NEST Decentralized Probabilistic Virtual Machine Model

nestprotocol.org

August 13, 2022

Abstract

This paper introduces and discusses the model and primary mechanism of NEST. After Ethereum, the demand for the randomness of on-chain smart contracts is dramatically increasing, and the need for risk management of assets is also rapidly growing. Therefore, the importance of stochastic assets and their related programming and applications has become self-evident. NEST introduces rich random sources for on-chain transactions and all smart contracts by designing the NEST oracle, quoting price information from the NEST oracle, and distribution transformation; therefore, it provides investors with customizable and vibrant stochastic information-based assets through the OMM mechanism and PVM. Stochastic calculations are also prepared for the future of various on-chain applications.

Contents

1 Introduction

Satoshi Nakamoto's ground-breaking insights about blockchain have drawn our attention from traditional assets. Blockchain and Bitcoin strengthen the security and openness of information and data by employing the consensus mechanism, so that information such as balances will not be used for double-spending. The scarcity of information in the process of transmission and use is ensured, which is a crucial step for information to be converted into assets. Without any room for debate, the purest digital asset is Bitcoin, which does not require a third party to confirm a transaction or prevent double-spending.

As a new concept of blockchain and encryption technology, Ethereum adheres to the ideologies of Satoshi Nakamoto and the transfer convenience of digital assets. It retains the information properties of digital assets, making it possible for programming, and this is the core value of the Ethereum virtual machine.

In the process of programming, the program calculated output number of assets will not increase (may reduce, destroyed, or locked, for example): If an address has x ETH, then x can be considered as a deterministic variable, which will become $f(x)$ in EVM after calculation. If no other assets are involved, the calculation must conform to $x \geq F(x)$, which is the first constraint of the Ethereum virtual machine.

However, Ethereum also has a downside: all asset information is deterministic (x is a deterministic variable). If we want x to be a random variable X, that is, the result of a random event determines the value of this variable X, it is uncertain before the occurrence of this random event, and we only have information about its distribution at most. In addition, X is introduced into programming and calculation to obtain this random variable's functional value $F(x)$. Note that $F(X)$ itself is also a random variable, which is beyond the reach of the current EVM mechanism. Fortunately, in the oracle we introduced stochastic price information based on the OMM (omnipotent market maker) mechanism to generate tokens of which the return can represent random information flow. The number of it before a given time period is uncertain, but settlement of a certain number of tokens at certain blocks can be done according to the expected distribution figures. This is the tokenization process of random information; we name this token NEST coins. Therefore, a NEST coin is a unit of

value for random data.

To achieve more applications involving random variables, after implementing the tokenization of random information, we can make functional transformations to these random variables; this process is to obtain random variables with different distributions. These distributions can be used in Defi, Gamefi, NFT, etc. Of course, including all transformation functions for practical purposes is unnecessary. Still, we can consider using some standard essential functions as a set of bases on which the NEST function space is derived. For example, suppose the transformation keeps the expected value decreasing all the time, namely $\mathbb{E}(X) \geq \mathbb{E}(F(X))$, where $\mathbb{E}$ denotes the expected value, then like the EVM constraint. In that case, this is a system with decreasing expectations or a deflationary system. We name this system the Probabilistic Virtual Machine (PVM). With PVM, we are no longer subject to the initial value constraint of ETH: for example, in derivatives trading under the AMM mechanism, the seller LP cannot guarantee that the option can be settled at any price. This is an incomplete probability space, and PVM solves this problem.

PVM is the primary technology and idea discussed in this paper. Under the widespread use of PVM, various applications no longer need to construct their own tokens system to obtain a viable economic exchange value. Instead, this exchange is mainly reflected in the risk-return structure swap, a more typical market behavior in the modern economy. In this context, we can call the NEST coin a universal coin because it has the potential for all economic relationships and exchanges.

2 History of EVM & PVM

From the original paper of Satoshi Nakamoto, the birth of Bitcoin, a series of homogeneous cryptocurrencies, and finally to the smart contract of Ethereum, the cryptocurrency industry has advanced significantly over the years. As an informationized asset, cryptocurrencies have attributes that other traditional assets do not have, which makes these assets have broader application space and enormous potential.

2.1 Informationized Assets

Turning information into assets was difficult before the advent of blockchain technology. As we have mentioned in the introduction, a piece of information has value if it is helpful to someone. However, the problem with the value of information is that it can easily be copied or double spent. If the intermediate receiver is not the designated user, encryption can be used to keep the information from being revealed in the transmission process. Since the intermediate receiver cannot decrypt the information, the value of information is not diminished. But once the information is in the hands of the user, the user can copy it to the same available environment; no matter whether the actual user of this environment is himself or others, the value of the information will be reflected, which is the meaning of double-spending. Encryption technology makes the information value of the communication process secure, but it cannot guarantee that the transmitted data is not double spent. Unlike ordinary commodities, information can be copied indefinitely, and the cost is negligible. In this context, it is challenging to guarantee information scarcity, so it is impossible to accurately give feedback on the value of information with price, let alone denote these pieces of information as assets.

The real solution to this problem is Bitcoin's mechanism. Bitcoin's answer is building a closed environment for information usage. If each address acts as a participant, its consistency variable is also the most important information – the BTC balance must be used on the Bitcoin blockchain. Beyond that, no off-chain reading or manipulation is sufficient to change this information.

Moreover, when information is exchanged on the chain, there is a cost to pay, and confirmation is done through a consensus mechanism. This mechanism makes it impossible to double spend the balance, effectively maintaining the scarcity of information in the closed environment of the Bitcoin blockchain. This solution allows us to look at information from the perspective of assets, which, like gold, precious metals, and commodities, are finite, exhaustible, and transferable.

Bitcoin creates informationized assets that not only possess the attributes of the above general assets but also have the characteristics of ease of transfer and preservation. In addition, because its roots are game theory-based, system participants will gradually follow the progress and

direction of this technology. Henceforth, this system is change resistant, unlike some commodities, which can be directly replaced by technological progress (like precious metals replacing seashells). This combination of the convenience of information and the resistance to change makes Bitcoin play a critical role in the future asset world. We refer to crypto assets and digital currencies as the seventh class of assets after traditional currencies, real estate, precious metals, commodities, equity, and debt.

2.2 The Informationized Assets Programming and EVM

Although digital assets have the property of traditional asset classes, the characteristics of information are still retained, which means that the conventional means of processing information are still effective, such as the programming of data. The difference is that the information after programming is still scarce and has asset properties (on-chain); this type of programming differs from traditional methodologies. It is programming based on value– not just data programming. Many people refer to this as the new generation of the Internet, Web3 - the Internet of value.

EVM is positioned to capture the value of the demand for informationized asset programming. Since the addresses, balances, and information within the Ethereum environment (in-block data) of crypto assets or digital currencies are standardized, they can be understood as base vectors. On this base, a Turing-complete virtual machine is built, unlike a traditional computer in computing and storage. However, for asset attributes, it can be understood that information (flow) carries A unit of value after function transformation. For example, an address A has x ETH, and when it enters EVM for programming, the output becomes $f(x)$ ETH, which strictly conforms to $x \geq f(x)$. In this sense, it is possible to divide the informationized assets on the blockchain into two parts: information and asset units. The former is processed as information, while the latter keeps the unit constant after output. This procedure is true if the condition $x \geq f(x)$ is satisfied.

This idea of adding a unit of value after function transformation has not been fully explored. For example, the ETH issuance mechanism is similar to BTC and does not fully integrate into EVM, this limits the application scope of ETH tokens. Furthermore, this leads to situations where efficient settlement or market clearing cannot be achieved.

There are many application examples where it is challenging to employ EVM. A typical example is financial derivatives, where the potential for future revenue streams is infinite, and the current ETH Token design cannot cope with this situation. Furthermore, whatever the initial input is, it is not enough to ensure a valid settlement. These problems might be solved if the ETH token integrates deep into the EVM, allowing instructions to generate new ETH. As the underlying infrastructure, it is enough for EVM to realize the completeness of its basic instructions. Whether the additional issuance of units of value is related to instructions is a problem involving the design of economic systems rather than purely technical systems.

3 NEST

To solve the above problems and introduce random variables and distribution transformation for on-chain programming, we designed NEST. Its main concepts include the NEST oracle, random information tokenization, Omnipotent market maker (OMM) mechanism, probabilistic virtual machine (PVM), and NEST coin.

3.1 Stochastic Asset

In the previous chapter, we have mentioned that blockchain technology and Ethereum make informationized assets and informationized asset programming a reality, so how do we introduce random information and random information flow into the chain?

3.1.1 Random Information

The issue of EVM and ETH Tokens discussed above is not only a design issue but also a question of whether random information flows can be introduced onto the chain. There are two straightforward methods to introduce random information sources into the chain. One is to reference information guaranteed or centralized, such as getting information from audited information nodes or centralized service providers. The second is to start with the Hash value of each block on the chain and treat this value as a uniformly distributed random variable. If random information sources are introduced in a guaranteed or centralized manner, this will upset the

decentralized risk-return structure of ETH. Transforming a random distribution from an on-chain source, such as hash values, involves the risk of generating pseudorandom numbers, affecting the consensus of the ETH community. How to introduce random information in a decentralized manner and ensure that the random information flow is not compromised is a complex problem in the industry. Existing projects that provide random information flow are not ideal solutions because they cannot be verified on the chain and are essentially wholly node-dependent. The real breakthrough for this problem comes from the NEST Oracle, which provides information generated entirely by users playing a decentralized role, as described in the following sections and attachments.

3.1.2 NEST Oracle

NEST oracle is the first fully decentralized price oracle, including two-way options, price chain, beta coefficient, and other modules. Introduction of price information flow on the chain is done via the arbitrage effect. Suppose we have a risk of centralization in introducing price information flow; the whole process will have a chance of completely changing the nature of things, which is why we are not willing to adopt other blockchain oracles. Essential features of NEST's oracle are as follows: First, it is entirely permissionless and open to quotation, so anyone can participate in the quotation and form a price on the chain. The second is on-chain verification. NEST prices must be verified on-chain rather than off-chain, which is the key to NEST's design. NEST is designed based on security and robustness, which enables everyone to change and influence the generation of prices on the chain. At the same time, it makes the price information converge to the equilibrium price under the protection of the mechanism.

All fully decentralized oracle must be designed based on a decentralized nature. These designs must find an optimal solution among information density, information bias, information costs, and security costs. While we cannot guarantee that the design of the NEST oracle is perfect, it does fulfill the criterion in the above-discussed areas and is backed by sound data support and feedback.

For more information about NEST oracle, please refer to the Appendix.

3.1.3 Random Information tokenization and Stochastic Asset

If there is a random variable X, when the variable X is stored at a given address A, it means that address A can establish a connection according to the distribution of X at a specified time. This connection can be either data or value: for example, when X is equal to a particular value x, the address will have x units of an asset (a token). This value-based connection is called tokenization of random variables. After we tokenize a random variable, it is identified by the system as a stochastic asset by the random variable plus the unique name of the token.

ETH Tokens cannot support random variable tokenization simply because their number is finite; the finite number will lead to the value range of the random variable being constrained. For example, a stochastic with a 1 in 1 billion chance of getting 1 billion ETH cannot be implemented with ETH because there is only 100 million ETH in total. In other words, the tokenization process of random variables supported by ETH is incomplete – specific values lead to event outcomes that cannot be realized. This is the fundamental reason why derivatives such as options mentioned above cannot be settled under the existing token system – their probability space is incomplete.

To solve this problem, we must redesign an on-chain asset that can be issued and destroyed in response to random information flows rather than mined in the BTC way. In the ERC20 paradigm, we need to add a mechanism to issue new tokens given the information flow. For example, we take the NEST coin as the unit of value of information flow. When the variable in information flow equals 10000, 10000 NEST will be sent to the corresponding address. At the same time, to ensure the value of system and tokens, we also need to make the total number of tokens under this mechanism decrease with a significant probability. Therefore, a reasonable plan is for each system to have a corresponding cost. We call this cost the production cost of stochastic assets; that is, every stochastic asset in the system needs to pay a consideration when it is generated. We have a relatively simple scheme for how to express this: When a stochastic X is generated, at least $\mathbb{E}(X)$ NEST is paid, where $\mathbb{E}(X)$ is the expected value of the random variable X. This is the constraint generated by the stochastic and is the first constraint of the NEST system.

3.1.4 OMM

If we look at the tokenization of random information from a trader's perspective, it can be understood as a trading mechanism. This trading mechanism is not the traditional financial trade practice where buyers and sellers match trades. Instead, everyone trades with the contract and swaps the risk-return structure by converting one stochastic asset into another in the contract. We call this system the OMM mechanism. You get a stochastic asset by paying its expected value, or you get a new stochastic asset by destroying a stochastic asset with a higher expected value. This process realizes the risk-return change and ensures the system's expected supply convergence. It should be pointed out that the system's supply is also a stochastic process. If we use its variance or second moment to measure the risk, the risk is shared by the investors holding the token. This is not quite the same as risk-taking or risk-independence in traditional finance. This brand-new risk feature is unique to digital currencies and is the direction of future technology development.

3.2 Programming with Stochastic Asset and PVM

This section will discuss the concept of stochastic asset programming in NEST and the critical requirements for PVM.

3.2.1 Distribution Transformation, Base Functions, and Application Space

If we assume that we have a random variable X, we can introduce a distribution transformation function F such that $F(X)$ becomes another random variable we need. Such transformations always exist and can be implemented through smart contracts. The simplest way to do that is to go from a uniform distribution to a normal distribution, or a geometric normal distribution back to a uniform distribution, and so on; you can find the corresponding function F to achieve that. Let us look at EVM from another perspective. Since any programming of reference variable X is essentially a function transformation, we can see that the combination of continuous distribution transformations and EVM's operation on smart contracts are very similar. Therefore, in the current EVM, the reference and calculation of the price information flow provided by the NEST oracle is the distribution transformation of the random process represented by the

NEST price information flow. Furthermore, since EVM always takes into account the complexity of computation and storage, the function range of the distribution transformation is the distribution range of the NEST price oracle – the function space represented by the distribution transformation and the distribution space of the NEST oracle – under a limited computation and storage complexity (e.g., the Gas limit).

We can simplify the system by expressing the distribution transformation as a linear combination of some underlying functions. Such a representation is entirely feasible in the range of continuous functions. Finding this base set of functions is similar to finding a simple basis for all continuous functions, such as polynomial functions. This basis approximates all continuous functions – in our case, all distribution transformations – by linear transformations. Of course, the number of bases could be infinite, but we do not necessarily need such a vast space of functions. We only need a few common functional transformations, especially those most commonly used in economic activities and other applications. To control the computational complexity within a given range, we do not even need these functions to be in the same family of functions (such as polynomials), as long as they are simple enough to be necessary. So, our goal is to find some standard, simple basic distribution functions whose linear combinations form a role space that covers the significant distribution functions. The underlying distribution function represents NEST's understanding of the current range of blockchain applications and is one of the keys to the entire system. In earlier versions, only a simpler family of functions is provided. Moving on, we will gradually improve the function family with the deepening of the application and demand. In the later stages, we can discuss and improve the extension of the function family through governance.

3.2.2 The Second Constraint Condition and PVM

Suppose we want to transform one random variable from a distribution into another and keep the value contraction in the system rather than letting the underlying values decrease. In that case, we need the second constraint: $\mathbb{E}(X) >= \mathbb{E}(F(X))$, which is very similar to the first constraint $X \geq F(X)$ in EVM. We call this the second constraint on the informationized asset settlement system. Without this second constraint, the whole system could become inflationary, leading to a constant dilution of the underlying asset

value. As we have mentioned earlier, a new stochastic has a cost of $\mathbb{E}(X)$, whereas programming based on stochastic assets starts from $\mathbb{E}(X)$ and guarantees that the expected value of each step of the program is not higher than that cost.

Given a random variable, we can express the distribution transformation as a linear combination of the underlying functions. These basic function has a corresponding expectations, such as $\mathbb{E}(F_1(X))$, $\mathbb{E}(F_2(X))$. These expected values form the cost of the underlying function, and the expected values of other complex distributions are linear combinations of the expected values of these underlying functions, so their costs are linear combinations of these underlying costs. We can compare the base functions to EVM instructions and base costs to EVM gas costs, and the whole process can be considered a PVM. All complex applications are nothing more than PVM calls. PVM effectively expands the boundary of EVM and greatly shortens some financial products' development process. The original complex logic is simplified into the exchange of a random variable, which is extremely important for industry development.

3.3 NEST coin Economics

NEST coin's economic model is more complex than other informationized assets. Generally speaking, the price of an asset is affected by the amount of application demand and supply, that is, the game between supply and demand. Therefore, the game relationships involved in NEST mechanics are more complex, which is why we call them game networks. The games involved in the NEST system are divided into three aspects: the game between the offerer and the verifier, the game between the token buyer and seller, the game between the stochastic investor and the system, resulting in the risk-sharing among all token holders.

The game between the bidder and the verifier: In the process of offering a price for the NEST oracle, a quotation that deviates from the actual market price can be arbitraged away by the verifier, who needs to provide a new quote. This process ensures the accuracy and verifiability of the information flow provided by the NEST oracle. In addition, bidders are rewarded with NEST coins for their quotes, thus becoming miners or minters in the system. Furthermore, the NEST oracle generates less than 3% of the total number of tokens per year through mining new tokens.

The game between buyer and seller of tokens: Like other tokens, based on market demand, it is influenced by people's judgment and expectations of the future market.

The game between investors and the system: Investors can select and customize stochastic tokens on NEST's platform to meet their risk requirements. This is a process for investors to actively manage their asset risks, which is also the main application direction of NEST PVM. During the investment process, the system accounts for the gains and losses of the stochastic assets selected by the investors. Since the first and second constraints guarantee a decrease in expectation, given a sufficient number of investors, the total amount of NEST coins will decline by the Law of Large Numbers.

Risk sharing for token holders: All NEST token holders will jointly bear the result of the change in the total amount of NEST coins. Since the supply of NEST coins is limited in the long run, the long-term value of NEST coins is guaranteed to rise.

3.3.1 NEST coin

Stochastic assets can be applied in a wide range of fields. Before generating any random tokens, the tokenization process of random information requires a unit representing the asset's value. All data flows in the NEST system can be unified with NEST coin as the unit of value. For example, if the value of a stochastic is a particular value x at some point, then the address that owns the stochastic will have x NESTs. NEST, therefore, becomes a systematic unit of account with the potential to generate different stochastic assets with varying information flows. The initial number of NESTs can be fixed, and then for every stochastic asset that is generated, it has to pay at least its expected value. In terms of expected total supply, the number of NESTs is decreasing, a downward curve. In terms of the specific supply amount, it is a distribution band that fluctuates around this downward curve. The management of the concentration of the distribution band depends on the control of the system.

3.3.2 Second Moment Control

As mentioned above, the expected value of NEST's supply is restricted, but its second moment changes. The system, therefore, has to do something

about this second moment. There are two kinds of management ideas in general. One is to limit the expansion of second moment. For example, from a specific engineering point of view, we can determine the total size of some random variables to ensure that their total second moment is small, or we can construct an approximation algorithm to describe the second moment of the population to control its boundary. Another approach is to use a market mechanism for specific function calls, where the number or size of calls increases by a more significant coefficient (greater than 1) to the expected payout value. This is equivalent to adjusting the second moment by the expected value or the cost. Both of these options can be tried and implemented. At the moment, we prefer the latter, which will be reflected in future versions, but it is more likely that both approaches will be adopted simultaneously.

3.3.3 The Equilibrium between Supply and Demand

As we have mentioned earlier, NEST's expected supply is convergent. Therefore, the expected total number of tokens should be deflationary, regardless of the oracle mining incentive. Furthermore, demand is growing. We chose the NEST coin as the universal coin based on the above application example. This is because NEST has built-in usage scenarios rather than relying on outside investment or speculation, meaning two things. The first is that NEST is a functional token, and the second is that it is so widely used that it encompasses most of the current range of blockchains. Therefore, NEST has a price logic built into it: the equilibrium between increasing expected demand and decreasing total supply drives NEST's expected price continue to rise.

4 Applications

From the above description, NEST can be simplified to a production and programming system for stochastic assets. This not only solves the derivatives settlement problem that ETH cannot accomplish but also dramatically simplifies the development process of Defi and Gamefi. It also avoids the problem of running a new unit of value while developing a project, allowing developers to focus on application development rather than token liquidity operations.

Because NEST is the new risk-trading archetype, it does not need to be hedged like traditional financial assets. It is therefore perfect for providing large amounts of liquidity. Below we shall discuss how this can be applied to different scenarios:

- Derivatives: Futures and options are the most direct beneficiaries of the NEST system. Because the contract is the only seller, there underlying derivatives need the demand parameters and corresponding expected values to be provided before they can be sold. For example, if someone needs a call option based on ETH/USDT, the strike price and strike date need to be entered to get the cost calculable, and the option can be exercised at expiration according to the conditions set.
- Parallel Asset/ Stablecoins : Based on the price system, parallel assets representing arbitrary price information flow can be generated, and their earning are the same as the original assets. For example, PBTC, PUSD, PETH, etc., the flow of price information generated in this way can be duplicated by collateralizing the NEST or the assets accepted by the system, or it can be produced based on the convergence algorithm (supported by the system's interest rate oracle), but the cost paid must be guaranteed to match the generated assets.
- Bond and interest rate oracle: The system can create an endogenous bond reflecting the overall supply and demand to reflect the time value of NEST, which can also be understood as an interest rate oracle, a system-level variable.
- Borrowing and lending: In the case of parallel assets, borrowing and lending become very simple, and liquidity is sufficient, provided, of course, that the overall convergence property of the system, the interest rate of borrowing and lending, needs to be discounted against the price information flow, and can reflect the future price changes and changes in the interest rate oracle.
- Dabs: An asset-backed bond based on NEST or ETH WBTC, backed by a NEST contract that guarantees the payment of the bond and whose interest rate can be maintained based on an interest rate oracle.
- Dex: Based on the NEST oracle and parallel asset/price coins, trading becomes more accessible and feasible with no liquidity barriers.

- Probability coin: An underlying stochastic asset that reflects a variety of distributions.
- Game item composition: Any Game item composition is a swap of economic relationships, so it is only necessary to call the PVM function to correspond the item to an NFT and their composition relationships.
- Gamefi cross-platform interoperability: When different platforms are developed with NEST, they have the same system algorithm for clearing and billing tokens. This is the benefit of One NEST and One Coin: This creative model can be highly convenient for Gamefi platforms to call NEST functions interchangeably in the NEST protocol, as long as NEST constraints are met.
- Putable NFT: A Nest-linked NFT structure that can set a floor price according to a given algorithm and be interconnected as a whole to trade with the system after certain constraints are met.

5 Summary

NEST creatively introduces random information flow with a decentralized oracle and tokenizes random information flow through the OMM mechanism, generating many stochastic assets. The NEST probabilistic virtual machine PVM, based on basic functions, can program stochastic assets, which applies to a large number of real-world scenarios. An inherent cost mechanism ensures the system supply converges, resulting in a new universal coin with an innate price appreciation logic. The NEST coin provides a whole new development tool and is a creative new asset for the blockchain world.

Appendix A Appendix: NEST Oracles

1 Introduction: The Challenge of Price Oracles

Price oracles commonly used in the DeFi industry generally reflect the asset price of centralized exchanges by "trusted" nodes, where the price is "uploaded" to the chain for usage by DeFi protocols. There is a basic problem with verifying such price data. Some DeFi projects utilize price data gathered from decentralized exchanges, however, because transaction volume is minimal, the pricing data is readily manipulated and vulnerable to attack. This creates a clear market need for an Oracle solution that directly checks the pricing to ensure the information is correct and timely but is also prohibitively expensive to attack. This system should also be decentralized to reduce the risks of centralization.

Oracle price data must meet the following key requirements:

- Accuracy: The price data on the oracle should truly reflect the market price.
- Price sensitivity: The price data on the oracle should react fast enough to market movements.
- Attack resistance: The cost of distorting or affecting the real price is extremely high for any attackers.
- Direct verification: The verifier can be any third party, and no centralized review or threshold is required.
- Distributed quotation system: no centralized review or threshold is required, and anyone can freely join or leave at any place and at any time.

2 NEST Solution

NEST provides a creative solution, including collateral asset quotation, arbitrage verification, price chain, beta coefficients, and other modules to form a complete NEST protocol. Taking the Ethereum network as an example, the schematic dia-

gram of the NEST protocol is described in Figure 1 below and we will discuss the details in the following subsections.

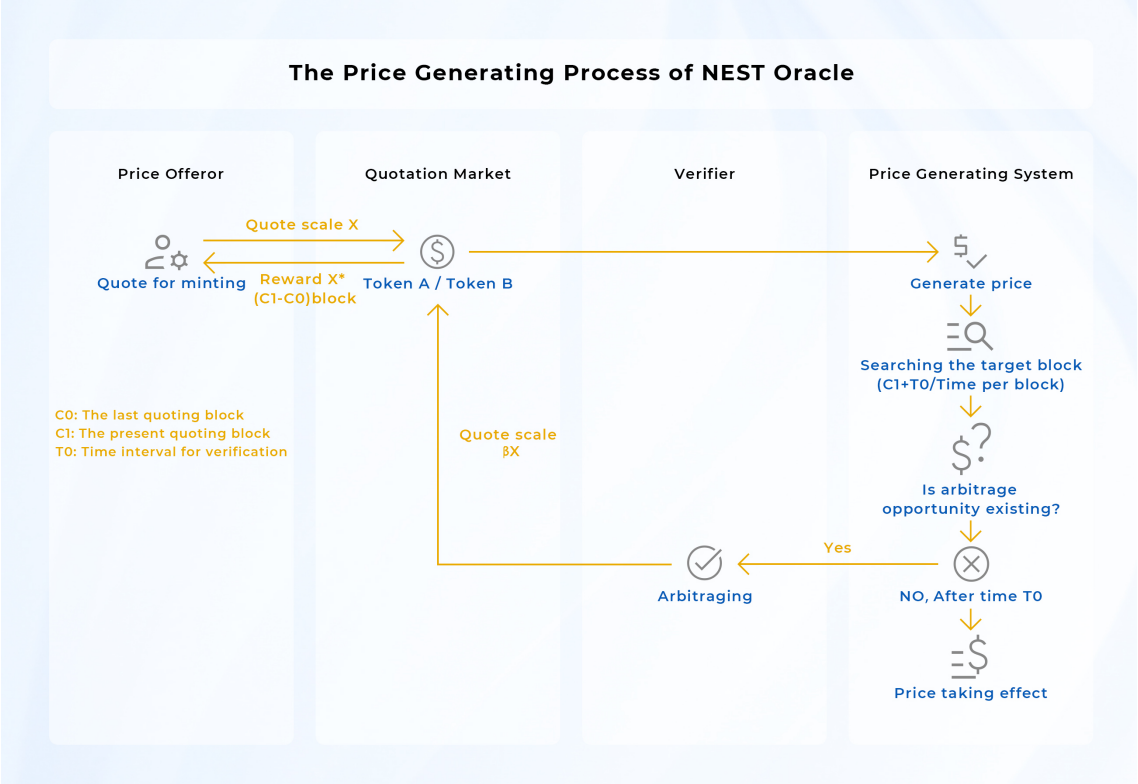

Figure 1: Diagram of NEST Protocol

2.1 Price Model of NEST Oracle

NEST oracle is the only truly decentralized oracle on the market today: given an off-chain price stream, how to design a decentralized game such that the game equilibrium can output a price stream with the smallest possible deviation from the off-chain price stream. NEST oracle solves this problem with quotation mining, two-way options, validation cycles, price chains and β factors. NEST provides a price sequence that does not change the distribution of asset prices but approaches a discrete sampling model, which is determined by the structure of the decentralized game, where the quote deviation and quotation density depend on the depth of the

arbitrage market and the price of the NEST token. Overall, NEST provides an efficient decentralized oracle that maintains the fundamental traits of asset prices. In practice, we tend to use highly efficient market prices, and hence choose the most liquid underlying assets such as BTC and ETH, etc.

The basic price model follows the Geometric Brownian Motion (GBM) model. Considering the characteristics of prices deviation and discrete time, we correct the prices using the k-factor as follows,

$$k = \max(\frac{|p_2 - p_1|}{p_1}, 0.002) + \sqrt{t} \cdot \max(\sigma, \sigma_0) \tag{1}$$

where p_2 and p_1 represent the current and previous prices respectively, t, measured by second, represents the difference between the time transaction happens and the time p_2 becomes effective. Furthermore, σ the instantaneous volatility follows

$$\sigma = \frac{|p_2 - p_1|}{p_1\sqrt{T}}$$

where T represents the time-lapse between p_1 and p_2 becoming effective. σ_0 denotes the regular volatility, set by the protocol (generally different values for different financial products).

The correct procedure follows

- when it comes to a call option, the long price is $(1 + k)p$ while the short price is $\frac{p}{1+k}$
- when it comes to a put option, the long price is $\frac{p}{1+k}$ while the short price is $(1 + k)p$

where p represents the base price.

Since price is verified on-chain, NEST has provided an open and transparent ecosystem for everyone. One of the most important points is openness: anyone can start a price information flow and motivate price providers to mint any kind of token. For example, a project can set up the price pair of its own token to USDT, and motivate others to provide price information by rewarding them with this token. This would help any project to expand the number of minters in its ecosystem.

2.2 Roles of NEST Protocol Actors

Participants in the NEST protocol are as below:

- Price Makers: The participants who submit price quotations to the protocol. This includes miners who quote prices for mining and verifiers who complete the transaction and quotation.
  - Miners: Providing quotations to receive NEST (ERC-20 Token). Miners are denoted as O, and anyone can become a miner.
  - Verifiers: If the quotation price deviates from the market price, the verifier can trade a quoted asset at the quoted price to earn revenue. The verifier needs to "force" a quotation at the time of the transaction and does not need to pay a commission nor participate in mining. Verifiers are denoted as A, and anyone can become a verifier.
- Price Callers: The contract or account that "calls" the NEST protocol quotations and pays the fee is called a price caller. Price callers are denoted as C. Any contract or account can become a price caller, but this will generally be reserved for other DeFi protocols and institutions.

2.3 Quotation Mining and Price Verification

One can easily start a quotation channel via NEST protocol where he/she needs to set the quotation pairs (one channel allows multiple pairs), quotation scale, commission fee, the token and scale of the collateral, etc.

Taking ETH/USDT as an example, miner O intends to quote a price of 1 ETH = 100 USDT. At this time, miner O needs to input the collateral NEST and the quoted assets, ETH and USDT, into the quoted contract. The scale is x ETH and $100x$ USDT, and the paid commission is λx ETH. Miners participate in mining based on a commission scale to earn NEST. The whole process is completely open and transparent, that is, anyone can assume the role of O, and the price and scale are set independently.

After miner O submits the collateral, assets and price to the quoted contract, verifier A believes that the price presents an arbitrage opportunity, and can trade either ETH or USDT at the quote from miner O, which is 1 ETH = 100 USDT. This mechanism ensures that the maker's price is either the fair price in the market or the equivalent price of the two assets recognized by himself/herself. In the view of miner O, 1 ETH and 100 USDT are equivalent, so it does not matter which asset the verifier trades. This process is the price verification period.

Essentially, miners, through quoting, also provide either bullish or bearish two-way options during the verification period, with the strike price as its quoted price. Verifiers, then, execute this option if they find that there is an arbitrage opportunity. Therefore, if miners want to minimize their costs, they need to report the price that is least likely to be traded during the verification period. This allows the miner's quotation has a certain ability to forecast future prices. For the verifier, whether they choose to arbitrage (execute) depends on the difference between the quote and market price. We call the minimum difference the verifier will take action on the "minimum arbitrage space"; this value also depends on the length of the verification period and the transaction cost.

The formula for quote mining is expressed by the following formula: Maker O quotes p, that is, 1 ETH = p USDT, the asset scale is x ETH, so the corresponding USDT quantity = $x \cdot p$. The commission scale for participating in mining is $w = \lambda \cdot p$, and verifier A can use the price p to trade $x \cdot p$ USDT for x ETH.

2.4 Price Verification Period

Opened quotes have an allotted period of time attached, denoted as T_0. This time determines the period of risk the maker takes and the price sensitivity. After the verification period, quotations that have not been traded are called "effective quotations" which includes two variables - price and quotation scale (p, x). Effective quotations form the block price mentioned in section 2.6. However, the price quoted that is already traded by the verifier will not be adopted. If a certain quoted price

is partially traded, the remaining part is also an effective quote, i.e. (p, x'). After the price verification period is complete, the maker's remaining assets will be made available to withdraw at any time.

The verification cycle affects miners, quotation costs, and price accuracy. The longer the time, the higher the option cost, and the more difficult it is to predict the future price. Judging by current DeFi market demands for price data and the volatility of mainstream assets, a reasonably set T_0 is between 5 to 10 minutes (pending adjustments and optimization based on the ETH network capacity and verifiers, scale, with the optimal time being within 1 minute). Note that if a price has passed the verification cycle, it indicates that there is no arbitrage space between this price and the current market equilibrium price (the minimum arbitrage space is determined by T_0 and transaction costs), thus representing the approximate current price; the existence of T_0 does not mean a delay in prices.

2.5 Price Chain

According to the above agreement, the verifier needs to force a new price after accepting the transaction of a maker. To put it simply, the verifier needs to offer a new price to close the opening left by the rejected price. For example, verifier A_1 and maker O accept the transaction with the price of p_0 (the maker O's quotation scale is x_0 with the collateral scale of y_0), so A_1 needs to quote a price p_1 to the contract immediately with the asset scale of x_1, and transfer x_1 ETH and $x_1 \cdot p_1$ USDT together with the collateral y_1 to the contract. Commission and mining participation rewards are not paid at this time. If verifier A_2 accepts the transaction with A_1, A_2 needs to quote the price p_2 with the asset scale of x_2 and the collateral scale of y_2. A continuous price chain with T_0 as the maximum quotation interval is formed:

$$p_0 \to p_1 \to p_2 \cdots$$

the quoted asset chain is

$$x_0 \to x_1 \to x_2 \cdots$$

and the collateral asset chain is

$$y_0 \to y_1 \to y_2 \cdots .$$

2.6 Block Price

The NEST Oracle determined price is recorded on the blockchain, with each block recording a price. The effective price in the block is generated by a certain algorithm. The price is called the block price or NEST-Price. Assuming the effective quotation of a block is (p_1, x_1), (p_2, x_2), $(p_3, x_3) \cdots$ the block price is

$$P = \frac{\sum_{i=1}^{M} p_i \cdot x_i}{\sum_{i=1}^{M} x_i}$$

where M represents the number of effective quotations in this block. If there are no effective quotations in a current block, the price of the most recent block will be used.

2.7 Price Sequence and Volatility

Each block of the Ethereum network corresponds to a price on NEST, thereby forming a price sequence. The price sequence has important functions, including:

- Provide an average price for DeFi operations, including the arithmetic average price of N consecutive blocks $j = 1, \cdots, N$

  $$P_s = \frac{\sum_{j=1}^{N} P_j}{N} ,$$

  or the weighted average price of N consecutive blocks:

  $$P_m = \frac{\sum_{j=1}^{N} P_j \cdot Y_j}{\sum_{j=1}^{N} Y_j}$$

  where $Y_j = \sum_{i=1}^{M_j} x_{ij}$ represents the total asset scale of all effective quotations in block j and M_j the number of effective quotations in block j.

- Provide volatility indicators for most DeFi derivatives, such as rolling volatility of 50 consecutive quotes, or various other volatility indicators customized for DeFi purposes.

- Other statistics.

2.8 Attack-Resistant Algorithm

If the scale of DeFi assets calling the NEST-Price is very large, there is a huge opportunity for attacks. An attacker may tamper with a normal quote, p_0, and changed it to p_1, or the attacker may trade maliciously, hoping that the price will not be updated (as prices cannot be adopted and updated once the price has been traded). With attackers willing to sacrifice the price difference between P_1 and P_0 in exchange for greater profits, the price-setting mechanism becomes invalid. So how does NEST prevent these kinds of attacks?

By increasing the cost for attackers. First, the price chain itself is an attack-resistant mechanism: attackers must offer an alternative price and the corresponding assets at this price after attacking the price. After the attack, attackers must either offer the same "correct" price or leave an arbitrage opportunity. There must be a verifier in the market to recognize the arbitrage opportunity and revise the quote.

Secondly, in order to amplify the cost to the attacker, we arrange every verifier's quotation asset scale as follows: the scale of the verifier's transaction is x_1, and the scale of the simultaneous quotation is $x_2 = \beta x_1$ with $\beta > 1$. Therefore, the verifier must quote at a price more than double the scale of the quotation. Notice that we only allow this amplification for quotation asset up to 4-round verification. On the other hand, we also enlarge the collateral asset in the same way but without 4-round limitation. As an example of $\beta = 2$, the quoted asset chain and the collateral asset chain in section 2.5 follow as

$$x_0 \to \beta x_0 \to \beta^2 x_0 \to \beta^3 x_0 \to \beta^4 x_0 \to \beta^4 x_0 \to \cdots \to \beta^4 x_0 \to \cdots$$

and

$$y_0 \to \beta y_0 \to \beta^2 y_0 \to \beta^3 y_0 \to \beta^4 y_0 \to \beta^5 y_0 \to \cdots \to \beta^n y_0 \to \cdots$$

respectively.

Attackers either offer huge arbitrage opportunities to the market (the scale increases by levels, making this kind of attack almost ineffective) or must continue to use an extremely high volume of assets to self-deal based on the market price to delay the opportunity for price adoption. For example, assuming that the verification period is set as $T_0 = 5$ minutes, if miner O makes one quotation at present, to prohibit this quotation become the effective price in coming 1 hour, the attacker needs at least $6144y_0$ collateral asset and $32x_0$ each quoted asset. Furthermore, the attacker needs at least $12284y_0$ collateral asset and $300x_0$ each quoted asset to paralyze NEST quotation for 1 hour if the miners make the quotation every 5 minutes in the coming 1 hour. Notice that the quotation channel zero set $y_0 = 100,000$ NEST. Only focusing on the collateral asset, 1,228,400,000 NEST makes this attack plan almost impossible to fulfill considering that the total circulation of NEST is not over 3 billion. This kind of attack-resistant ability cannot be achieved by centralized exchanges.

2.9 Incentives and Economics

Miners obtain NEST Tokens through paying ETH commissions and taking certain price fluctuation risks. Verifiers earn profits directly based on the calculation of price deviation while also bearing the risk of the quoted transaction, so for the verifiers, the cost/benefit is relatively clear. For the miners, the model of quotation mining requires a corresponding economic foundation. ETH contributed by miners is denoted as X, and will be returned back to NEST holders regularly, usually on a weekly basis. This process builds an automatic distribution model, so that each NEST Token has intrinsic value, which is verifiable on-chain. Only relying on the quotation miner's ETH is not enough to complete the logical closed-loop system, which returns to the original intention of constructing the price oracle. The fact that the on-chain price is a core demand for all DeFi products means it is often regarded as the most integral part of DeFi infrastructure. DeFi developers and users should pay the corresponding fees when using NEST-Price denoted as Z. Therefore, the

value of NEST is denoted as X+Z. In general, the cost of obtaining NEST is X and NEST creates value for NEST holders throughout the whole ecosystem. The value of NEST is typically greater than the overall cost. For each miner, the cost is uncertain, so there exists a trading possibility. Under the assumption that the overall value is greater than the overall cost, NEST holders with different costs can compete with each other to achieve organic equilibrium, which is similar to the equilibrium found in the stock market. All tokens in the entire NEST ecosystem are generated by mining, and there is no reservation or pre-mining. All costs of generating NEST will be returned to NEST holders, and NEST is only used for incentives. The NEST model achieves complete decentralization, as anyone can join the system, and its characteristics are similar to that of Bitcoin. The NEST protocol upgrades the DAO method, where adjustments need to be first proposed and then approved by a 51% majority via community voting before being implemented.

2.10 The New Characteristics of Latest NEST

The most recent version of NEST is NEST 4.4. The new characteristics of NEST 4.4 compare to the early versions are:

- Improved techniques: allow price offering for multiple assets, in one smart contract, one can start the price information flow for more than ten different assets. In this way, gas fee can be saved handsomely. The efficiency of uploading information is much better.

- Improved economic models: cancel the quotation commission fee. Calling quotation price from NEST is also free now. In the meantime, the mint production is reduced to 1/6 compared to before. The circulation increases slower, slower than 3% per year. In the long run, these changes will guarantee the increasing value of NEST. The total number of NEST will not exceeding 3,000,000,000 (3 billion). The threshold of price information offering is lower, only 0.01 ETH and assets of the same value is needed to be deposited.

3 The Application of NEST-Price

Although NEST focuses on on-chain price data, it can also design price-equilibrium products including the following:

(1). Equilibrium Token: A digital asset that represents economic equilibrium formed by excess collateralization and market arbitrage mechanisms. This can also represent the equilibrium exchange relationship between prices. Equilibrium tokens can be regarded as on-chain valuation units composed of token generation contracts, arbitrage mechanisms, and feedback correction mechanisms. The important significance of equilibrium tokens is in their unique foundation, which increases or decreases following the changes of the entire public chain, such as the Ethereum blockchain. Secondly, they can be proven on chain with a risk-reward structure different from ETH.

(2). Decentralized Transactions: Traditional decentralized transactions are mainly based on peer-to-peer quotation matching. This is fundamentally flawed, as the core of modern exchanges is bilateral auctions, which have the characteristics of forced ordering and forced transactions at prices for both parties. This type of feature involves calculation characteristics, which do not match the current serial queuing mechanisms of the blockchain. A meaningful decentralized transaction would be a market-making system, that is, a two-way forced acceptance of quotations, which can be achieved perfectly with the NEST quotation mechanism.

(3). Automatic Settlement Mortgage Loan: Due to on-chain data, a loan contract that involves liquidation or automatic settlements can quote prices and automatically trigger restrictions, so that loan behavior is not limited to the options of contract structures.

(4). Futures: A distributed futures model is similar to an equilibrium token currency, but it also introduces arbitrage from any third party. This can am-

plify the transaction scale of forward transactions or directly earn revenue from transaction price fluctuations. This was impossible to design before now. All general futures require a centralized institution to perform forced liquidations, but distributed futures do not bear the risk of centralization.

(5). Volatility Products: Derivatives based on the volatility of equilibrium prices are used to hedge or smooth derivatives risks due to the on-chain equilibrium price sequence.

The above only takes the most basic products in finance as an example. Through using NEST-Price, a complete spectrum of decentralized financial products that differ from previous basic peer-to-peer transactions can be designed. Due to the introduction of global variables, the entire DeFi ecosystem is set to enter a new era. As for why DeFi needs global variables, this is because of the nature of finance and general equilibriums, rather than partial equilibriums. A simple local supply and demand relationship is insufficient; there needs to be an effective and complete pricing system based on the whole market arbitrage mechanism. This is not possible for the commodity economy, as simple peer-to-peer transactions cannot solve fundamental financial problems. However, in order not to bear the risk of centralization but also to have generally equal characteristics, global variables like "price" are needed. This variable cannot be introduced centrally, so our oracle scheme is a fundamental part of the infrastructure underpinning the entire field of decentralized finance.

4 Quotation Risk of NEST-Price

As with all financial products and services, NEST-Price is not without risk. Whilst many risks are unable to be described or recognized due to their inherently personal nature, here is a brief description of the quotation risk of NEST-Price:

(1). Due to the existence of the minimum arbitrage, there may be some risks when using NEST-Price for financial services that require extremely high price accuracy. This should be taken into account when designing.

(2). The market arbitrage mechanism is not aggressive enough, which is reflected in inadequate efforts by arbitrageurs. When there is a huge opportunity for arbitrage, no one notices it. This requires higher market acceptance and recognition as the industry develops further.

(3). Although the price cannot be attacked directly, the price mechanism can be attacked indirectly through attacks on NEST. For example, attackers can take more than 51% of the NEST tokens and then modify important parameters to invalidate the quotation mechanism. This problem can be prevented by limiting key parameters while increasing the NEST market's size, making 51% of attacks more difficult to achieve.

(4). The risk of code vulnerabilities or significant external changes. If there are vulnerabilities in the underlying Ethereum code, the NEST system code, or a significant change in the external environment, the price caller will be affected. This can be corrected through on-chain governance and contract forks.

Appendix B The Accuracy of the NEST Price

The Accuracy of the NEST Price

NEST Research Academy

September 2020

ABSTRACT

This short article develops a model to estimate the difference between the NEST price and a source price, e.g. price from an exchange. Under plausible assumptions, we show that the difference can be as small as 0.003 when volatility is small. It can even be lower if the transaction cost in the blockchain gets lower.

1. Model Setup

A *price-provider* is an individual who inputs a price into the NEST system and waits for a certain number of blocks passing to be verified by other individuals. The operation is equivalent to write an American type call and put option that anyone else can exercise it by using the input price as the exercise price. Thus, the price-provider shall minimize the value of this option by carefully choosing an input price. Precisely, the price-provider's objective problem is

$$P^* = \arg\min_P \left(\max_\tau E^Q[e^{-r\tau}|S_\tau - P|] \right), \tag{1}$$

where $\tau \leq T_0$ is a stopping time and T_0 is a fixed time horizon[1] , P is the input price decided by the price-provider. In other words, the price-provider has to minimize the value of one American type option by choosing an appropriate exercise price P. Here asset price $S_t, t \geq 0$ shall be referred to the price in an exchange at time t. Thus, the market is complete and we price the derivative in a risk-neutral framework by taking the expectation under the risk-neutral probability Q.

Denote the solution to the above problem by $P^* = P(S_0; \sigma)$, where σ is the volatility of the source price sequence S_t. Noting that the price-provider inputs a price optimally based on all of his information from a centralized market and/or from the decentralized world.

1.1 Arbitrageur

The price-provider writes an American option when he inputs a price K. It seems that anybody can exercise the option without any cost. However, the NEST requires that the one (arbitrageur) who exercises the derivative must input another price and lock in as much as β times the original asset requirement. In other words, to exercise one option, the arbitrageur

[1]For the NEST system, the time horizon T_0 actually is random because the time interval between two successive Ethereum blocks is. The framework in this note can be extended to study this case.

has to write β units of the same type of American options, where $\beta > 1$ is a specific multiplier.

One arbitrageur who wishes to make profit from the derivative can construct (sell) a portfolio in the outside market that replicates the derivative. Then the arbitrageur can make a risk-free profit the same as the value of the derivative. However, there is risk that the arbitrageur can not obtain the opportunity to exercise the derivative because it is competitive to take the arbitrage. Therefore, instead of making the risk-free profit, a realistic strategy is to make a *quick* profit in the sense of statistic arbitrage as follows.

The arbitrageur does nothing but waits until the difference between the outside asset price and the input price P is sufficiently large. Then he exercises the option and buys or sells in the exchange simultaneously to make money without any risk. Such an opportunity may not be available for all time, but in long time there are many chances. So statistically the arbitrageur can make money.

We calculate the following objective function for the arbitrageur:

$$\max_{\tau} E[(|S_\tau - P| - A)1_{|S_\tau - P| > A, \tau < T_0}], \tag{2}$$

where A represents all costs of the transaction, including Ethereum transaction fee and the value of the derivative multiplied by β. The stopping time τ in the above indicates that the arbitrageur will wait for the best time to take the arbitrage. However, considering the competitive environment, most likely, the profit is taken when the first time a target is reached. So the objective function turns to be

$$E[(|S_\eta - P| - A)1_{\eta \leq T_0}], \tag{3}$$

where $\eta = \inf\{t : |S_t - P| - A > \epsilon\}$ and ϵ is the minimum target profit of the arbitrageur. Along with the arbitrage-taking method (3), the corresponding loss (or the cost of inputing a price) of the price-provider is

$$E(|S_\eta - P|1_{\eta \leq T}).$$

The price-provider shall minimize the cost by choosing an appropriate K. That is, the objective function of the price-provider is

$$\min_P E[|S_\eta - P|1_{\eta \leq T_0}].$$

In fact, we should price it in a risk-neutral sense:

$$V^*(0) = \min_P E^Q[e^{-r\eta}|S_\eta - P|1_{\eta \leq T_0}],$$

where r is the risk-free interest rate. It yields that the price-provider can construct a portfolio in the outside market to hedge this derivative, so that his loss is a deterministic value same as V^*.

2. A Solution of the Model

Given the design of the NEST, we let

$$A = \beta V^*(\eta),$$

where $V^*(\eta)$ denotes value of the same derivative at time η. We let ϵ be the transaction fee in the blockchain (the gas fee).

Aware of the way the option is exercised, the price-provider actually considers the objective problem as follows.

$$V^*(0) = \min_P E^Q[e^{-r\eta}|S_\eta - P|1_{\eta \leq T_0}] = \min_P E^Q[e^{-r\eta}(A+\epsilon)1_{\eta \leq T_0}] = \min_P E^Q[e^{-r\eta}(\beta V^*(\eta)+\epsilon)1_{\eta \leq T_0}]. \tag{4}$$

We assume that the asset price follows a Brownian motion with drift:

$$S_t = S_0 + \mu t + \sigma Z_t,$$

where Z_t is a standard Brownian motion. Then $V^*(\cdot)$ is identical at any time. The recursive formula (4) is simplified (for a stationary solution under constant state variables μ and σ)

$$V^* = \min_P E^Q[e^{-r\eta} 1_{\eta \leq T}](\beta V^* + \epsilon). \tag{5}$$

Exploiting the density function of η, the first hitting time of Brownian motion, we can evaluate the expecation in (5) and solve for V^* and P^* numerically.

Set $\mu = r = 0$, $\epsilon = 0.003$ (the gas fee of one transaction in the Ethereum divided by 10 (ETHs)), $S_0 = 1$, we obtain the following results.

For $\sigma = 0.0001,\ 0.001,\ 0.003$ per second:

$\beta = 1.5$: $V^* = 0.0030, 0.0104, 0.0327$; probability of arbitrage: 0.0726, 0.3353, 0.3765

$\beta = 2$: $V^* = 0.0003$, 0.0092, 0.0291; probability of arbitrage= 0.0792, 0.4301, 0.4755,

$\beta = 3$: $V^* = 0.0002$, 0.0074, 0.0233; probability of arbitrage= 0.0894, 0.6064, 0.6696,

where the probability of arbitrage is defined by $E^Q[1_{\eta \leq T}]$. For all of these cases, the optimal input-price $P^* = S_0 = 1$. Since S_t is assumed to be a Brownian motion without a drift, this answer is obvious.

The sensitivity analysis regarding verification during time T, probability of arbitrage, β, volatility σ are shown in Figure 1 and 2.

2.1 Difference between NEST Price and Price of Exchange

By the preceding analysis, the difference between the NEST price and the price from an exchange is bounded by $a := \beta V^* + \epsilon$. Figure 3 indicates the upper bound can be as small as 0.003. The upper bound can be decreased if the transaction (arbitrage) cost in the blockchain becomes small. Alternatively, We may increase the asset requirement of inputing

a price to decrease the relative weight of ϵ. For example, if we increase the asset requirement to 50 ETHs, the difference bound turns to be 0.002 only.

Figures

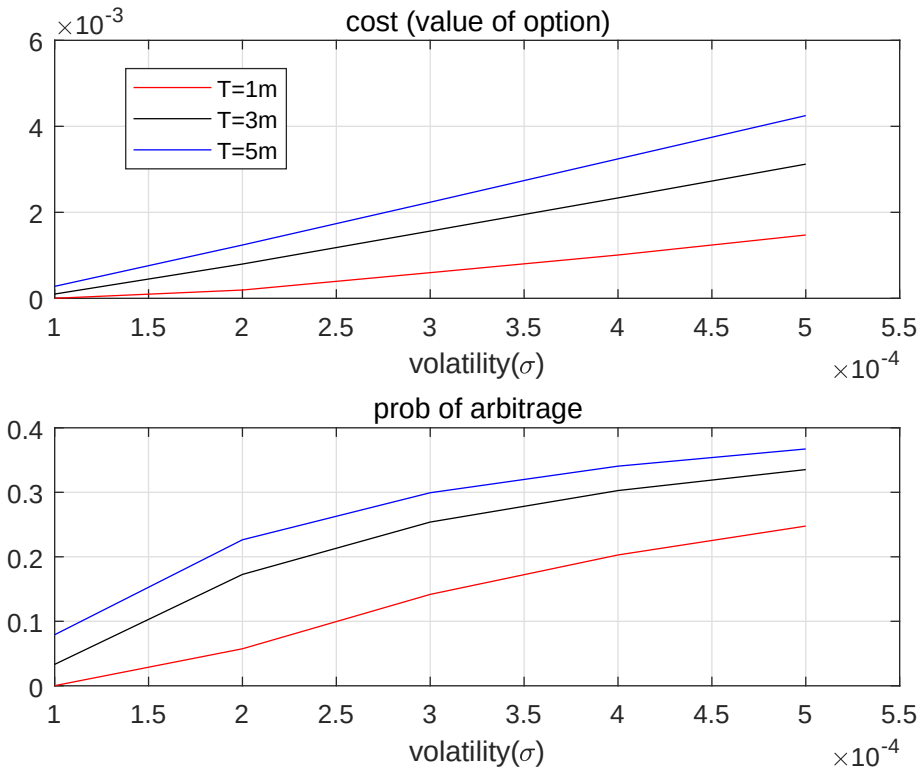

Figure 1. This figure depicts effects of volatility σ on cost of price-inputing and probability of arbitrage.

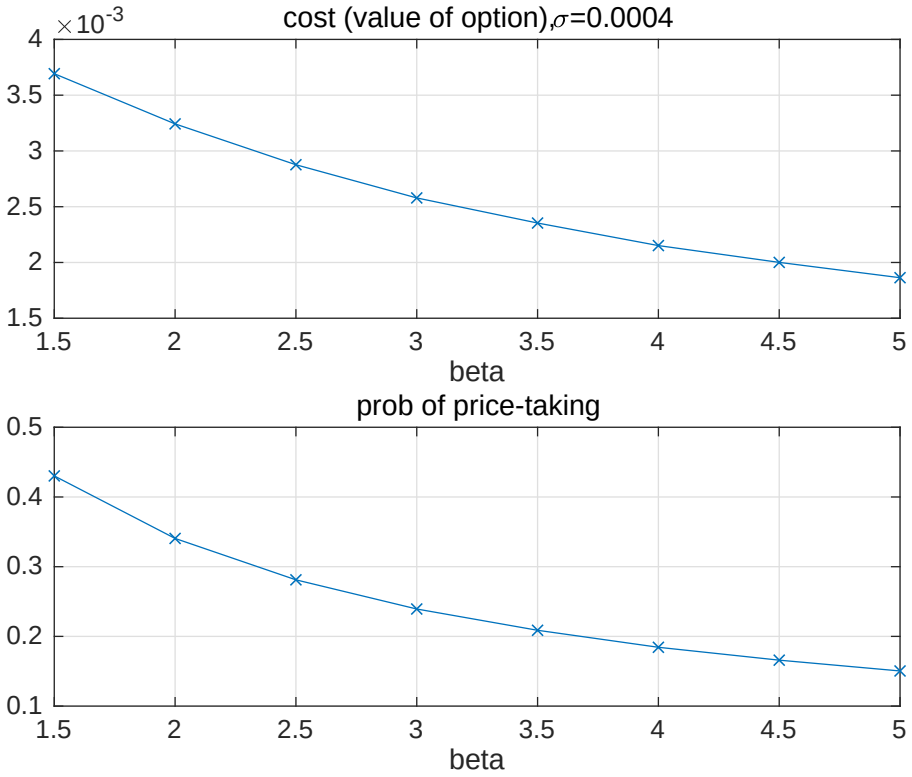

Figure 2. This figure depicts the effect of β on cost of price-inputing and probability of arbitrage.

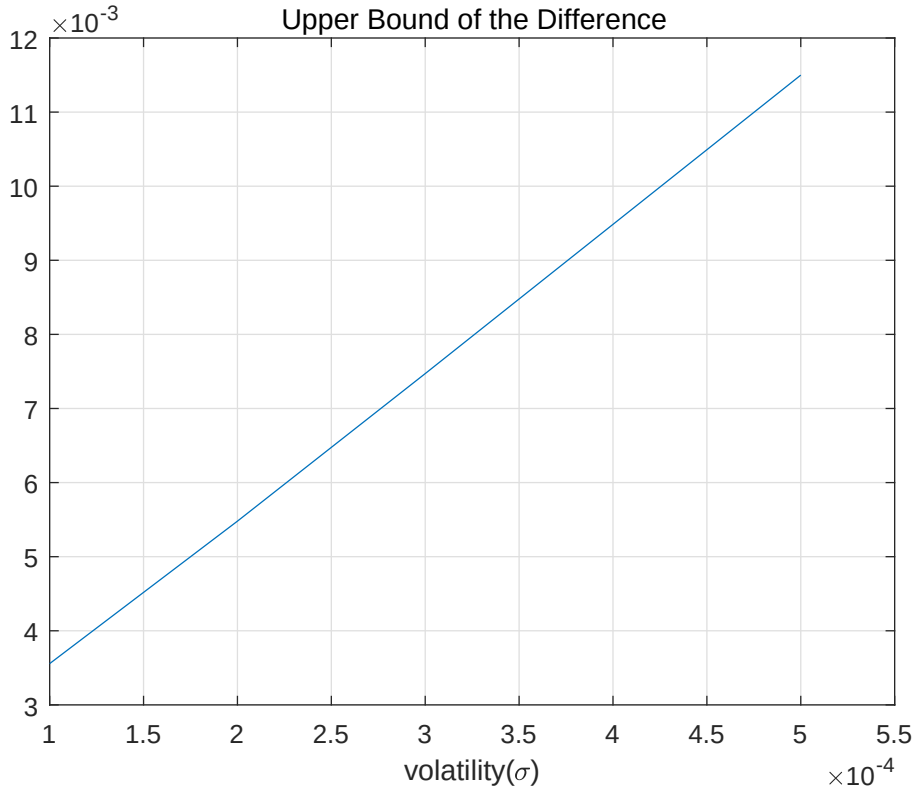

Figure 3. This figure shows the upper bound of difference between the NEST price and the price of an exchange at the same time.