Algorand Pipelined Auctions

Jing Chen and Silvio Micali

The general context of this document is provided by the [Proposal for Decentralizing Algorand Governance.](#) Familiarity with that document may help the reader with the terminology and the setting of this one. This said, the material presented here is of quite independent interest. The basic governance auction is a blockchain implementation of a Dutch auction. (The fact that the currency involved are tokens rather than, say, US dollars; that the auction establishes a governance-reward rate, rather than, say, the interest rate of a bond; or that the winners of the auction lock their tokens and participate in governance, rather than, say, lend their money, is quite tangential. The underlying auction mechanism is the same.)

We distinguish two types of unlocked accounts (*U accounts,* for short) and governing accounts (*G accounts* or *governors*, for short) that are locked for a given period --- herein, one year.

NOTE. There may be locked accounts other than G accounts - for instance, accounts that have been locked as part of the execution of a smart contract. G accounts are locked only because they have won some governance at Dutch auction.

1. The Basic Governance Auction

The basic governance auction is a Dutch auction open to all U accounts, and only to U accounts. The goal of such a basic auction is to establish:

(i) a new set of governors, and

(ii) the governance reward rate, Grr, the new governors enjoy.

(As discussed, the locking period for a governor is always assumed to consist of one year.)

A basic governor auction is organized by the Foundation and specifies:

1. the *reward pool* --- i.e., the total number of governance reward tokens, R, from the Foundation's treasury that it is ready to distribute to the governors during the year in which they are locked;

2. the number of stages of the auction, k; and

3. for each stage s, a stage reward rate $r_s \in (0,1]$, so that $r_1 < r_2 < \cdots < r_k$.

Each stage i corresponds to a separate sequence of blocks of the blockchain. For instance, stage 1 corresponds to some 100 consecutive blocks; stage 2 to the next 100 blocks, and so on.

The auction is played by placing bids in each stage, and by its end produces the set of new governors and the rate Grr.

NOTATION. At any point in time during the execution of the auction, for each U account x,

- $algos_x$ is the number of tokens then owned by x;
- $committed_x$ is the number of tokens x has already committed to be locked, if it wins the auction; and
- $available_x$ (= $algos_x - committed_x$) is the number of x's tokens still available to be committed.

Stage *i*. *In stage i, a valid bid of a U account x consists of a digital signature $SIG_x(a, i, \text{commit})$, where*

- $a \leq available_x$ and
- $SIG_x(a, i, \text{commit})$ is posted in any of the 100 blocks associated to stage i.

Such a digital signature further includes information (such as the auction number, etc.) that prevents the signature from being "played back" later. For simplicity, we omit explicitly specifying this information.

COMMENT. In stage i, x may also commit to lock a total of a available tokens by means of multiple bids. That is, x may first commit to lock a_1 of its available tokens, then a_2 of his remaining available tokens, and so on. Each such posting publicly resets the values of both $available_x$ and $committed_x$.

CLARIFICATION. Throughout the duration of the auction, an account x cannot post a transaction involving any of its already "committed" tokens. Such tokens are, in essence, put in escrow during the mechanism execution.

NOTATION. At the end of a stage i, considering all unlocked accounts x and their valid bids, set

$$C_i = \sum_x committed_x.$$

Auction end

If there is a stage i such that $C_i \cdot srr_i \geq R$, then the first such a stage is the final stage, f. Else, $f = k$. That is, if there is no such stage i, then the final stage is stage k.

At the end of f, $Grr = srr_f$. Moreover, each unlocked account x is automatically split into two separate accounts both controlled by x, an ordinary U account x_1 and a G account x_2, as follows

- *If $C_f \cdot srr_f = R$, then x_1 has $available_x$ tokens and x_2 has $committed_x$ tokens (locked for one year)*
- *Else (i.e., $C_f \cdot srr_f > R$) prespecified tie-breaking rules determine the number of tokens of x_1 and x_2.*

More precisely, all the tokens committed by x in the first $f - 1$ stages belong to x_2 and tie-breaking rules determine which of the tokens committed by x in stage f belong to x_2. All other tokens of x belong to x_1.

For example, one may use *first-come-first-serve* tie-breaking rules. In this case, letting the number of tokens committed in the first $f-1$ stages be $R-n$, after ordering all commitment posted in the 100 blocks (by block first, and then by posting position within each block), the first committed tokens will be owned by G accounts, and the other tokens by U accounts.

Remarks

Stage-i blocks.

Having 100 consecutive blocks correspond to stage i guarantees that, with high probability, most proposers of these blocks will be honest, and thus that (since honest proposers do not censor any valid transaction they see) hundreds of thousands of valid stage-i bids can be included in the 100 blocks corresponding to stage i.

Bidding options.

A user may very well decide up-front all the bids she intends to place, leaving her bidding application in charge of monitoring the blockchain and propagating her stage-i bids in time to be included in the blocks devoted to stage i. On the other hand, the user may prefer bidding in real time, as the auctions develops, in which case her bidding application should help her by providing her with accurate and timely information about the auction execution, such as the total number of currently committed tokens.

2. Pipelined Blockchain Auctions

It is important to be able to conduct governance-reward-rate auctions frequently (e.g., once a month) while keeping the locking period of their winners to be one-year long. A main reason to do so, in our Algorand setting, is that new tokens enter the ecosystem all the time. We must thus allow these tokens to participate in governance as soon as possible, without having to wait for the next year.

We must also be cognizant, however, that running several Dutch auctions a year, each with its own pool of reward tokens, may complicate the bidding account owners' strategic thinking and cause them to act in some unintended way.

For instance, why should an account lock its tokens in the Dutch auction on, say, February 1st, if it believes that it will earn a higher governance reward rate in the Dutch auction on March 1st?

To prevent such strategic complications, we have architected a way to enable an account that has already locked its tokens in a prior auction to participate to a later auction and, if it wins, to *"seamlessly switch"* some or all of its already locked tokens to the new locking period and new governance reward rate.

Let us now explain how to modify the discussed implementation to enable auction pipelining.

THE NEW SETTING

Pipelined GRR auctions continue to be Dutch auctions. As before, a pipelined auction specifies its own reward pool (for which the number of tokens is still denoted by R), its number of stages (still denoted by k), and its stage reward rates ($r_1 < r_2 < \cdots < r_k$). There is, however, a main difference:

- *Participants.*
  All U accounts as well as all G accounts (still locked by a previous GRR auction) may participate to a new GRR auction.

Moreover, we insist on:

- *Separate execution periods.*
  The sequence of consecutive blocks devoted to the stages of two different pipelined auctions have no block in common. (This separation may be bypassed but simplifies handling the participation of already locked accounts. This is so because it ensures that each locked account can participate in a single pipelined auction at a time.)

NOTATION.

During the execution of a given GRR auction, for each (U or G) account x,

- $tokens_x$ still is the number of tokens then owned by x;
- $committed_x$ still is the number of tokens of x has already committed to be locked; and
- $available_x$ (= $tokens_x - committed_x$) still is the number of x's available tokens.

The set of all governors is denoted by G. For each governing account $g \in G$, let A_g be the auction in which g became a governor. Then, at each point in time,

- FRC_g, the *future reward of the committed tokens of* g, is the number of reward tokens that the committed tokens (in the present auction) of g stand to receive in the remainder of g's locking period in A_g according to the governance reward rate of A_g.

That is, $FRC_g = committed_g \cdot r_g \cdot RLP_g$ where:

- r_g is the governance-reward rate g is enjoying thanks to auction A_g; and
- $RLP_g \in [0,1)$, the *remaining locking period of* g, is the fraction of the year for which g is locked due to auction A_g.

COMMENT. With governance auctions occurring at most once a month, RLP_g (and thus FRC_g) can be considered a constant because the length of an auction is much smaller than the possible remaining locking period of g (indeed, a few hours vs. one month). In any case, RLP_g (and thus FRC_g) can be always recomputed at any given point in time during the current auction.

Stage *i*. *In stage i of a pipelined auction, a valid bid of a U or G account x still consists of $SIG_x(a, i, \text{commit})$, where $a \leq available_x$.*

At the end of a stage i, considering all U and G accounts x and their valid bids,

$$C_i = \sum_x committed_x.$$

Auction end. *If there is a stage i such that*

$$C_i \cdot srr_i \geq R + \sum_{g \in G} FRC_g,$$

then the first such a stage is the final stage, f. Else, the final stage is $f = k$.

At the end of f, it is still the case that $Grr = srr_f$. Moreover,

IF $C_i \cdot srr_i = R + \sum_{g \in G} FRC_g$, *, then:*

1. *Each U account x is still automatically split into two separate accounts: a U account x_1 with $available_x$ tokens and a G account x_2 $committed_x$ tokens; and*

2. *Each G account g is automatically split into two separate G accounts, g_1 and g_2, where:*
   - *g_1 has $available_g$ tokens and continues to be locked according to A_g, that is, with the same governance-reward rate and the same locking period of A_g;*
   - *g_2 has $committed_g$ tokens and is locked in the current auction A, that is, for one entire year and with governance-reward rate srr_f.*

ELSE (i.e., if $C_i \cdot srr_i > R + \sum_{g \in G} FRC_g$) *then prespecified tie-breaking rules determine the number of tokens of x_1 and x_2 for each U account x, and those of g_1 and g_2 for each G account g.*

More precisely, all the tokens committed by x and g in the first $f - 1$ stages respectively belong to x_2 and g_2, while tie-breaking rules determine which of the tokens committed by x and g in stage f, if any, respectively belong to x_2 and g_2. All other tokens of x and g respectively belong to x_1 and g_1.

Note, however, that having different classes of accounts participating to the auction allows for more tie-breaking rules. For instance, such rules may give precedence to G accounts (and, within such accounts, to G accounts that have been locked for longer).

REMARK. In a sense, in a pipelined auction A, each bid of a tokens of a G account $g \in G$ "enriches" the reward pool of A with the reward tokens that those a tokens of g stood to receive according to A_g.

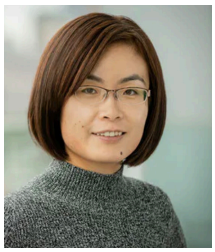
3. Generalizations

We have focused on a specific implementation of pipelined Dutch auctions, but other variants are possible within the same framework. The notion of switching from one Dutch auction to the next, by canceling the terms and conditions of a first auction and embracing those of a second auction in case of a winning bid, informs all such variants. Similar principles apply to auctions other than Dutch ones.

Also, when money, bound by the terms and conditions of a first auction, is used in a winning bid B of a second auction, we have presented a specific way of utilizing in the latter auction the rewards that such money was going to receive for the remainder of the term of the first auction. Other uses of these "saved" rewards are certainly possible in the second auction, within the discussed framework. For instance, (1) the saved rewards may not affect at all the termination condition of the second auction; (2) they may enrich the reward pool of the second auction in another way; (3) they may be used to benefit especially bid B; or (4) a combination thereof.

Also, the rewards "saved" in the first auction could be used in some future auction (rather than in the second auction), used elsewhere, or returned (to the Foundation in the current main example).

The specific usage discussed here was to satisfy additional goals in our envisaged application of pipelined auctions.



JING CHEN | Head of Theory Research and Chief Scientist

Jing is an Assistant Professor in the Computer Science Department at Stony Brook University. She is also an Affiliated Assistant Professor in the Economics Department and an Affiliated Member of the Stony Brook Center for Game Theory. Her main research interests are distributed ledgers, game theory, and algorithms. Jing received her Bachelor and Master degrees in Computer Science from Tsinghua University, and her PhD in Computer Science from MIT. She did a one-year postdoc at the Institute for Advanced Study, Princeton. Jing received the NSF CAREER Award in 2016.

SILVIO MICALI | Founder, Algorand

Silvio Micali has been on the faculty at MIT, Electrical Engineering and Computer Science Department, since 1983. Silvio's research interests are cryptography, zero knowledge, pseudorandom generation, secure protocols, and mechanism design and blockchain. In particular, Silvio is the co-inventor of probabilistic encryption, Zero-Knowledge Proofs, Verifiable Random Functions and many of the protocols that are the foundations of modern cryptography.

In 2017, Silvio founded Algorand, a fully decentralized, secure, and scalable blockchain which provides a common platform for building products and services for a borderless economy. At Algorand, Silvio oversees all research, including theory, security and crypto finance.

Silvio is the recipient of the Turing Award (in computer science), of the Gödel Prize (in theoretical computer science) and the RSA prize (in cryptography). He is a member of the National Academy of Sciences, the National Academy of Engineering, the American Academy of Arts and Sciences and Accademia dei Lincei.

Silvio has received his Laurea in Mathematics from the University of Rome, and his PhD in Computer Science from the University of California at Berkeley.