Interactive coin offerings

Jason Teutsch
TrueBit Foundation
jt@truebit.io

Vitalik Buterin
Ethereum Foundation
vitalik@ethereum.org

September 24, 2017

Abstract

Ethereum has emerged as a dynamic platform for exchanging cryptocurrency tokens. While token crowdsales cannot simultaneously guarantee buyers both certainty of valuation and certainty of participation, we show that if each token buyer specifies a desired purchase quantity at each valuation then everyone can successfully participate. Our implementation introduces smart contract techniques which recruit outside participants in order to circumvent computational complexity barriers.

1 A crowdsale dilemma

This year has witnessed the remarkable rise of token crowdsales. Token incentives enable new community structures by employing novel combinations of currency rewards, software use rights, protocol governance, and traditional equity. Excluding Bitcoin, the total market cap of the token market surged over 60 billion USD in June 2017[1]. Most tokens originate on the Ethereum network, and, at times, the network has struggled to keep up with purchase demands. On several occasions, single crowdsales have consumed the network's entire bandwidth for consecutive hours.

Token distributions can take many forms. Bitcoin, for example, continues to distribute tokens through a competitive, computational process known as *mining*. In this exposition, we shall concern ourselves exclusively with the now typical situation in which an anonymous class of buyers wishes to purchase yet-to-be-generated ERC20 tokens over the Ethereum network in exchange for Ethereum's native currency. Unlike an equity distribution

[1] https://coinmarketcap.com/charts/

event in which prospective buyers can estimate share values based on existing and potential future revenue streams, tokens sales may not project any revenue at all. Since traditional analysis fails to estimate initial market valuation for new tokens, buyers must rely on new signals and methods for determining market prices. The token issuer, on the other hand, faces the unprecedented challenge of not knowing her buyers. In particular, she cannot tell whether or not two distinct purchasing addresses belong to the same person.

The Ethereum community has experimented with various sale configurations for ERC20 tokens. In a *capped sale*, for example, the project issuing the ERC20 token announces a fixed price for each new token as well as the maximum (and minimum) number of tokens to be sold. Capped sales can reach tens of millions of dollars and sell out in a matter of minutes, leaving buyers unable to participate, disappointed, and frustrated. Uncapped sales, which run without such maximums, provide buyers little clue as to the fraction of total tokens their contribution will ultimately purchase. Other distribution experiments, including hidden caps and reverse Dutch auctions, have suffered similar fates [1]. Indeed increasing purchase power and limited supply may cause buyers in a reverse Dutch auction to jump in too soon.

Recently, Buterin [1] distilled two desireable, mutually exclusive properties of crowdsales.

Proposition. *No token crowdsale satisfies that both:*

(I) *a fixed amount of currency buys at least a fixed fraction of the total tokens, and*

(II) *everyone can participate.*

Proof. If one unit of currency purchases at least p fraction of the tokens, then the total sale revenue cannot exceed $1/p$. □

Clearly any fixed valuation scheme cannot guarantee universal participation, however, we shall construct a crowdsale protocol such that, if each participant specifies a desired purchase quantity at each valuation, then the ultimate token cost to percentage ratio satisfies all buyers (with respect to both valuation and participation).

2 The bypass

Our interactive construction aims to establish an equilibrium of purchase amounts whose sum is satisfactory to all buyers at some uniform valuation.

Given that a liquid market for the new token does not exist prior to the crowdsale, we shall use the crowdsale process itself to reach a common view of the token's present value. Our protocol will be fair in the sense that parties endowed with large amounts of capital or Ethereum mining power cannot gain either participation advantage or lower cost per percentage of total tokens. Furthermore, incentives will counteract buyers' resistance to partake in an initially illiquid market. Lastly, we remark that the crowdsale is a game of perfect information in that the public has guaranteed access to all sales data.

Our approach distinguishes itself from prior crowdsale distributions in that:

1. buyers can *withdraw* their contributions after committing them to the sale (within certain limits), and

2. the protocol exploits sophisticated bookkeeping capabilities of smart contracts.

The corresponding crowdsale is interactive in the sense that potential buyers may enter and exit the crowdsale based on behaviors of other buyers and in doing so tend the valuation towards a market equilibrium. The protocol also allows sufficient time for informal, social interactions.

The interactive crowdsale begins after the token issuer deploys a smart contract on Ethereum's blockchain. For the purposes of this discussion, a *smart contract* is a universally trusted machine that, over time, takes in and pays out Ethereum's native currency and deploys a newly generated token. A combination of messages and native currency from pseudonymous Ethereum *addresses*, each held by some potential buyer, algorithmically determines these payments and deployments. The smart contract thus effectively collects and retains the crowdsale's token balances. Through their respective addresses, buyers purchase with native tokens and eventually receive newly minted crowdsale tokens in return.

New messages from addresses to smart contracts are broadcast via *blocks* which occur at regular intervals (approximately every 15–20 seconds[2]). In this way, blocks measure the passage of time on the Ethereum network. The crowdsale's smart contract can take the current block number as input and therefore alter its behavior as a function of time.

We now describe the components of a simple, interactive coin offering. We shall present a more detailed specification in Section 4.

[2] https://etherscan.io/chart/blocktime

Basic step: In each block epoch, buyers can either purchase tokens or voluntarily withdraw funds from the crowdsale. Buyers specify a maximum sale valuation at which they are willing to participate, and if the sale amount ever reaches this personal threshold, the buyer's *bid* is canceled and she receives a refund. In Section 7, we add support for bid activation triggered by sale lower bounds.

Withdrawal lock: After a certain number of blocks, voluntary withdrawals are no longer permitted. In a 30-day crowdsale, for example, the smart contract might permit voluntary withdrawals during the first 20 days, but during the last 10 days, only automatic withdrawals are allowed.

Inflation ramp: Buyers who purchase tokens early receive a discounted price. The maximum bonus might be 20% (a typical amount for crowdsales today). The bonus decreases smoothly down to 10% at the beginning of the withdrawal lock, and then disappears to nothing by the end of the crowdsale.

Individual buyers may submit multiple bids in the crowdsale in order to indicate distinct bid amounts for various valuations. In particular, they may choose personal thresholds exceeding the total amount of currency in circulation in order to guarantee a successful bid. The time prior to the withdrawal lock provides an opportunity for buyers to calibrate their purchase amounts, and the period after the withdrawal lock pushes the sale valuation to converge towards an equilibrium value. The inflation ramp reduces entrance inertia and encourages formation of a liquid market.

3 Cast of characters

We shall assume certain uniform characteristics among crowdsale participants which will enable us to derive crowdsale invariants in Section 5.

3.1 Buyers

Buyers express their individual participation goals by submitting a series of bids, each of which constitutes a "step."

Definition. A buyer's *valuation table* is a piecewise step function from the crowdsale's total sale amount to the buyer's contribution amount.

Buyers may wish to purchase tokens for many different reasons, and we make no specific assumptions in this regard. For the purposes of our model, all buyers satisfy the following properties.

1. *Demand is inverse to supply.* The total sale amount affects individuals' inclinations to contribute, and in a liquid market, buyers will purchase at least as many tokens at lower valuations as they will at higher ones. In Section 7, we shall relax the latter part of this assumption and modify the core procotol of Section 4 accordingly. We explicitly do not assume that individuals agree on the volume of trade that constitutes a liquid market nor on the valuation threshold(s) at which contribution inclinations decline.

   We shall show in Section 5.1 that each buyer's cumulative purchases ultimately and effectively converge to a monotonically decreasing valuation table under our assumption that buyer demand decreases at higher valuations. Buyers failing to match this profile would complicate our protocol and analysis. First, accommodating and monitoring lower entry bounds in the protocol places additional computational stress on the crowdsale smart contract. In Section 7, we discuss an efficient workaround. Second, buyers can anyway achieve an effect similar to a non-decreasing valuation table by manually purchasing additional tokens later in the sale once the total sale amount has reached the target threshhold. If sufficiently many buyers were to apply this strange strategy, however, the network might become congested from a positive feedback loop, and then these strategies would fail to execute properly.

2. *Preference for liquid markets.* Buyers have intrinsic inertia against entering a new crowdsale. Tokens held by few owners may be difficult to exchange and therefore have uncertain value. Given the risks of purchasing first, and barring other incentives, most buyers prefer to wait for others to purchase before they do. Waiting times may vary from buyer to buyer. We discuss other possible sources of inertia and their circumvention in Section 7.

3. *Reliance on social influences.* Buyers depend on social influences to make purchase decisions. Since the immediate value of new tokens depends largely on others' beliefs, buyers necessarily interact, either directly or indirectly, with other buyers. At the beginning of a crowdsale, Buyers lack reliable information with which to valuate the new

token. Social gossip moves at a much slower pace than pure algorithmic trading and consequently dictates the crowdsale pace. Finally, we assume that social interactions will lead each buyer to eventually, but well before the end of the crowdsale, converge to a final valuation table.

4. *Preference for simplicity.* Complex procedures for purchasing tokens decreases participation. The tolerable threshold varies from buyer to buyer, and particular sets of rules or steps may encourage or discourage certain types of buyers.

5. *Pseudonymity.* Buyers need not disambiguate their identities in order to participate in the crowdsale. In fact, we expect each buyer to compose her valuation table with bids from multiple pseudonymous addresses.

3.2 Adversaries

We define an *adversary* to be any entity which performs network actions, including purchases and withdrawals, in order to decrease his cost per percentage of total tokens. We assume that the adversary has significant financial and mining resources, but not enough to create an extended denial-of-service attack which prevents other bids from entering the crowdsale. In particular, we shall assume that the amount of time that the adversary can sustain significant congestion or censorship on the network is negligible with respect to durations of the crowdsale. We also assume that the adversary restricts his actions to the Ethereum network. He cannot, for example, physically restrain other buyers who wish to participate in the crowdsale. Finally, we assume that the crowdsale smart contract always processes bids correctly.

4 ICO protocol

We now describe the operations of the crowdsale's smart contract. For the purposes of this presentation, *valuation* of the crowdsale refers the total value of tokens sold with respect to the native currency as opposed to the value of the total number of tokens generated. We shall assume that any tokens created but not sold in connection to the crowdsale event represent a fixed fraction of the total number of tokens generated. Therefore, regardless of valuation, a given fraction of crowdsale tokens represents a fixed fraction of the total tokens generated.

As new bids enter the crowdsale (Step 1 below), the protocol nullifies active bids with minimal personal caps (Step 3). Step 3.3 issues partial refunds in order to enforce a monotone valuation invariant (see Section 5.2). Consequently, in case several bids tie for minimal personal cap, the protocol refunds an equal fraction of each. Buyers may stagger their bid values so as to avoid such ties, however, the monotone valuation invariant persists even without such action.

Initialization. All addresses have "inactive" status. Fix time thresholds $0 \leq t < u$, where t corresponds to the "withdrawal lock" threshold of Section 2, and set $s = 0$. Let $p(s)$ be a positive-valued, decreasing function representing the purchase power of a native token at stage s. Finally, define the *crowdsale valuation at the present instant* as follows.

$$V = \begin{cases} 0 & \text{if no addresses are active;} \\ \sum_{A \text{ active}} v(A) & \text{otherwise.} \end{cases}$$

Here A is an address, and $v(A)$ is a function mapping addresses to quantity of tokens as specified below.

Main Loop. The following four steps are repeated in each block while $s \leq u$, where u delimits the end of the crowdsale.

STEP 1: RECEIVE BIDS.

1. Any "inactive" address A may send to the crowdfund smart contract:
   - a positive quantity of native tokens $v(A)$ along with
   - a positive-valued *personal cap* $c(A) > V$.
2. The smart contract then
   - sets the address balance $b(A) = v(A) \cdot p(s)$, effectively implementing the inflation ramp (Section 2), and
   - sets A's status to "active."

STEP 2: VOLUNTARY WITHDRAWALS (IGNORE THIS STEP IF $s \geq t$).

The following only applies prior to the withdrawal lock at time t. Any "active" address A may signal that it wishes to cancel its bid from any previous stage. Upon such signal, the crowdfund smart contract does the following:

1. refunds $v(A)$ native tokens back to A, and
2. sets A's status to "used."

Step 3: Automatic withdrawals.

While there exists an active address B whose personal cap is exceeded by the present crowdsale valuation, i.e. $V > c(B)$, repeat the following three steps. We update the value V only after the dashed bullets in each iteration of the while loop, regardless of whether or not (4.1) holds.

1. Let $B_1, \ldots, B_k$ be the (distinct) active addresses with minimal personal cap at the present moment, i.e.

$$c(B_i) = \min\{c(A) : A \text{ is active}\}$$

for all $i \leq k$, and let

$$S = \sum_{i=1}^{k} v(B_i).$$

2. If removing the bids of $B_1, \ldots, B_k$ does not suffice to satisfy all personal caps from active addresses, i.e.

$$V - S \geq c(B_1), \tag{4.1}$$

then the crowdsale smart contract kicks out the entirety of these bids. In more detail, the smart contract:
   - refunds $v(B_i)$ to B_i for all $i \leq k$, and
   - sets the statuses of $B_1, \ldots, B_k$ to "used."

3. Otherwise, the reverse inequality of (4.1) holds, and only some fraction of each of $v(B_1), \ldots, v(B_k)$ comes out of the crowdsale. Let $0 < q < 1$ be the minimum (positive) fraction of these quantities that must be removed in order to satisfy all remaining personal caps, i.e.

$$q = \frac{V - c(B_1)}{S}.$$

For all $i \leq k$, the crowdfund smart contract:
   - refunds $q \cdot v(B_i)$ to B_i,
   - sets the new value of $v(B_i)$ to be $(1-q) \cdot v(B_i)$, and
   - sets the new value of $b(B_i)$ to be $(1-q) \cdot b(B_i)$.

The B_i's remain "active," and the total crowdsale valuation is now exactly $c(B_1)$ because the procedure above removes exactly qS native tokens from the crowdsale's smart contract.

STEP 4: Increment s.

Final Stage. Each "active" address A receives $b(A)$ tokens at the end of the crowdsale.

Note that the loop in Step 3 eventually terminates because V decreases with each iteration while $\min\{c(B) : B \text{ is an active address}\}$ increases.

5 Analysis

We conclude by highlighting some properties of the interactive coin offering protocol detailed in Section 4. Our analysis relies on two, key, quantitative invariants, namely that valuation is monotonically increasing over time, and that all personal caps remain above the current valuation in each block.

5.1 Personal cap invariant

We argue that the final crowdsale valuation and purchase amounts satisfy every buyer's valuation table (see Section 3.1). In order to parse this statement, we need to explain two things:

1. how the buyer's cumulative purchases from various addresses formally correspond to a valuations table, and

2. what it means for a crowdsale to "satisfy" a valuation table. Let V be the final valuation of the crowdsale.

Regarding item 1., note that net purchase effect of each bid from an address A is a single-step valuation table (modulo a single point):

$$\mathcal{A}(V) = \begin{cases} v(A) & \text{if } V \in [V_0, c(A)); \\ \text{some value in } [0, v(A)] & \text{if } V = c(A); \\ 0 & \text{otherwise.} \end{cases}$$

Here V_0 is the crowdsale valuation at the time the address generates the bid. We cannot specify a definite value for $\mathcal{A}[c(V)]$ because the purchase amount at this valuation depends on the bids made by other addresses. A could either receive a partial refund in Step 3 of Section 4, or none at all. The buyer's cumulative purchase is determined by the sum of his bids. In other words, the buyer's (stepwise) valuation table equals the sum of the single-step valuation tables for his addresses, which answers item 1. As

noted in Section 3.1, the valuation table graph restricted to the right of the last bid made by the buyer is a monotonically decreasing function because single-step bid functions may end but not begin after that point.

We now turn our attention to item 2. We say that a valuation V and purchase amount a satisfy a buyer's valuation table T if the following holds:

(a) $a = T(V)$ if V is an interior point of some valuation table step; otherwise

(b) a equals some value between the left and right limits of $T(x)$ as x approaches V.

There are two cases to consider. If the final valuation V from the crowdsale does not equal $c(A)$ for any of the buyer's active addresses A, then V is not a limit point of any of the single-step valuation tables for the buyer's addresses, and $T(V)$ is simply the sum of the single-step valuation tables evaluated at V. In this case, $T(V)$ matches the purchase amount exactly, satisfying (a) above. Otherwise, V matches the personal cap for some address(es). Then V is a limit point of some step on the buyer's valuation table, and the actual purchase amount is the sum of $v(A)$ over active addresses A whose personal cap does not equal V plus some fraction of the $v(A)$'s for addresses A whose personal cap matches V, yielding (b) as desired.

In summary, the crowdsale satisfies the buyer's valuation table which closely resembles the sum of her bids.

5.2 Monotone valuation invariant

We shall demonstrate that regardless of what bids buyers submit to the crowdsale smart contract, valuation is monotonically increasing after the distinguished time threshold t. After time t, voluntary withdrawals do not occur, so we need only consider the other protocol steps. By the loop condition in Step 3, we may assume that the valuation at the beginning of Step 1 is greater than or equal to the personal cap for each active address. The active addresses in each iteration of the loop in Step 3 are a subset of those addresses which were active at the end of Step 1, and furthermore the valuation at the end of Step 3 is no less than the personal cap of every active address, whether the loop terminated after Step 3.2 or Step 3.3. To recap, the valuation at the beginning of Step 1 is less than or equal to the minimum of all active addresses at that time, which is less than or equal to the valuation at the end of Step 3, which proves the claim.

The invariant property above allows the interactive coin offering to resist "pushout attacks" from rich buyers, or *whales*, of the following form.

Suppose there are two existing bids purchasing 30 tokens each, and each bid has a personal cap of 79 tokens. Now say that a whale bids a 50 token purchase with a large personal cap of 200 tokens. The total of all bid amounts would now be 110 tokens, exceeding the personal caps of the original two bids. If those two bids were to come out, the whale would have a bid with a valuation of 50, which is lower than the original valuation of 60 tokens. The invariant above proves this cannot happen.

5.3 Fairness

The protocol treats large purchases and small purchases equitably. Whales with low personal caps get pushed out of the crowdsale in just the same way as buyers who purchase a fraction of a token. Furthermore, any buyer can specify any non-trivial personal cap that they please, and the fees for submitting transactions to the crowdsale smart contract are flat fees based on Ethereum gas prices (See Section 6). Finally, the smart contract handles all purchases publicly, which means that all prospective buyers have perfect information about all other bids.

5.4 Censorship

In the past, whales have benefited from network congestion during capped crowdsales. One BAT crowdsale buyer, for example, paid $6660 towards a single transaction fee to ensure that his transaction entered the current block, effectively preventing others buyers from participating [1]. In theory miners can potentially mimic or amplify this bias by censoring transactions during the crowdsale. While these types of denial-of-service attacks may succeed in quick crowdsales, they become impractically costly over extended crowdsales, such as the one presented here.

5.5 Last minute withdrawls

A whale who bid a huge number of tokens but then withdrew his purchase in the last block of the crowdsale could deter other buyers from participating and thereby obtain an artificially low valuation. For this reason, the protocol forbids voluntary withdrawals in the later part of the crowdsale. By prohibiting last-minute actions, the protocol increases the chance of converging to stable bids prior to the end of the crowdsale.

5.6 Overcoming inertia

The protocol design encourages early participation and maintains a low barrier to crowdsale entry. An inflation ramp (Section 2) incentivizes buyers to enter the crowdsale early and form a liquid market. Moreover, the beginning of the crowdsale offers a low-risk trial period in which buyers can withdraw their purchase bids at the negligible cost of Ethereum gas fees. Finally, the crowdsale has a relatively simple user interface. Anyone who simply wishes to purchase tokens without risking automatic withdrawals can submit a transaction to the crowdsale's smart contract with a personal cap exceeding the total number of native tokens in circulation (a predictable value in Ethereum).

6 Lightweight implementation

In Section 3.2, we made the simplifying assumption that "crowdsale smart contracts always process bids correctly." We now refine this simplistic point-of-view. Computational tasks of minimal complexity, that is, those quantitatively resource bounded by Ethereum's per block *gas limit*[3], execute correctly so long as enough *gas*, or block founder payment, accompanies the transaction which initiates the task. Smart contracts themselves may call tasks so long as:

1. the task itself runs within the per block gas limit (and available network bandwidth),

2. the smart contract has sufficient *ether*, or native currency, to pay for the task execution, and

3. the smart contract remains dormant between the blocks in which users interact with it.

The ICO protocol's main loop (Section 4) requires maintenance of a list of addresses with various personal caps, a way of finding the set of addresses with minimal personal caps, and a mechanism for deleting this set from the list. Traditional heap data structures require $O(\log n)$ time to execute one or more of these operations on a list of size n. Hence if the address list grows sufficiently large, then the smart contract cannot maintain the heap without violating item 1 above. Moreover, per item 2, who pays for each of these operations and when? The autonomous crowdsale smart contract could

[3] https://ethstats.net/

become increasingly expensive over time. It will be clear from inspection that our construction below satisfies item 3.

Given the infeasibility of maintaining a heap data structure within the crowdsale smart contract, one might wonder whether the smart contract could recruit outside parties to perform the necessary heap maintenance through incentives. Outside parties, however, might supply incorrect data. How would the smart contract know whether the supplied address actually has a minimal personal cap? The smart contract's inability to verify and delete the actual minimum opens a potential attack vector. Suppose there exists a bid of 50 with personal cap 60, a bid of 35 with personal cap 80, and that the current valuation is 115. If an attacker reports that 80 is the minimum, active personal cap, then the $(35, 80)$ bid would come out entirely while part of the $(50, 60)$ bid remains active. According to the protocol specification, however, the entire $(50, 60)$ should come out while instead the $(35, 80)$ bid remains completely active.

We mitigate against the above attack by maintaining the active bids in a sorted *linked list*, a data structure wherein each element in the list contains a pointer to its successor. Bids with minimal personal caps occur at the start of the list, so the smart contract can quickly find and/or delete them. While an insertion operation in a sorted linked list requires $O(n)$ time, the smart contract can easily check its correctness. The new element must simply have a personal cap greater than or equal to the personal cap of its inbound neighbor and less than or equal to its outbound neighbor. Thus we satisfy item 1.

Upon receipt of a new bid, the smart contract offers a public reward for anyone who correctly indicates where to insert it into the linked list. The buyer who submits this bid also includes a small payment to incentivize this insertion. For the purposes of determining the incentive, we assume that the local, off-chain $O(n)$ operation requires negligible computation time and that the price of ether is sufficiently stable that a constant incentive amount suffices. Finally, the smart contract executes the automatic withdrawals as described in Step 3 of Section 4. The incoming bid(s) cover the gas costs for this action as well. The smart contract has a hard-wired upper bound on the maximum number of bids deleted/reduced per block in order to ensure a bound on the gas cost. This bound ensures a limit on the incoming bid's fixed costs and guarantees that the deletion step fits within Ethereum's gas limit, thereby satisfying item 2.

7 Personal minimums

Funding inherently makes a project more valuable. Starting costs vary but may include personnel, legal fees, marketing, office space, licenses, travel, equipment, or administration. Buyers may perceive substantially higher risk with tokens associated with an underfunded project. Furthermore, they may not agree on the fixed cost required to get a venture off the ground. A crowdsale which permits each buyer to specify a *personal minimum* above which she would wish to participate therefore reduces this entrance risk. Below we describe a practical mechanism for realizing personal minimums.

Suppose that a crowdsale smart contract receives 3 bids, each with capital contribution 10 and personal minimum 30. While the crowdsale valuation may never reach the activation thresholds of these three bids, each bid would gladly activate were the other two bids to activate first. How can a crowdsale smart contract recognize this relationship amongst a deluge of other submitted bids?

We describe an incentivized, method for monitoring personal minimums which functions with limited gas resources. In short, the crowdsale smart contract rewards users who submits a *target valuation* x and a *target set* of bids such that:

1. x exceeds all personal minimums of bids in the target set, and
2. the sum of capital contributions from bids in the target set exceeds x.

The target set here may include all active bids, but for the purposes of nontriviality, it must include inactive bids as well. The two properties above suffice to justify the activation of all (inactive) bids in the target set, and the smart contract can easily verify these conditions. We append this operation to Step 1 in Section 4.

Each bid with a personal minimum includes a flat fee to pay for outside users to *poke* it in via the operation above. A simultaneous poke of five bids, then, would collect the sum of fees from each of these five bids. The rewards should suffice to compensate for the fact that multiple pokes with identical pairs of target valuation and target set may occur and that only the first receives the reward. Note that the computational complexity of identifying a target valuation and corresponding set grows as the set of bids grows, but for crowdsales with less than a million bids, local, off-chain compute times remain negligible. Finally, we remark that an SSTORE call in Ethereum costs 5000 gas, and therefore, with the current gas limit of 6.7 million, one could poke in as many as 1300 bids in a single transaction.

8 Conclusion

Crowdsales pose critical, timely, and challenging game-theoretic questions. While reasonable assumptions and deductive reasoning can guide our intuition, ultimately we must also rely on empirical evidence to arrive at definitive cryptoeconomic conclusions. The protocol discussed herein offers a means for achieving fair valuation equilibrium.

Acknowledgments. The authors are grateful to Christian Reitwießner for help with implementation details. We also thank Ryan Zurrer for useful comments.

Reference

[1] Vitalik Buterin. Analyzing token sale models. http://vitalik.ca/general/2017/06/09/sales.html, June 2017.