all the x's $x_2, \ldots, x_n$ except for x_1 have been revealed. What is the value of $\Psi_1(x_2, \ldots, x_n)$? At time $i = 1$, the expected value of Ψ_0 is 1 if $\psi(0, x_2, \ldots, x_n) = \psi(1, x_2, \ldots, x_n) = 1$, is 1/2 if exactly one of $\psi(0, x_2, \ldots, x_n)$ and $\psi(1, x_2, \ldots, x_n)$ is 1, and 0 if they are both 0. The call option allowing one to buy Ψ_0 for \$1/2 is therefore worth \$1/2 in the first case and \$0 in the other two cases. Since $\Psi_1(x_2, \ldots, x_n)$ is comprised of two such options we get (by a slight abuse of notation)

$$\Psi_1(x_2, \ldots, x_n) = \psi(0, x_2, \ldots, x_n) \wedge \psi(1, x_2, \ldots, x_n) = \forall x_1 \psi(x_1, x_2, \ldots, x_n).$$

Continuing this reasoning, we get by a simple induction that for all i

$$\Psi_i(x_{i+1}, \ldots, x_n) = \forall x_1 \ldots \forall x_i \psi(x_1, \ldots, x_i, x_{i+1}, \ldots, x_n),$$

which finally leads to

$$\Psi_n = \forall x_1 \ldots \forall x_n \psi(x_1, \ldots, x_n).$$

Therefore Ψ_n will be worth \$1 if ψ is a tautology and \$0 otherwise. This immediately implies that pricing Ψ_n, even when the formula ψ is given explicitly (i.e. we can perfectly model the n-step future) is **NP**-hard.

More importantly, the price of Ψ_n will distinguish between ψ being a tautology and ψ having one unsatisfying assignment x_{unsat}. Since the probability of x_{unsat} being actually

realized is 2^{-n}, this means that the price of Ψ_n detects a probability 2^{-n}-event n steps into the future! Even though x_{unsat} is extremely unlikely to be realized, the possibility of it occurring affects the price of Ψ_n. A real-life analogue of this are examples of securities being affected by a small change in the probability of the US default. While a US default is extremely unlikely, changes in its estimated probability (such as changes in rating between AAA and AA) affect securities prices with this risk built into them quite a bit.

3.2 PSPACE-hardness

Theorem 1. *Consider the multi-step option Φ from (1). Even when the basic security Φ_0 can be priced in polynomial time, and the environment is such that a bit of information is observed uniformly at random at each step, valuing a multi-step option Φ_n is* **PSPACE***-hard.*

Proof. Let $\phi(x_1, \ldots, x_n)$ be any boolean formula. As in the previous section, we assume that x_i is revealed at time i, starting with x_n. We will define the chain of compound options given by (1) with $\Phi_0(x_1, \ldots, x_n) = \phi(x_1, \ldots, x_n)$. Let us calculate the risk-neutral price for Φ_n working from Φ_0 backwards. Recall that

$$\Phi_1(x_2, \ldots, x_n) = 2 \times put(\Phi_0, 1/2, 1).$$

The value of the put option $put(\Phi_0, 1/2, 1)$ depends on the values of $\Phi_0(0, x_2, \ldots, x_n)$ and $\Phi_0(1, x_2, \ldots, x_n)$. The option is worthless, unless $\Phi_0(0, x_2, \ldots, x_n) = \Phi_0(1, x_2, \ldots, x_n) = 0$, in which case it is worth $1/2$. Thus

$$\Phi_1(x_2, \ldots, x_n) = \neg\Phi_0(0, x_2, \ldots, x_n) \wedge \neg\Phi_0(1, x_2, \ldots, x_n) = \neg\exists x_1 \phi(x_1, x_2, \ldots, x_n).$$

Similarly,

$$\Phi_2(x_3, \ldots, x_n) = \neg\exists x_2 \neg\exists x_1 \phi(x_1, x_2, \ldots, x_n) = \forall x_2 \exists x_1 \phi(x_1, x_2, \ldots, x_n),$$

and continuing by induction we obtain (assuming n is even for convenience)

$$\Phi_n = \neg\exists x_n \ldots \neg\exists x_2 \neg\exists x_1 \phi(x_1, x_2, \ldots, x_n) = \forall x_n \exists x_{n-1} \forall x_{n-2} \ldots \forall x_2 \exists x_1 \phi(x_1, x_2, \ldots, x_n).$$

Therefore, pricing Φ_n is equivalent to determining the truth value of a general quantifier-bounded formula (QBF), where the basic formula ϕ is poly-time computable. This problem is **PSPACE**-hard (see e.g. [AB09] for more information about **PSPACE** and QBF), which completes the proof. □

3.3 Tightness: pricing compound securities is in PSPACE

In this section we show that our lower bound from the previous section is tight. That is, we show that the problem of pricing securities of the type slightly generalizing (2). We define an n-period environment with an underlying security $S(x_1, ..., x_n) \in \mathbb{R}$, where $x_i \in_U E_i$ is the

event occurring at step i. Moreover, $|E_i| < 2^{n^{O(1)}}$ (i.e. events have polynomial descriptions)[2], and S is poly-time computable. The pricing is risk-neutral, and contracts are formulated based on prices in future rounds. In its fullest generality, a compound option $\mathcal{S}$ on S is given by a sequence of functions: the type functions $t_i : (x_n, \ldots, x_{i+1}) \mapsto t \in \{call, put\}$, the strike functions $s_i : (x_n, \ldots, x_{i+1}) \mapsto s \in \mathbb{R}$, and the volume functions $v_i : (x_n, \ldots, x_{i+1}) \mapsto v \in \mathbb{R}$. At time $i = 0$ (i.e. at the end), a share of $\mathcal{S}_0$ pays $S(x_1, ..., x_n)$. At time i a share of $\mathcal{S}_{i+1}$ allows one to either buy or sell (depending on the value of $t_i(x_n, \ldots, x_{i+1})$) $v_i(x_n, \ldots, x_{i+1})$ shares of $\mathcal{S}_i$ at price $s_i(x_n, \ldots, x_{i+1})$ each. The goal is to price a share of $\mathcal{S}_n$. Using a simple recursive algorithm, we show this can be done in **PSPACE**:

Theorem 2. *If the functions S, t_i, s_i, v_i are in* **PSPACE***, then pricing $\mathcal{S}_n$ is in* **PSPACE** *as well.*

Proof. Consider the following algorithm:

Price$(i, t, s, v, x_n, \ldots, x_{i+1})$:
  if $i = 0$
    return $S(x_1, \ldots, x_n)$;
  else
    $expectedPrice = \frac{1}{|E_i|} \sum_{x_i \in E_i}$ **Price**$(i-1, t, s, v, x_n, \ldots, x_{i+1}, x_i)$
    if $t_i(x_n, \ldots, x_{i+1}) =$ 'call'
      return $v_i(x_n, \ldots, x_{i+1}) \cdot \max(0, expectedPrice - s_i(x_n, \ldots, x_{i+1}))$
    else
      return $v_i(x_n, \ldots, x_{i+1}) \cdot \max(0, s_i(x_n, \ldots, x_{i+1}) - expectedPrice)$

A simple inductive argument shows that **Price**$(i, t, s, v, x_n, \ldots, x_{i+1})$ correctly prices $\mathcal{S}_i$ given history $(x_n, \ldots, x_{i+1})$. Therefore **Price**(n, t, s, v) will correctly price $\mathcal{S}_n$. It remains to analyze the space complexity of the **Price** algorithm. Note that each instance of the algorithm (without the recursive calls) only uses a polynomial amount of space: to evaluate the functions t, v, s, and S, to enumerate x_i over E_i, and to accumulate the main $\sum$ of $expectedPrice$. Next observe that each recursive call reduces the value of i, thus the depth of the recursion is n, which multiplies the space cost of the algorithm by a factor of n at most. Therefore computing $\mathcal{S}_n$ can be done in **PSPACE**. □

4 The oracle complexity of bounded-layered securities

In this section we consider a more specialized (and, arguably, more realistic) scenario where the securities are of bounded compounding depth, such as depth $k = 3$ or $k = 4$. Intuitively, in this case we expect that the problem of pricing securities would be in polynomial time, but with an exponent growing with k. To the best of our understanding, the existing

[2]Assuming we have an explicit distribution of x_i, we may without loss of generality relabel E_i, perhaps enlarging it slightly, so that it is uniform on E_i.

complexity-theoretic frameworks are not fine enough to prove an $n^{\Omega(k)}$ bound under an accepted hypothesis[3]. Therefore we will prove bounds in the *oracle* model, where the model of the security's behavior is given by an oracle rather than by an explicit formula. This relaxation is justified by the fact that pricing complex securities is usually accomplished by means of Monte Carlo simulations. The oracle set up has the added benefit of enabling *unconditional* results[4].

Specifically, we look at the complexity of pricing multi-layered tranche securities. The simplest tranche contract S_1 on a security S_0 is one that at time $t = 1$ pays the portion of S_0 that is between prices p and $p + \delta$[5]. In addition, we assume that the price of $P(S_0, t = 0)$ is always in $\{0, 1\}$ (this is the case, for example, if S_0 is a debt obligation which may or may not be honored at time $t = 0$, and its price at $t = 1$ is the probability it will be honored, while the price at $t = 0$ is whether it was ultimately honored). The payout of S_1 at time $t = 1$ is given by

$$P(S_1, t = 1) = \begin{cases} 0 & \text{if } 0 \le P(S_0, t = 1) \le p \\ 1 & \text{if } p + \delta \le P(S_0, t = 1) \le 1 \\ \frac{P(S_0, t=1) - p}{\delta} & \text{if } p < P(S_0, t = 1) < p + \delta \end{cases} \tag{4}$$

We have a model which serves as an oracle which outputs random samples for the market between $t = 1$ and $t = 0$, thus giving us samples of the value of $P(S_0, t = 0)$. More specifically, we assume that the state of the market between times $t = 1$ and $t = 0$ are labeled by a distribution $x_1 \in_U E_1$, and $P(S_0, x_1, t = 0)$ is a deterministic function to which we have an oracle access. Using this oracle, we need to estimate the price $P(S_1, t = 1)$.

4.1 Pricing a single tranche contract

As a warm-up, we will show that the query complexity of pricing $P(S_1, t = 1)$, even assuming that $P(S_0, t = 1) \notin (p, p + \delta)$ (and therefore $P(S_1, t = 1)$ must be either 0 or 1) is $\Omega(\delta^2/p)$. While there are several ways of doing it, we will use information-theoretic formalism since it leads to the shortest proof we know of. For the multi-layered case we will need a more elaborate martingale argument. To prove the lower bound, we create an environment where E_1 is exponentially large, and one of the two cases hold: under process $\mathcal{P}_0$, for each $x_1 \in E_1$, $P(S_0, x_1, t = 0) = B_p$ i.i.d. Bernoulli, while under process $\mathcal{P}_1$, $P(S_0, x_1, t = 0) = B_{p+\delta}$. In the former case the price $P(S_1, t = 1) \approx 0$, while in the latter it is ≈ 1. Therefore, pricing $P(S_1, t = 1)$ requires distinguishing between $\mathcal{P}_0$ and $\mathcal{P}_1$ based on samples. In other words, we need to distinguish between samples coming from the two processes.

Denote $V := P(S_1, t = 1)$ the value of S_1 at time 1. Suppose that we have a uniform prior $V \in_U \{0, 1\}$, and we need to use samples to distinguish $V = 0$ from $V = 1$ with a good

[3]It is possible that one can get a computational hardness result under a sufficiently strong version of the Exponential Time Hypothesis.

[4]Note that even the hardness result from the previous section depends on the (very plausible) assumption that $\mathbf{P} \neq \mathbf{PSPACE}$

[5]Thus S_1 represents the $(p, p + \delta)$ tranche of S_0. Equivalently, we could have considered a security representing the $(1 - p - \delta, 1 - p)$-tranche.

probability. Let $Y = Y_1, Y_2, \ldots, Y_N$ be the sequence of price signals obtained by querying $P(S_0, x_i, t = 0)$. Note that by the definition of the process it does not matter which value x_i gets queried as they are i.i.d. given V. We will use mutual information. Recall that the mutual information between two random variables X, Y is defined as

$$I(X;Y) := H(X) - H(X|Y),$$

where $H(X) = \sum_x \Pr[X = x] \log(1/\Pr[X = x])$ is Shannon's entropy. Mutual information captures the amount of information Y reveals about X (the quantity turns out to be symmetric). The basics of information theory can be found in [CT91]. If N queries of the Y_i's suffice to detect V with a high degree of confidence, this means that the mutual information $I(V; Y_1, \ldots, Y_N) = \Omega(1)$. We will show that

$$I(V; Y_1, \ldots, Y_N) = O\left(N \cdot \frac{\delta^2}{p}\right), \tag{5}$$

and therefore it must be the case that $N = \Omega\left(\frac{p}{\delta^2}\right)$. By the chain rule for mutual information,

$$I(V; Y_1, \ldots, Y_N) = \sum_{i=1}^{N} I(V; Y_i | Y_1, \ldots, Y_{i-1}).$$

Theorem 3. *The mutual information between $I(V; Y_i|Y_1, \ldots, Y_{i-1}) = O\left(\frac{\delta^2}{p}\right)$, hence $\Omega\left(\frac{p}{\delta^2}\right)$ queries are needed to to value S_1 even if we restrict the price of S_1 to $\{0, 1\}$.*
If no such restriction is made, then the number of queries needed to get a price on a security within an error of $\varepsilon > 0$ is $\Omega\left(\frac{p}{(\varepsilon\delta)^2}\right)$

Proof. Denote $Z := Y_1, \ldots, Y_{i-1}$. Conditional mutual information can also be written in terms of the Kullback-Leibler Divergence as follows:

$$I(V; Y_i|Z) = E_{VZ} D(Y_i|_{VZ} \| Y_i|_Z).$$

As above, Y_i is drawn from the Bernoulli $B_{p+\delta}$ when $V = 1$ and from B_p when $V = 0$. Since Y_i is a boolean variable, we can write:

$$E_{SZ} D(Y|_{VZ} || Y|_Z) = D(B_\alpha || B_\beta)$$

for some α and β which are determined by V and Z. Here $p \leq \alpha, \beta \leq p + \delta$. The following simple claim is proved in the Appendix:

Claim 4. *For $p < 1/2, \delta < 1/4$, if $p \leq \alpha, \beta \leq p + \delta$, then $D(B_\alpha || B_\beta) = O(\delta^2/p)$.*

The claim immediately implies that $I(V; Y_i|Z) = O(\delta^2/p)$.

For the second statement, we can say that S_1 is a multi-tranche security whose payoff function is tighter than the one used so far, since we now need to distinguish between a price of $P(S_1, t = 1) = 0$, versus, say, price $P(S_1, t = 1) = \varepsilon$, as opposed to just distinguishing 0 from 1. This corresponds to distinguishing a stream of i.i.d. variable B_p from a stream of $B_{p+\varepsilon\delta}$. A proof very similar to the first part demonstrates that the number of queries needed is $\Omega\left(\frac{p}{(\varepsilon\delta)^2}\right)$. □

4.2 Multiple layers

Extending the examples from last section, we consider an environment where E_i is exponentially large and is observed between times $t = i$ and $t = i - 1$. We are able to query the value $P(S_0, x_k, \ldots, x_1, t = 0)$. Extend the definition of S_1 to a multi-layered S_i by generalizing (4):

$$P(S_i, x_k, \ldots, x_i, t = i) = \begin{cases} 0 & \text{if } 0 \leq P(S_{i-1}, x_k, \ldots, x_i, t = i) \leq p \\ 1 & \text{if } p + \delta \leq P(S_{i-1}, x_k, \ldots, x_i, t = i) \leq 1 \\ \frac{P(S_{i-1}, t=i) - p}{\delta} & \text{if } p < P(S_{i-1}, x_k, \ldots, x_i, t = i) < p + \delta \end{cases} \tag{6}$$

Here, we assume that k is a constant (such as 3 or 10), and are interested in the dependence of the query complexity on p and δ.

At first, we place no restrictions on the intermediate prices of $P(S_i, x_k, \ldots, x_i, t = i) \in [0, 1]$. In this scenario, it is quite easy to obtain an exponential (in k) bound on the complexity of pricing $P(S_k, t = k)$:

Theorem 5. *$\Omega\left(\frac{p}{\delta^{2k}}\right)$ queries are required to price a k-layered multi-tranche security. Therefore, the number of queries becomes $\Omega\left(\frac{1}{\delta^{2k}}\right)$ when p is constant.*

It is also not hard to see that the bound in the theorem is tight.

Proof. The proof follows by observing that if the value of S_0 ultimately only depends on x_1, then distinguishing $P(S_k, t = k) = 1$ from $P(S_k, t = k) = 0$ is equivalent to distinguishing $P(S_{k-1}, t = k - 1) = p + \delta$ from $P(S_{k-1}, t = k - 1) = p$, is equivalent to distinguishing $P(S_{k-2}, t = k - 2) = p + p\delta + \delta^2$ from $P(S_{k-2}, t = k - 2) = p + p\delta$, etc. Ultimately, it is equivalent to distinguishing $P(S_0, t = 1) = p + p\delta + p\delta^2 + \ldots + p\delta^{k-1} + \delta^k$ from $P(S_0, t = 1) = p + p\delta + p\delta^2 + \ldots + p\delta^{k-1}$, which by the second part of Theorem 3 requires $\Omega\left(\frac{p}{\delta^{2k}}\right)$ queries. □

4.3 Multiple layers, assuming integral intermediate pricing

One may argue that the main source of hardness in Theorem 5 is the fact that the multi-layered structure allows one to amplify tiny differences in the prices of S_0 at time $t = 1$. Such an amplification becomes impossible if we make the following additional constraint on the environment: at any time t, the expected price of S_t must be $\in \{0, 1\}$. One way to think about it is S_t being a tranche contract valued at time t which may either pay or not pay, depending on the predicted behavior of S_{t-1} at time $t - 1$ after $x_t \in E_t$ is observed.

We show that under this scenario the query complexity drops slightly to $\Theta(p/\delta^2)^k$, although it remains exponential in k. Therefore amplification *per se* is not the only source of hardness in pricing multi-level securities – demonstrating a certain level of robustness to these results.

4.3.1 An upper bound on pricing multi-layered securities

Claim 6. *We can differentiate between the current value of the security being $S_k(t=k)=1$ or $S_k(t=k)=0$ with a high probability using $O\left(\frac{p \cdot log\left(\frac{C}{\delta}\right)}{\delta^2}\right)^k$ observations (C is a universal constant).*

We will show this bound to be tight up to the polylog factors. The proof uses Chernoff bounds and is quite routine. It is deferred to the Appendix.

4.3.2 A matching lower bound

We extend the definition of the one-layer process to a k-layered process that is "barely integral" at all layers as follows. The environment has exponentially large $E_k, \ldots, E_1$, where $x_i \in E_i$ is observed between times i and $i-1$. To each setting of a prefix of variables $x_k, \ldots, x_i$ corresponds a value $P(S_i, x_k, \ldots, x_{i+1}, t=i) \in \{0,1\}$. The value of $P(S_i, x_k, \ldots, x_i, t=i-1)$ is drawn i.i.d. from $B_{p+\delta}$ if $P(S_i, x_k, \ldots, x_{i+1}, t=i)=1$ and from B_p if $P(S_i, x_k, \ldots, x_{i+1}, t=i)=0$. In addition, we start with the prior $P(S_k, t=k) \sim B_{1/2}$, and we need to determine its value.

Theorem 10. *The number of queries required to price a k-layered multi-tranche security without intermediate fractional pricing is $\left(\Omega\left(\frac{p}{\delta^2}\right)\right)^k \cdot p$.*

Proving this theorem will require some preparation. The key to our analysis will be understanding the effect of a single observation of $P(S_0, x_k, \ldots, x_1, t=0)$ on the predicted values along the entire chain $P(S_1, x_k, \ldots, x_2, t=1)$, $P(S_2, x_k, \ldots, x_3, t=2)$, all the way to the main quantity of interest $P(S_k, t=k)$. By a slight abuse of notation we denote by $P(S_i)$ the estimated value (before the last query) of $P(S_i, x_k, \ldots, x_{i+1}, t=i)$. We denote this predicted value if the answer to the query was 0 by $P(S_i) - \Delta_i$, so that the change in the estimated value of $P(S_i, x_k, \ldots, x_{i+1}, t=i)$ effected by querying $P(S_0, x_k, \ldots, x_1, t=0)$ and getting answer 0 is Δ_i. Note that since expected value is a martingale, and the probability of getting answer 1 is $\approx p$, the predicted value conditioned on observing a 1 is approximately $P(S_i) + \Delta_i \cdot (1-p)/p$. We would like to claim that while Δ_1 can be fairly large – as $P(S_0, x_k, \ldots, x_1, t=0)$ can tell us a lot about $P(S_1, x_k, \ldots, x_2, t=1)$, we expect this knowledge to diminish as we look at higher levels of compounding. In fact it does, which eventually gives us a handle on the "progress" made by the querying algorithm.

Claim 7.

$$\Delta_i = \Delta_{i-1} \cdot O\left(\frac{\delta}{p}\right) \cdot P(S_i).$$

The proof is fairly straightforward, but tedious, and can be found in the Appendix.

We will consider the tree of all possible observed sequences $x_k, \ldots, x_1$. Each node on this tree is parametrized by a partial sequence of observations $\eta_i = x_k, \ldots, x_{i+1}$. After τ queries, let $p_{\eta,i}(\tau)$ be the current estimated valuation of the node η_i. Let $\Delta_{\eta,i}(\tau)$ be the change from $p_{\eta,i}(\tau-1)$ to $p_{\eta,i}(\tau)$ *assuming one of the leafs in the subtree of η_i has been queried (and*

0 *otherwise).* Note that the estimated value may change even with other queries, through changes in estimated values of the ancestors of η_i.

The following is proved similarly to Claim 7, with the proof again deferred to the Appendix. Note that the $O(\bullet)$ statement is just a corollary of Claim 7.

Claim 8. *If* $p_{\eta,i}(\tau-1), p_{\eta,i-1}(\tau-1) \in (p/2, 2p)$ *then* $\Delta_{\eta,i}(\tau) = \Theta(\delta \cdot \Delta_{\eta,i-1}(\tau))$.

The $\Delta_{\eta,i}(\tau)$ is now a family of random processes, defined by the querying strategy. At each time τ values along exactly one path from the root to a leaf are affected. We know that the process at each node is a martingale that has a probability $\approx 1-p$ of decreasing and a probability of $\approx p$ on increasing at each step. Unlike the previous lower bound, for consistency we set up the example so that *a-priori* the probability value $p_{\eta,k}(0)$ of the root is p (and not 1/2). Thus the total change in the value of the root, $|\sum_\tau \Delta_{\eta,k}(\tau)| \geq p/2$ approximately holds, since by the end of the process the value of the root $p_{\eta,k}(\tau_{end})$ is determined to be 0 or 1.

The statement of Claim 8 is such that a different process Δ' that we will define will be useful in analyzing the query complexity of the entire process. Δ' is the restriction of the Δ's to paths along which all values are within the interval $(p/2, 2p)$. Specifically, we define

$$\Delta'_{\eta,i}(\tau) := \begin{cases} \Delta_{\eta,i}(\tau) & \text{if } p_{\eta,j}(\tau) \in (p/2, 2p) \text{ for } j = i, i-1, \ldots, 1 \\ 0 & \text{otherwise} \end{cases}$$

Our proof strategy is in two stages: (1) use Claim 8 to argue that the total change $\sum_\tau \Delta'_{\eta,k}(\tau)$ is small unless many queries are made; (2) prove that if the total change in terms of $\sum \Delta'$ is small then also the total change in terms of Δ is small (note that the converse direction is trivial since $|\Delta'| \leq |\Delta|$); (3) therefore to attain the necessary magnitude of $\sum_\tau \Delta_{\eta,k}(\tau)$, we need to make many queries.

We start with the second point, linking the magnitude of the sum of Δ' with the magnitude of the sum of the Δ's:

Lemma 9. *For any stopping rule* T*, and any node* η_i*,*

$$\mathrm{E}\left[\sum_{\tau=0}^{T} \Delta^2_{\eta,i}(\tau)\right] < (\Theta(1/p))^i \cdot \mathrm{E}\left[\sum_{\tau=0}^{T} \Delta'^2_{\eta,i}(\tau)\right]$$

Proof. We will prove the claim by induction on the layer index i. The statement is obviously true for $i = 0$, since for $i = 0$ the node η_0 is either queried or not, and the equality holds in either case. Suppose the claim holds for layers up to $i-1$; our goal is to prove it for i.

We define T' as

$$T' = \min(T, t_{out})$$

where t_{out} is the first time that $p_{\eta,i} \notin \left(\frac{p}{2}, 2p\right)$.

By Claim 7, if we are only concerned with $p_{\eta,i}(\tau-1) \in \left(\frac{p}{2}, 2p\right)$,

$$\sum_{\tau=0}^{T'} \Delta^2_{\eta,i}(\tau) = O\left(\delta^2 \sum_{\tau=0}^{T'} \sum_{\eta_{i-1} \text{ child of } \eta_i} \Delta^2_{\eta,i-1}(\tau)\right)$$

By the inductive hypothesis we have

$$\mathrm{E}\left[\sum_{\tau=0}^{T'}\Delta_{\eta,i}^2(\tau)\right] < \left(\Theta(1/p)\right)^{i-1}\cdot O\left(\mathrm{E}\left[\delta^2\sum_{\tau=0}^{T'}\sum_{\eta_{i-1}}\sum_{\text{child of }\eta_i}\Delta_{\eta,i-1}'^2(\tau)\right]\right).$$

By Claim 8 this implies

$$\mathrm{E}\left[\sum_{\tau=0}^{T'}\Delta_{\eta,i}^2(\tau)\right] < \left(\Theta(1/p)\right)^{i-1}\cdot O\left(\mathrm{E}\left[\delta^2\sum_{\tau=0}^{T'}\Delta_{\eta,i}'^2(\tau)\right]\right).$$

All that is needed to complete the proof is to show that

$$\mathrm{E}\left[\sum_{\tau=0}^{T}\Delta_{\eta,i}^2(\tau)\right] \le O(1/p)\cdot\mathrm{E}\left[\sum_{\tau=0}^{T'}\Delta_{\eta,i}^2(\tau)\right]. \tag{7}$$

This is done via a variance argument which uses basic properties of martingales, and is deferred to the Appendix. □

Theorem 10. *The number of queries required to price a k-layered multi-tranche security without intermediate fractional pricing is $\Omega\left(\frac{\Theta(p)}{\delta^2}\right)^k\cdot p$.*

Remark 11. *For p is not too small, $p=\Theta(1)$, we get a tight lower bound of $\Omega\left(\frac{\Theta(1)}{\delta^2}\right)^k$. With some additional work, the bound in Theorem 10 most likely can be improved to $\Omega\left(\frac{\Theta(p)}{\delta^2}\right)^k$, but at the expense of further complicating the proofs.*

Proof. Note that $E[\sum_{\tau=0}^{T}\Delta_{\eta,k}^2(\tau)]$ is $\Theta(p)$ (we go from a prior probability p to a near-full knowledge about the price of the root node). Therefore, by Lemma 9,

$$\mathrm{E}\left[\sum_{\tau=0}^{T}\Delta_{\eta,k}'^2(\tau)\right] > \left(\Theta(p)\right)^{k+1}. \tag{8}$$

By Claim 8 and because value of $\Delta_{\eta,k}'(\tau)$ is 0 unless $p_{\eta,j}(\tau)\in\left(\frac{p}{2},2p\right)$ for all $j=k,\ldots,1$, we get

$$\mathrm{E}\left[\sum_{\tau=0}^{T}\Delta_{\eta,k}'^2(\tau)\right] < \mathrm{E}\left[\sum_{\ell\text{ leaf node}}\sum_{t=0}^{T}\Delta_{\ell,k}'^2(\tau)\cdot\Theta(\delta)^{2k}\right] = \mathrm{E}[T]\cdot\Theta(\delta)^{2k}. \tag{9}$$

Putting equations (8) and (9) together, we get

$$\mathrm{E}[T] \ge \frac{1}{\Theta(\delta)^{2k}}\cdot\mathrm{E}\left[\sum_{\tau=0}^{T}\Delta_{\eta,k}'^2(\tau)\right] > \frac{\Theta(p)^{k+1}}{\Theta(\delta)^{2k}},$$

which completes the proof of the theorem. □

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*, volume 1. Cambridge University Press Cambridge, 2009.

[ABBG09] Sanjeev Arora, Boaz Barak, Markus Brunnermeier, and Rong Ge. Computational complexity and information asymmetry in financial products. *Princeton Center for Computational Intractability, http://www. cs. princeton. edu/ rongge*, 2009.

[BP02] Dimitris Bertsimas and Ioana Popescu. On the relation between option and stock prices: a convex optimization approach. *Operations Research*, 50(2):358–374, 2002.

[CGP08] Yiling Chen, Sharad Goel, and David M Pennock. Pricing combinatorial markets for tournaments. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 305–314. ACM, 2008.

[CT91] T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley Series in Telecommunications. John Wiley & Sons Inc., New York, 1991. A Wiley-Interscience Publication.

[FKPW05] Lance Fortnow, Joe Kilian, David M Pennock, and Michael P Wellman. Betting boolean-style: a framework for trading in securities based on logical formulas. *Decision Support Systems*, 39(1):87–104, 2005.

[May11] Philip Z Maymin. Markets are efficient if and only if p= np. *Algorithmic Finance*, 1(1):1–11, 2011.

[Pen01] David M Pennock. Np markets, or how to get everyone else to solve your intractable problems. In *Workshop on Economic Agents, Models, and Mechanisms at the 17th International Joint Conference on Artificial Intelligence (IJCAI*. Citeseer, 2001.

[Zuc11] David Zuckerman. Pseudorandom financial derivatives. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 315–320. ACM, 2011.

A Deferred proofs

Claim 4. *For $p < 1/2, \delta < 1/4$, if $p \leq \alpha, \beta \leq p + \delta$, then $D(B_\alpha||B_\beta) = O(\delta^2/p)$.*

Proof. Kullback-Leibler Divergence is defined as

$$D(P\|Q) = \Sigma p(x) log \frac{p(x)}{q(x)}$$

Now the observation can be either 1 or 0, so

$$D(B_\alpha\|B_\beta) = \alpha \log \frac{\alpha}{\beta} + (1-\alpha) \log \frac{1-\alpha}{1-\beta} = \log \frac{1-\alpha}{1-\beta} + \alpha \log \frac{\alpha - \alpha\beta}{\beta - \alpha\beta}.$$

Using the Taylor series expansion of $\log(1+x)$ (assuming $\alpha, \beta < 3/4$):

$$\begin{aligned}
D = -\frac{\alpha-\beta}{1-\beta} - \frac{(\alpha-\beta)^2}{2\cdot(1-\beta)^2} - \frac{(\alpha-\beta)^3}{3\cdot(1-\beta)^3}\ldots \\
+ \alpha\left(\frac{\alpha-\beta}{\beta-\beta\alpha} - \frac{(\alpha-\beta)^2}{2\cdot(\beta-\beta\alpha)^2} + \frac{(\alpha-\beta)^3}{3\cdot(\beta-\beta\alpha)^3}\ldots\right) \\
D = -\frac{\alpha-\beta}{1-\beta} + \frac{\alpha-\beta}{1-\alpha} + \frac{(\alpha-\beta)^2}{\beta(1-\alpha)} - \frac{(\alpha-\beta)^2}{2\beta(1-\alpha)^2} - \frac{(\alpha-\beta)^2}{2(1-\beta)^2} + O\left(\frac{(\alpha-\beta)^3}{\beta^3}\right) \\
D = \frac{(\alpha-\beta)^2}{(1-\beta)(1-\alpha)} + \frac{2(\alpha-\beta)^2(1-\alpha) - (\alpha-\beta)^2}{2\beta(1-\alpha)^2} - \frac{(\alpha-\beta)^2}{2(1-\beta)^2} + O\left(\frac{(\alpha-\beta)^3}{\beta^3}\right) \\
D = \frac{(\alpha-\beta)^2(1+\beta-2\alpha)}{2\beta(1-\beta)(1-\alpha)^2} - \frac{(\alpha-\beta)^2}{2(1-\beta)^2} + O\left(\frac{(\alpha-\beta)^3}{\beta^3}\right) \\
D(B_\alpha||B_\beta) = \frac{(\alpha-\beta)^2(1-2\alpha-\beta+4\alpha\beta-\beta^2-\beta\alpha^2)}{2\beta(1-\beta)^2(1-\alpha)^2} + O\left(\frac{(\alpha-\beta)^3}{\beta^3}\right) \\
D(B_\alpha||B_\beta) = \frac{(\alpha-\beta)^2(1-2\alpha-\beta+4\alpha\beta-\beta^2-\beta\alpha^2)}{2\beta(1-2\alpha-\beta+4\alpha\beta-\beta^2-\beta\alpha^2+\alpha^2\beta^2-\alpha^2\beta+\alpha^2-2\alpha\beta^2-\beta)} \\
+ O\left(\frac{(\alpha-\beta)^3}{\beta^3}\right)
\end{aligned}$$

Therefore,

$$D(B_\alpha||B_\beta) = O\left(\frac{(\alpha-\beta)^2}{\beta}\right) \tag{10}$$

where $p \leq \alpha, \beta \leq p + \delta$ and $0 < p$, which is a small fraction. This is largest when α is $p + \delta$ and β is p

$$D(B_\alpha||B_\beta) = O\left(\frac{\delta^2}{p}\right)$$

□

Claim 6. *We can differentiate between the current value of the security being $S_k(t=k)=1$ or $S_k(t=k)=0$ with a high probability using $O\left(\frac{p\cdot\log\left(\frac{C}{\delta}\right)}{\delta^2}\right)^k$ observations (C is a universal constant).*

Proof. (Sketch) Our algorithm is to sample n values of the security one level up from the current security being looked at in the multi-layer security. For example, while deciding whether $P(S_1, x_k, \ldots, x_2, t=1)=0$ or $P(S_1, x_k, \ldots, x_2, t=1)=1$, we sample n values of $P(S_0, x_k, \ldots, x_2, x_1, t=0)$, one level into the future. By the integrality condition, in the worst case, we need the n queries to distinguish the Bernoulli Distribution B_p from $B_{p+\delta}$. Even if we needed to distinguish $B_{p+\delta/3}$ from $B_{p+2\delta/3}$, we could do it by Chernoff bounds, using $O\left(\frac{p\cdot\log\left(\frac{C}{\delta}\right)}{\delta^2}\right)$ queries, we can achieve correct identification with error $< 3\delta/C < \delta/3$. The answer can now be fed to the next layer to yield the claim. □

Claim 7.

$$\Delta_i = \Delta_{i-1}\cdot O\left(\frac{\delta}{p}\right)\cdot P(S_i).$$

Proof. Denote by $ev1$ all the queries collected so far that correspond to the history $x_k, \ldots, x_i$, and denote by ev all other queries. Let obs be the answer to the query $P(S_0, x_k, \ldots, x_1, t=0)$, and $Q_i := P(S_i, x_k, \ldots, x_{i+1}, t=i) \in \{0,1\}$ and $Q_{i-1} := P(S_i, x_k, \ldots, x_i, t=i-1) \in \{0,1\}$. By definition,

$$\Delta_i = P[Q_i=1|ev, ev1] - P[Q_i=1|ev, ev1, obs=0].$$

Conditioning on Q_{i-1}, we get

$$\begin{aligned}\Delta_i = {} & P[Q_i=1|ev, ev1, Q_{i-1}=0]P[Q_{i-1}=0|ev, ev1] \\ & - P[Q_i=1|ev, ev1, obs=0, Q_{i-1}=0]P[Q_{i-1}=0|ev, ev1, obs=0] \\ & + P[Q_i=1|ev, ev1, Q_{i-1}=1]P[Q_{i-1}=1|ev, ev1] \\ & - P[Q_i=1|ev, ev1, obs=0, Q_{i-1}=1]P[Q_{i-1}=1|ev, ev1, obs=0]\end{aligned}$$

When conditioning on Q_{i-1}, conditioning on $obs=0$ is irrelevant. Also, we see that

$$\begin{aligned}P[Q_{i-1}=0|ev, ev1, obs=0] - {} & P[Q_{i-1}=0|ev, ev1] \\ & = P[Q_{i-1}=1|ev, ev1] - P[Q_{i-1}=1|ev, ev1, obs=0] = \Delta_{i-1}\end{aligned}$$

Therefore the term is now

$$\Delta_i = (P[Q_i=1|ev, ev1, Q_{i-1}=1] - P[Q_i=1|, ev, ev1, Q_{i-1}=0])\Delta_{i-1}$$

by Bayes rule

$$
\begin{aligned}
&= \left(\frac{P[Q_{i-1}=1|ev,ev1,Q_i=1]P[Q_i=1|ev,ev1]}{P[Q_{i-1}=1|ev,ev1]} - \frac{P[Q_{i-1}=0|ev,ev1,Q_i=1]P[Q_i=1|ev,ev1]}{P[Q_{i-1}=0|ev,ev1]} \right) \\
&\qquad \times \Delta_{i-1} \\
&= \left(\frac{P[Q_{i-1}=1|ev,ev1,Q_i=1]}{P[Q_{i-1}=1|ev,ev1]} - \frac{P[Q_{i-1}=0|ev,ev1,Q_i=1]}{P[Q_{i-1}=0|ev,ev1]} \right) \Delta_{i-1} P[Q_i=1|ev,ev1] \\
&\leq \left(\frac{P[Q_{i-1}=1|ev1,Q_i=1]}{P[Q_{i-1}=1|Q_i=0,ev,ev1]} - \frac{P[Q_{i-1}=0|ev1,Q_i=1]}{P[Q_{i-1}=0|Q_i=0,ev,ev1]} \right) \Delta_{i-1} P[Q_i=1|ev,ev1]
\end{aligned} \tag{11}
$$

since the denominators are resized to exaggerate the difference, and conditioning on ev is irrelevant if conditioning on Q_i as well.

Consider the following equations which follow from Bayes rule, and the fact that conditioned on Q_{i-1}, $ev1$ is independent of Q_i:

$$
\frac{P[Q_{i-1}=1|Q_i=1,ev1]}{P[Q_{i-1}=1|Q_i=0,ev1]} = \frac{P[Q_{i-1}=1|Q_i=1]P[ev1|Q_i=0]}{P[Q_{i-1}=1|Q_i=0]P[ev1|Q_i=1]}
$$

since $P[ev1|Q_{i-1}=1]$ cancels out from the numerator and the denominator. Therefore,

$$
\frac{P[Q_{i-1}=1|Q_i=1,ev1]}{P[Q_{i-1}=1|Q_i=0,ev1]} = \frac{(p+\delta)P[ev1|Q_i=0]}{pP[ev1|Q_i=1]}
$$

$$
P[ev1|Q_i=0] = P[ev1|Q_{i-1}=1]P[Q_{i-1}=1|Q_i=0] + P[ev1|Q_{i-1}=0]P[Q_{i-1}=0|Q_i=0]
$$

by breaking up the above expression through conditioning on Q_{i-1}.
Denote $\alpha = P[ev1|Q_{i-1}=1]$, $\beta = P[ev1|Q_{i-1}=0]$. Then

$$
P[ev1|Q_i=0] = \alpha p + \beta(1-p)
$$

Similarly

$$
P[ev1|Q_i=1] = \alpha(p+\delta) + \beta(1-p-\delta)
$$

Therefore,

$$
\frac{P[Q_{i-1}=1|Q_i=1,ev1]}{P[Q_{i-1}=1|Q_i=0,ev1]} = \frac{(p+\delta)(\alpha p + \beta(1-p))}{p(\alpha(p+\delta) + \beta(1-p-\delta))}
$$

Which for $\delta < p < 1/3$ implies

$$
\frac{P[Q_{i-1}=1|Q_i=1,ev1]}{P[Q_{i-1}=1|Q_i=0,ev1]} \leq \frac{p+\delta}{p} \cdot \left(1 + \frac{\delta}{p}\right)
$$

By exactly the same reasoning,

$$
\frac{P[Q_{i-1}=0|Q_i=1,ev1]}{P[Q_{i-1}=0|Q_i=0,ev1]} \geq \frac{1-p-\delta}{1-p} \cdot \left(1 - \frac{\delta}{p}\right)
$$

So, going back to where we left off after (11),

$$\Delta_i \leq \left[\frac{p+\delta}{p}\cdot\left(1+\frac{\delta}{p}\right) - \frac{1-p-\delta}{1-p}\cdot\left(1-\frac{\delta}{p}\right)\right]\Delta_{i-1}P[Q_i=1|ev,ev1]$$

Therefore, we have proved the claim, since the equation above implies that

$$\Delta_i = O\left(\frac{\delta}{p}\right)\cdot\Delta_{i-1}P[Q_i=1|ev,ev1],$$

which is our claim. □

Claim 8. *If $p_{\eta,i}(\tau-1), p_{\eta,i-1}(\tau-1) \in (p/2, 2p)$ then $\Delta_{\eta,i}(\tau) = \Theta(\delta\cdot\Delta_{\eta,i-1}(\tau))$.*

Proof. As before, let $ev1$ be results of queries corresponding to the subtree of η_{i-1} and let ev be all other queries, accumulated by time $\tau-1$. As in the proof of Claim 7 let $Q_i := P(S_i, x_k, \ldots, x_i, t=i) \in \{0,1\}$ and $Q_{i-1} := P(S_i, x_k, \ldots, x_{i-1}, t=i-1) \in \{0,1\}$, where node η_{i-1} corresponds to the sequence $x_k, \ldots, x_{i-1}$.

We start off by defining a few terms. For $y, z \in \{0,1\}$,

$$a_{yz} = Pr[Q_{i-1}=y|ev1, Q_i=z]$$

$$b_y = Pr[Q_i=y|ev,ev1]$$

$$\Gamma_{yz} = Pr[Q_i=y|Q_{i-1}=z] = \frac{Pr[Q_{i-1}=z|Q_i=y]Pr[Q_i=y]}{Pr[Q_{i-1}=z]}$$

Now, by the third line of equation (11),

$$\Delta_{\eta,i}(\tau) = \left(\frac{P[Q_{i-1}=1|ev,ev1,Q_i=1]}{P[Q_{i-1}=1|ev,ev1]} - \frac{P[Q_{i-1}=0|ev,ev1,Q_i=1]}{P[Q_{i-1}=0|ev,ev1]}\right)\Delta_{\eta,i-1}(\tau)P[Q_i=1|ev,ev1]$$

In terms of our notation above, and because conditioning on ev is irrelevant while conditioning on Q_i, this equals

$$\begin{aligned}\Delta_{\eta,i}(\tau) &= \left(\frac{a_{11}}{a_{11}b_1+a_{10}b_0} - \frac{a_{01}}{a_{01}b_1+a_{00}b_0}\right)\Delta_{\eta,i-1}(\tau)P[Q_i=1|ev,ev1] \\ &= \frac{a_{11}a_{00}b_0 - a_{01}a_{10}b_0}{(a_{11}b_1+a_{10}b_0)(a_{01}b_1+a_{00}b_0)}\Delta_{\eta,i-1}(\tau)P[Q_i=1|ev,ev1]\end{aligned}$$

Since $a_{11}b_1+a_{10}b_0$ is approximately p (by definition), and b_0 and $a_{01}b_1+a_{00}b_0$ are approximately $1-p$, we get

$$\begin{aligned}\Delta_{\eta,i}(\tau) &\approx \frac{1}{p}\cdot(a_{11}a_{00}-a_{01}a_{10})\Delta_{\eta,i-1}(\tau)P[Q_i=1|ev,ev1] \\ &= \frac{1}{p}\cdot\frac{Pr[Q_{i-1}=1|ev1]Pr[Q_{i-1}=0|ev1]}{Pr[Q_i=1|ev1]Pr[Q_i=0|ev1]}\times \\ &(Pr[Q_i=1|Q_{i-1}=1]Pr[Q_i=0|Q_{i-1}=0] - Pr[Q_i=1|Q_{i-1}=0]Pr[Q_i=0|Q_{i-1}=1])\times \\ &\qquad\qquad \Delta_{\eta,i-1}(\tau)P[Q_i=1|ev,ev1]\end{aligned}$$

We can see that $\frac{Pr[Q_{i-1}=1|ev1]Pr[Q_{i-1}=0|ev1]}{Pr[Q_i=1|ev1]Pr[Q_i=0|ev1]}$ is equal to (up to a multiplicative constant) $\frac{p(1-p)}{p(1-p)}$ if $Pr[Q_{i-1}=1|ev1] \in \left(\frac{p}{2}, 2p\right)$, so we have (again, using the definition of Γ above)

$$\Delta_{\eta,i}(\tau) \approx \frac{1}{p}(\Gamma_{11}\Gamma_{00} - \Gamma_{10}\Gamma_{01})\Delta_{\eta,i-1}(\tau)P[Q_i = 1|ev, ev1]$$
$$= \frac{1}{p} \cdot \frac{Pr[Q_i = 0]Pr[Q_i = 1]}{Pr[Q_{i-1} = 0]Pr[Q_{i-1} = 1]} \times$$
$$(Pr[Q_{i-1} = 1|Q_i = 1]Pr[Q_{i-1} = 0|Q_i = 0] - Pr[Q_{i-1} = 0|Q_i = 1]Pr[Q_{i-1} = 1|Q_i = 0]) \times$$
$$\Delta_{\eta,i-1}(\tau)P[Q_i = 1|ev, ev1]$$

Since $\frac{Pr[Q_i=0]Pr[Q_i=1]}{Pr[Q_{i-1}=0]Pr[Q_{i-1}=1]}$ does not depend on any evidence, it represents the original probabilities, and thus exactly equals 1 (since $P[Q_{i-1} = 1] = P[Q_i = 1] = p$ initially), so we are left with

$$\Delta_{\eta,i}(\tau) \approx \frac{1}{p}(Pr[Q_{i-1} = 1|Q_i = 1]Pr[Q_{i-1} = 0|Q_i = 0] - Pr[Q_{i-1} = 0|Q_i = 1]Pr[Q_{i-1} = 1|Q_i = 0]) \times$$
$$\Delta_{\eta,i-1}(\tau)P[Q_i = 1|ev, ev1]$$
$$= \frac{1}{p}[(p + \delta)(1 - p) - (1 - p - \delta)p]\Delta_{k-1}Pr[Q_i = 1|ev, ev1]$$
$$= \frac{1}{p}\delta\Delta_{\eta,i-1}(\tau)Pr[Q_i = 1|ev, ev1]$$

For the node probability value being within our restricted $\left(\frac{p}{2}, 2p\right)$ environment, this means that

$$\Delta_{\eta,i}(\tau) = \Theta(\delta)\,\Delta_{\eta,i-1}(\tau)$$

completing the proof of the claim. □

Proof of equation (7)**.**

Proof. We will specifically show that

$$\mathrm{E}\left[\sum_{\tau=0}^{T} \Delta^2_{\eta,i}(\tau)\right] \leq \frac{8}{p} \cdot \mathrm{E}\left[\sum_{\tau=0}^{T'} \Delta^2_{\eta,i}(\tau)\right].$$

Firstly, note that T equals T' whenever $p_{\eta,i}(\tau) \in \left(\frac{p}{2}, 2p\right)$ for $\tau = 0, 1, \ldots, T$.
Next, by a general property of martingales,

$$\mathrm{E}\left[\sum_{t=0}^{T} \Delta^2_{\eta,i}(\tau)\right] = Var(p_{\eta,i}(T)) \text{ and } \mathrm{E}\left[\sum_{t=0}^{T'} \Delta^2_{\eta,i}(\tau)\right] = Var(p_{\eta,i}(T')).$$

Therefore, all we have to prove is:

$$Var(p_{\eta,i}(T)) \leq \frac{8}{p} \cdot Var(p_{\eta,i}(T')). \tag{12}$$

We have

$$Var(p_{\eta,i}(T)) = E[(p_{\eta,i}(T) - p_{\eta,i}(0))^2]$$
$$= E[(p_{\eta,i}(T) - p_{\eta,i}(0))^2 | T = T'] \cdot Pr[T = T'] + E[(p_{\eta,i}(T) - p_{\eta,i}(0))^2 | T > T'] \cdot Pr[T > T'] \tag{13}$$

and

$$Var(p_{\eta,i}(T')) = E[(p_{\eta,i}(T') - p_{\eta,i}(0))^2]$$
$$= E[(p_{\eta,i}(T) - p_{\eta,i}(0))^2 | T = T'] \cdot Pr[T = T'] + E[(p_{\eta,i}(T') - p_{\eta,i}(0))^2 | T > T'] \cdot Pr[T > T']. \tag{14}$$

The first parts of (13) and (14) match. The contribution of $(p_{\eta,i}(T') - p_{\eta,i}(0))^2 | T > T'$ to (14) is at least $(p/2)^2 = p^2/4$, while the contribution of $E[(p_{\eta,i}(T) - p_{\eta,i}(0))^2 | T > T']$ to (13) is at most $Var(B_{p+\delta}) < 2p$. Therefore the ratio is bounded by $8/p$, completing the proof. □