

DELOC: A PROTOCOL FOR DECENTRALIZED EXCHANGE LIMIT ORDER CROSSING

GIACINTO PAOLO SAGGESE AND PAUL SMITH

CONTENTS

1. Introduction	2
1.1. Token swap as financial primitive	2
1.2. Limit orders as an interface to token swap	2
1.3. How exchanges create value	3
1.4. The advantages of DELOC	3
2. Matching liquidity	4
2.1. Breaking the symmetry	5
2.2. Basket case	5
3. Limit orders	5
3.1. Base/quote tokens	6
3.2. Exchange price properties	6
3.3. Order attributes and notation	6
3.4. Order clearing quantity and exchange rate	7
3.5. Execution of orders	7
3.6. Limit order as a set of inequalities	8
3.7. Order normalization	9
4. Reference clearing prices and prioritization	9
4.1. Lit exchange bid-ask midpoint reference price	9
4.2. Order crossing	10
5. Formulation of the general DaoSwap problem	10
5.1. Limit order inequalities	10
5.2. Supply and demand match	11
5.3. Combining the constraints	11
5.4. Solution of the general DaoSwap problem	11
5.5. Additional constraint 1: Unique clearing price per token pair	12
5.6. Additional constraint 2: No arbitrage	12
5.7. DaoCross problem as simplified DaoSwap problem	13
6. Fees and tokenomics	13
7. Implementation details	13
7.1. DaoCross and DaoSwap Architecture	13
7.2. Off-chain computation	13
7.3. Ensuring a timely solution	13
7.4. Performing the swap	13
8. Alternative solutions for token swapping	13
8.1. DaoSwap	14

Date: April 12, 2023.

With contributions from Samarth KaPatel, Grisha Pomazkin, Juraj Smeriga, Danya Tikhomirov, Nina Trubacheva, Dan Yachmenev, Vladimir Yakovenko, and Tamara Zhordaniya.

8.2. DaoCross	14
8.3. Limit Order Book (LOB)	15
8.4. Automatic Market Makers (AMM)	15
9. Comments on AMMs	16
10. CPMM as an LOB	17
10.1. Buy order	17
10.2. Supply liquidity	17
References	18

1. INTRODUCTION

DELOC¹ is a new protocol that enables decentralized token exchange using limit orders. It brings together the advantages of trading efficiency and ease-of-use of centralized exchanges with the transparency and self-custody of decentralized approaches, without the risk of impermanent loss, contrary to automatic market maker-based approaches.

1.1. Token swap as financial primitive. Swapping tokens is one of the most important and widely used primitives in decentralized finance. In March 2023, the average daily trading volume in cryptocurrencies was over \$34 Billion USD, after reaching a peak of \$516 Billion USD per day in May 20, 2021².

Trading volume on decentralized exchanges has been increasing over time and now is approaching 20% of the total traded cryptocurrency volume³. As of March 2023, Uniswap has transacted more than \$1.4 Trillion USD and 139 million trades⁴. The move from centralized to decentralized exchanges has been accelerating due to scandals involving CEXs (e.g., FTX bankruptcy), regulatory crackdown, and thanks to end-users' increased embrace of the principles of self-custody and decentralization. We believe that this trend towards larger trading volumes and will only accelerate in the future.

1.2. Limit orders as an interface to token swap. A limit order expresses a commitment to buy or to sell an asset, up to a certain quantity, at a price that is as good or better than the limit price. As supply and demand are represented by quantity available at a given price, limit orders provide a way for buyers and sellers to participate in price discovery and exchange.

We adopt the perspective that limit orders may be thought of as an interface to market participation. That is, limit orders provide participants with a means to engage with markets. We further promote this interface as one that is natural, flexible, and elegant. It is natural in that it expresses supply or demand directly in terms of quantity and price, with a clearly defined maximum possible exposure in a commitment to trade. The interface is flexible because it allows participants to construct personalized supply and demand curves by combining multiple limit orders. The elegance follows from the simplicity and ease-of-use of the interface.

As consequence of these benefits, limit orders and their aggregation into limit order books (LOBs) are ubiquitous in traditional financial markets and dominate cryptocurrency trading on centralized exchanges. Interestingly, however, limit orders currently do not play as prominent a role in the realm of decentralized finance. Instead, alternative approaches that rely upon automated market makers (AMMs) have taken center stage, and these approaches offer a different set of advantages and disadvantages.

¹Temporary code name

²<https://www.statista.com/statistics/1272903/cryptocurrency-trade-volume>

³<https://www.theblock.co/data/decentralized-finance/dex-non-custodial/dex-to-cex-spot-trade-volume>

⁴<https://uniswap.org/>

1.3. How exchanges create value. To better understand the comparative advantages and disadvantages of different approaches to token exchange, we step back and consider the purpose and functions of exchanges. Exchanges create value by bringing together different types of participants, such as investors, traders, hedgers, brokers, arbitrageurs, and market makers. Market participants have different goals (e.g., hedging, investing, speculating), horizons (from long-term investment to low-latency trading), risk tolerance, liquidity preferences, and beliefs about security values. The opportunity to trade arises from these differences. By providing a common meeting ground, exchanges facilitate matching supply and demand.

Exchanges have several roles:

- Establish the rules of the trading process and institutional roles, so that the trading process is structured, monitored, and standardized
- Provide a certain amount of oversight by monitoring and certifying financial statements and governance procedures
- Generate market data for market participants, such as trades and quote changes

Exchanges are compensated for this value creation by collecting transaction fees as a small percent of traded volume.

Typically, an exchange requires two components:

- (1) Trade matching: participants find a counterparty agreeing on quantity and price of the assets to swap (this includes price discovery)
- (2) Trade settlement (clearing): the assets are actually swapped after finding matching counterparties

In decentralized finance, a token swap can be accomplished in different ways, e.g., using

- a centralized exchange (CEX) (e.g., Coinbase, Binance, OkX)
- a decentralized exchange (DEX) (e.g., Uniswap, SushiSwap)

We consider CEXs a temporary bridge between traditional finance and the new vision of decentralized finance and we do not consider them a viable long-term solution to the problems that DeFi is poised to address.

Centralized exchanges are usually organized around different versions of a central limit order book (CLOB), whereas decentralized exchanges have been organized as

- off-chain order books (e.g., 1inch)
- on-chain order books (e.g., SwapSwap, KyberSwap)
- automated market makers (AMM) (e.g., Uniswap)

Off-chain solutions can run into custodial and censorship issues that stand in direct conflict with the ethos of decentralization and self-determination. Unfortunately, limitations of current blockchain technology restrict the amount of computation that can be performed on-chain, although these limits are continuously being removed by advancements in research and blockchain technology (e.g., layer 2 chains, optimistic and zero-knowledge rollups). Thus we consider solutions based on self-custody that do not require trusting a third party (or at least allow one to verify its fairness) in line with the principles of decentralization.

Off-chain order books are an hybrid version of CEX and DEX where the price discovery and trade matching happens off-chain, while the trade settlement is performed on-chain.

On-chain order book matching has the advantages of being custodial, since users are completely in charge of their funds. A major disadvantage of using on-chain limit order books in decentralized exchange is cost of computation. Continual submission and cancelation of orders incurs costs, as does the ongoing process of matching supply and demand.

1.4. The advantages of DELOC. DELOC retains the key advantages of on-chain decentralized exchange while overcoming the issues of cost. Additionally, it includes an implementation that

preserves the familiar LOB interface but deepens the pool of available liquidity in matching supply and demand.

DELOC addresses cost issues primarily in two ways: (1) discretizing time; and (2) scaling with layer 2 solutions. By addressing these issues, we combine the novel advantages of decentralized exchange with the utility and ease-of-use of the familiar interface of limit orders.

Behind the interface, we propose two smart contracts⁵ for ERC-20⁶ token exchange that match supply and demand differently depending upon whether there exists an external reference price (DaoCross) or whether the contract also performs the role of price discovery (DaoSwap).

DaoCross is a decentralized liquidity pool that crosses buy and sell orders with respect to an external reference price, i.e., a price oracle. This allows traders to interact directly with each other and share price improvement with respect to exchanges, e.g., by trading at bid-ask midpoint. Additionally, order intentions are not publicly disclosed prior to a cross, which enables block or other high-volume traders to execute quickly with minimal market impact.

DaoSwap performs price discovery by matching supply and demand at regular time intervals. In the simplest case, the clearing price is determined by a Walrasian-style auction ([16]). In cases where liquidity is pooled more broadly, the auction clearing mechanism satisfies a collection of conditional inequalities.

While DaoSwap and DaoCross differ in how the token exchange rate is determined, both share several common advantages:

- (1) Low cost: they implement a DeFi primitive that allows trading tokens peer-to-peer without incurring spread costs
- (2) Capital efficiency: participants provide liquidity only when they wish to trade, and only in the amount they wish to trade
- (3) No impermanent loss: liquidity providers are not exposed to the risk of impermanent loss (a form of unrealized loss that becomes realized upon withdrawing liquidity), unlike the case with other decentralized exchange protocols based on automated market makers
- (4) No intermediary custody: exchanged tokens always remain under the control of their respective owners
- (5) No rent extraction from high-frequency traders: speed alone does not confer an advantage to traders, as the discrete timing prevents predatory tactics such as front-running, limit order scalping, and spoofing, which are known issues seen with continuous limit order books
- (6) Ease-of-use: the interaction between buyers and sellers occurs through smart contracts, with a website front-end available

2. MATCHING LIQUIDITY

Suppose there are two parties, Alice an ETH holder and Bob a **wBTC** holder (wrapped Bitcoin, an ERC-20 compliant coin that tracks Bitcoin), who wish to swap tokens at a competitive rate. Suppose also that there are many additional ETH and **wBTC** holders who may be interested in participating in a swap. How should the liquidity be pooled? At first glance, it appears that there should be a single pool of liquidity. The wrinkle manifests in determining how to express a desire to trade. A simple expression of a desire to trade is a limit order representing a commitment to trade up to q in quantity of a token at a token exchange rate up to p . But what if some parties express quantity in terms ETH and some in terms of **wBTC**, and some parties express limit prices in terms of ETH per **wBTC**, while others express limit prices in terms of **wBTC** per ETH? Under these conditions, setting a single clearing price and determining fill prioritization rules raises several questions. We

⁵<https://ethereum.org/en/developers/docs/smart-contracts/>

⁶<https://ethereum.org/en/developers/docs/standards/tokens/erc-20/>

will consider and address these in the sequel, but to begin, we will discuss a more constrained setting.

2.1. Breaking the symmetry. To simplify, we follow the practice of FX (foreign exchange) futures markets, which break the symmetry by specifying two non-interchangeable roles:

- (1) a *base* currency (for quantity)
- (2) a *quote* currency (for price)

The roles are expressed by writing *base currency/quote currency*, e.g., “EUR/USD”.

A limit order then consists of a quantity expressed in the units of the base currency and a price expressed in terms of the quote currency (per unit or suitable multiple of base currency). See, for example, [7] (e.g., footnotes on page 25) for usage of this terminology. See [8] for how this works in practice for Euro FX contracts, where contract units are denominated in Euros and price is quoted in U.S. dollars and cents per Euro increment.

By breaking the symmetry, one may run a standard limit order book, hold a classical Walrasian-style auction, and handle size quantization effects by using one of a variety of priority rules based on quantity, price, and time. See [4] for discussion around priority rule variations and their effects on market quality outcomes.

An effect of breaking the symmetry in this way is that either liquidity for a pair of currencies may be expressed in only one of the two forms, or liquidity for a pair of currencies is split across two separate contracts (with roles of base and quote token reversed), each with its own limit order book. Using our example, this would mean treating wBTC/ETH and ETH/wBTC as separate token pairs, even though the union of the underlying parties and sources of liquidity are typically the same.

2.2. Basket case. It seems plausible that by retaining the symmetry in the two-token case and instead treating a pair of tokens as a single source of liquidity, one may contrive a more efficient exchange mechanism.

This minor expansion, however, suggests widening even further the universe of eligible swaps. For example, if there are three tokens available for exchange, one could contemplate many-for-one, one-for-many, or many-for-many token swaps, all to be carried out in an atomic fashion, and perhaps with complex constraints. A toy case along these lines one may consider is as follows:

Problem 2.2.1 (Triangular liquidity). Suppose there are the following three parties:

- Party A, who holds wBTC and wants ETH
- Party B, who holds ETH and wants DAI
- Party C, who holds DAI and wants wBTC

How should an auction be structured to “best” facilitate exchange?

While there may be other use cases and even demand for such auctions, unfortunately, the problem in general is NP-complete (e.g., [17]), which is a significant limiting factor in terms of feasibility and auction expense.

In the sequel we propose such an exchange mechanism valid for any collection of standard limit orders (all base/quote token limit orders may be pooled and cleared in a single optimization).

3. LIMIT ORDERS

In the previous section, we discussed the notions of base currency and quote currency from foreign exchange. Here we extend them to tokens.

3.1. Base/quote tokens. A token (ordered) pair consists of a *base token* and a *quote token*. The shorthand notation for expressing the pair is *base token/quote token* (e.g., “wBTC/ETH”), and as such indicates the respective roles of the tokens.

By convention, *quantity* to exchange is expressed in terms of the base token, and *price* is a token exchange rate expressed in terms of the quote token per unit of base token (or multiple thereof).

Example 3.1.1 (base/quote tokens). In the pair wBTC/ETH, wBTC is the base token and ETH is the quote token. Interest to trade is expressed in terms of orders with wBTC as base token and ETH as quote token, as follows:

- A “buy” order represents a commitment to purchase the base token wBTC and pay with the quote token ETH
- A “sell” order represents a commitment to sell the base token wBTC and receive the quote token ETH
- Quantity q is in terms of the base token (“buy/sell a certain amount of wBTC”)
- Limit price is in terms of the quote token (“buy/sell wBTC with a limit price of p ETH per wBTC”)

A party who places a “buy” order must have the quote token (ETH) in custody, i.e., in its wallet. A party who places a “sell” order must have the base token (wBTC) in custody.

3.2. Exchange price properties. If prices of base and quote tokens are expressed in terms of a common currency (e.g., USD) then it holds that:

$$p_{quote_per_base} = \frac{p_{quote}}{p_{base}},$$

where the numeraire plays the role of a quote token for both p_{quote} and p_{base} and is implicit.

Note that

$$p_{quote_per_base} = \frac{1}{p_{base_per_quote}}.$$

3.3. Order attributes and notation. We introduce the following tuple notation for a general limit order (valid for both DaoCross and DaoSwap).

Definition 3.3.1 (Limit order). A DaoCross or DaoSwap limit order is represented by a tuple of the form

```
(timestamp,
 action,
 quantity,
 base_token,
 limit_price,
 quote_token,
 deposit_address)
```

The quantities are arranged to make the order simple to read in natural language: “At timestamp `timestamp` create an order to `action` up to a number `quantity` `base_token` tokens for a limit price of `limit_price` with respect to the token `quote_token` and deposit the resulting tokens at `deposit_address`.”

We have previously discussed the roles of `action`, `quantity`, `base_token`, `limit_price`, and `quote_token`. The roles of `timestamp` include determining eligibility in a swap and can extend to

influencing order fill priority. The `deposit_address` attribute mirrors standard functionality available in traditional finance, e.g., in purchasing T-Bills on TreasuryDirect⁷. This feature facilitates account management through the use of multiple wallets. For example, a miner may have a supply of `wBTC` collected and held in one wallet which it would like to systematically diversify into `ETH` and perhaps other tokens. In specifying a deposit address, it may use a separate wallet to collect incoming `ETH`.

Example 3.3.1 (Limit order notation). The order

(1678660406, buy, 3.2, `ETH`, 4.0, `wBTC`, 0xdeadcd0de)

corresponds to the natural language description: “At timestamp Mon Mar 13 2023 02:33:25 GMT+0000, the user commits to buy up to 3.2 units of `ETH` in exchange for `wBTC` up to a `limit_price` of 4.0 `wBTC` per `ETH` with proceeds deposited at 0xdeadcd0de”.

3.3.1. *Extracting attributes from an order.* Given an order $o_i = (\text{buy}, q, A, p, B)$ we indicate with:

- $action(o_i)$ the desired action (`buy` or `sell`)
- $q_{base}(o_i)$ the maximum desired quantity q
- $base(o_i)$ the base token A
- $p(o_i)$ the limit price p (in terms of quote per base)
- $quote(o_i)$ the quote token B

3.3.2. *Order short notation.* In the sequel, when clear from the context we may:

- omit `timestamp` and `deposit_address` when not relevant to the discussion
- omit the (infinite) `limit_price` for market orders

3.4. **Order clearing quantity and exchange rate.** In DaoCross, orders are collected from users during a finite period of time, after which tokens are redistributed among users according to their limit orders.

Equilibrium prices for the tokens are determined based on the available limit orders and a criteria designed to maximize the welfare of the participants in the swap. At the same time, orders are matched in (generically) many-to-many relationships.

Swaps between base/quote token pairs occur at a single exchange rate, i.e., the clearing (or equilibrium) price is the same for all executed orders. On the other hand, quantity exchanged is specific to each order and each order’s constraint on quantity exchanged cannot be violated. In the same way, each order can be executed with a non-null quantity only when the limit price is compatible with the clearing exchange rate.

In the sequel we use an asterisk to denote clearing quantity and exchange rate:

- $q_{base}^*(o_i)$ denotes quantity of the base token exchanged in a swap from order o_i
- $p_{quote_per_base}^*$ denotes exchange rate between the quote and base tokens for all the swaps

3.5. **Execution of orders.** Next we introduce some examples of market order and limit order behavior when a clearing exchange rate has been set. In particular, consider a swap for the tokens `ETH`, `wBTC` and assume that the exchange rate between `ETH` and `wBTC` is fixed at 0.2 (i.e., 0.2 `wBTC` can be exchanged for 1 `ETH` and vice versa).

Example 3.5.1 (Market order notation and clearing). The following market orders omit `timestamp`, `limit_price`, and `deposit_address` in favor of emphasizing the amounts of token exchanged:

- An order (`buy`, 1.0, `ETH`, `wBTC`) means buying 1 `ETH` in exchange for 0.2 `wBTC`
- An order (`sell`, 1.0, `ETH`, `wBTC`) means selling 1 `ETH`, receiving the corresponding amount of 0.2 `wBTC`

⁷<https://www.treasurydirect.gov>

- An order (buy, 1.0, wBTC, ETH) means buying 1 wBTC, paying with 5 ETH
- An order (sell, 1.0, wBTC, ETH) means selling 1 wBTC in return for 5 ETH

Next we consider the behavior of limit orders under the same prevailing exchange rate and we assume that there is sufficient supply of tokens to fully fill the orders.

Example 3.5.2 (Executable buy limit order). A limit order

$$(\text{buy}, 1.0, \text{ETH}, 0.5, \text{wBTC})$$

means “buy up to 1 ETH in exchange for wBTC at a rate up to 0.5 wBTC per ETH.” In this case, since the price of one ETH is equal to 0.2 wBTC, the order can be executed at the prevailing market rate.

Example 3.5.3 (Non-executable buy limit order). On the other hand, a limit order

$$(\text{buy}, 1.0, \text{ETH}, 0.1, \text{wBTC})$$

requires that the rate of wBTC per ETH be lower than the current market rate, and so the limit price prevents a token swap from being carried out.

Example 3.5.4 (Non-executable sell limit order). A limit order

$$(\text{sell}, 1.0, \text{ETH}, 0.5, \text{wBTC})$$

means “sell up to 1 unit of ETH in exchange for wBTC at a rate down to 0.5 wBTC per ETH.” Since the current rate of wBTC per ETH is 0.2, which is below the limit price of 0.5, the order cannot be executed.

3.6. Limit order as a set of inequalities. Any limit order as defined above can be translated into inequalities involving quantity of exchanged tokens and token exchange rates. Multiple limit orders can be converted into a system of inequalities, which collectively constrain potential exchange outcomes.

3.6.1. Inequalities for buy order. An order of the form $o_i = (\text{buy}, q, A, p, B)$ means “buy q units of token A in exchange for token B with a limit price up to p B per unit of A”, and it corresponds to the following constraint on realized quantity exchanged and clearing exchange rate:

$$(p_{B\text{-per-}A}^* \leq p(o_i)) \wedge (0 \leq q_A^*(o_i) \leq q(o_i)) \vee (q_A^*(o_i) = 0)$$

The constraint on the quantity exchanged is conditioned on the corresponding price’s satisfying the desired limit price constraint: if the desired limit price constraint is not met, the swap cannot be carried out and the exchanged quantity is 0.

Example 3.6.1 (Inequalities for buy order). An order of the form $o_1 = (\text{buy}, 2, A, 3, B)$ means “buy 2 units of token A in exchange for token B with a limit price up to 3 B per unit of A”, and it corresponds to the following constraint on realized quantity exchanged and clearing exchange rate:

$$(p_{B\text{-per-}A}^* \leq 3) \wedge (0 \leq q_A^*(o_1) \leq 2) \vee (q_A^*(o_1) = 0)$$

Example 3.6.2 (Inequality for buy market order). A market order (i.e., without a limit price) has no constraint on exchange rate and thus is reduced to

$$0 \leq q_A^*(o_1) \leq q(o_i)$$

3.6.2. *Inequalities for sell order.* An order of the form $o_i = (\text{sell}, q, A, p, B)$ means “sell q units of token A in exchange for token B with a limit price of at least p B per unit of A ”, and it corresponds to the following constraint on realized quantity exchanged and clearing exchange rate:

$$(p_{B_per_A}^* \geq p(o_i) \wedge (0 \leq q_A^*(o_i) \leq q(o_i)) \vee (q_A^*(o_i) = 0))$$

Example 3.6.3 (Inequalities for sell order). In the same way, an order like $o_2 = (\text{sell}, 3, A, 2, B)$ means “sell 3 units of token A in exchange for token B with a limit price of at least 2 B per unit of A ”, which corresponds to

$$(p_{B_per_A}^*(o_2) \geq 2) \wedge (0 \leq q_A^*(o_2) \leq 2) \vee (q_A^*(o_2) = 0)$$

3.7. Order normalization. When the exchange rate between two tokens is known, it is possible to convert buy/sell orders for both A and B tokens into buy/sell orders where all orders have token B as the base token, according to the transformation below.

Let

$$o_1 = (\text{buy}, q, A, p_{B_per_A}, B).$$

Suppose that $p_{B_per_A}^*$ is fixed and known. Then the order

$$o_2 = (\text{sell}, q', B, 1/p_{B_per_A}, A)$$

may be handled in order crossing in the same way that o_1 may be provided that

$$q' = q \cdot p_{B_per_A}^*$$

4. REFERENCE CLEARING PRICES AND PRIORITIZATION

DaoCross relies on a *price oracle* for determining the effective token exchange rate in a cross. The price oracle may come from a lit exchange, centralized or decentralized, or even on-chain automated market makers such as Uniswap ([2, §2.2]).

The case of using an automated market maker as a price oracle is relatively straightforward, provided there is an automated market maker trading the target currency pair with sufficient liquidity.

To use an exchange as a price reference, we must consider some additional steps. First, exchanges typically use a dollar stablecoin (e.g., USDC, USDT) as the quote currency and all other currencies as base currencies. Our example of $wBTC/ETH$ would be considered a cross pair in the world of traditional finance. Because most cross pairs are not traded directly on exchanges, we must derive a clearing price from pairs involving stablecoins.

4.1. Lit exchange bid-ask midpoint reference price. Let bid_{BTC} , ask_{BTC} be stablecoin-denominated top-of-book bid-ask prices for $wBTC$ (e.g., expressed in USDC), and bid_{ETH} , ask_{ETH} be the analogous prices for ETH . Then, a commitment to buy one $wBTC$ and pay ETH would clear at a midpoint price of

$$wBTC/ETH_{\text{midpoint}} = \frac{\text{bid}_{BTC} + \text{ask}_{BTC}}{\text{bid}_{ETH} + \text{ask}_{ETH}}$$

expressed in $wBTC$ per ETH . Note that if the dollar price of BTH is significantly more than the dollar price of ETH , then this price implies paying many multiples of ETH for BTC (as is the case at the time of this writing).

In general, DaoCross may use the following reference price for a base/quote token pair, where the bids and asks come from a lit exchange and are expressed in terms of stablecoin prices:

$$\frac{\text{bid}_{\text{quote token}} + \text{ask}_{\text{quote token}}}{\text{bid}_{\text{base token}} + \text{ask}_{\text{base token}}} \quad (1)$$

When using either an on-chain price oracle or an exchange as a price oracle, it is possible to perform a time or volume weighted average of prices so as to avoid extreme price variations and to mitigate the risk and effects of any market manipulation attempts.

4.2. Order crossing. At regular time intervals, order submissions are cut off and reference prices are determined. An element of randomness is used to determine order cutoff times and reference price cutoff times in order to mitigate manipulative behaviors.

An order is eligible for matching if its timestamp is within the cutoff window and if the external reference price does not exceed its limit price.

Except on occasions where eligible buy/sell volume is perfectly matched, not all orders can be fully crossed. There are also discretization effects to consider arising from discrete order sizes and a discrete price grid [5][§3.2.1]. See [4] for a discussion of the trade-offs surrounding different prioritization rule choices. See [15] for the prioritization rules used by one of Morgan Stanley’s equity liquidity pools.

4.2.1. Priority rules. DaoCross prioritizes fills according to:

- Volume (higher volume comes first in priority)
- Price (higher limit price breaks volume ties)
- Timestamp (earlier timestamp breaks ties in volume and price)

If, in the unlikely scenario that all three of these parameters perfectly agree, a certain priority is not guaranteed.

4.2.2. Matching algorithm. The mechanism for matching eligible buy and sell orders consists of two priority queues, one for eligible buy orders and one for eligible sell orders. Priority is determined according to the volume/price/timestamp priority rules introduced above. Top-of-queue orders are compared, and the lesser of the two volumes is fully filled. Once an order is fully filled at the established reference price, it is removed from the priority queue. The procedure continues until one of the two priority queues is empty.

4.2.3. Computational complexity. Let n denote the sum (or max) of the number of eligible buy and sell orders. Priority queue construction occurs in $O(n)$ time, each order removal costs $O(\log n)$, and order removal occurs $O(n)$ times. So, the computational time complexity of the task is $O(n \log n)$. Memory requirements are $O(n)$.

5. FORMULATION OF THE GENERAL DAO-SWAP PROBLEM

The general problem of determining the token equilibrium prices and the allocation among the swap participants can be formulated in terms of the inequalities on quantities and prices corresponding to the orders and on the need that the total quantity exchanged of each token be preserved across the swaps, i.e., the quantity bought for each token is equal to the quantity sold.

5.1. Limit order inequalities. Let $\{o_i\}_{i=1}^n$ be a set of limit orders, each of the form

$$o_i = (a_i, q_i, \pi_i, p_i, \tau_i)$$

where the quote token τ_i is implicit in the quantity q_i and the quote token τ_i and base token π_i are implicit in the limit price p_i and in the action a_i .

For each order o_i , let x_{i1} denote the realized amount of base token exchanged, and let x_{i2} denote the realized amount of quote token exchanged.

Remark 5.1.1. Regardless of the direction of the order action a_i , we require the quantities of token exchanged to be nonnegative:

$$x_{i1} \geq 0, \quad x_{i2} \geq 0.$$

While not a limit-order imposed constraint per se, it helps clarify conventions, which is that a positive quantity of token must be exchanged for a positive quantity of token.

Remark 5.1.2. With this notation, the limit order constraint on quantity may be expressed as

$$x_{i1} \leq q_i$$

Additionally, for each o_i , let a_{i1} and a_{i2} belong to $\{-1, +1\}$, with sign determined by o_i in the following way: if a_i (the action) of o_i is to buy, then set $a_{i1} = -1$ (the buy order is taking liquidity of the base token), and if it is to sell, then set $a_{i1} = 1$. We always have $a_{i2} = -a_{i1}$, and if the action of a_i is to sell rather than to buy, then its signs are reversed with respect to the buy action.

The effective transaction price (in the event that positive tokens are exchanged) is represented by x_{i2}/x_{i1} . If the action is to buy (base token), then p_i represents an upper bound, and the opposite is true if the action is to sell.

Remark 5.1.3. The limit order constraint on price may be expressed as

$$a_{i2}x_{i2} \leq x_{i1}p_i$$

This holds true over the whole range of valid values for the x_{ij} (that is, including equality with zero).

5.2. Supply and demand match. Let T denote the intersection of the union of all base tokens and the union of all quote tokens:

$$T := \left(\bigcup_{i=1}^n \pi_i \right) \cap \left(\bigcup_{i=1}^n \tau_i \right)$$

The set of tokens T is the set of tokens eligible for swapping.

For each token $u \in T$, define the indicator function $\mathcal{T} : T \rightarrow \{0, 1\}$ via

$$\mathcal{T}_u(v) = \begin{cases} 1 & \text{if } v = u \\ 0 & \text{if } v \neq u \end{cases}$$

Then, for each $u \in T$, the following conservation law must hold:

$$\sum_i \mathcal{T}_u(\pi_i) \cdot a_{i1}x_{i1} + \mathcal{T}_u(\tau_i) \cdot a_{i2}x_{i2} = 0$$

Note that this translates into one equality per token participating in the swap. These equalities express the notion that each filled order must have, across the collection of orders, suitable counterparties.

5.3. Combining the constraints. We now collect the constraints that must be satisfied:

$$\begin{cases} x_{i1} \geq 0, & x_{i2} \geq 0 & \forall i \in \{1, \dots, n\} \\ x_{i1} \leq q_i & & \forall i \in \{1, \dots, n\} \\ a_{i2}x_{i2} \leq x_{i1}p_i & & \forall i \in \{1, \dots, n\} \\ \sum_{i=1}^n \mathcal{T}_u(\pi_i) \cdot a_{i1}x_{i1} + \mathcal{T}_u(\tau_i) \cdot a_{i2}x_{i2} = 0 & \forall u \in T \end{cases} \quad (2)$$

5.4. Solution of the general DaoSwap problem. Given the constraints, our goal is to find quantities x_{i1}, x_{i2} such that exchanged quantity is maximized.

Note that this problem falls in domain of linear programming. We may arrange the nonnegative quantities x_{i1}, x_{i2} , $i \in \{1, \dots, n\}$ in a vector x . Our goal is to find x that maximizes

$$c^T x$$

where $c^T = [1 \ \dots \ 1]$ is of size $2n$ subject to

$$Ax \leq b$$

(TODO(Paul): reformulate the constraints using A and b) and

$$x \geq 0.$$

The problem is polynomial-time in the ideal setting of infinite divisibility. Discretization (and renormalization) move the problem to the domain of integer programming, which is NP-hard.

In our setting, where the typical exchanged value is much larger than the discretization grid, obtaining a linear programming solution and discretizing after-the-fact may be acceptable in practice.

5.5. Additional constraint 1: Unique clearing price per token pair. In the DaoCross case, a single exchange rate per exchanged token pair prevails. One may make a case that a desirable property for a swap is that, for participants trading the same token pair, no participant gets a better or worse price than any other.

Suppose the effective exchange rate for an order involving tokens $\nu \in T$ and $\eta \in T$ is the same for all filled orders. In other words, the clearing exchange rate is unique for all participants. Then, for all i, j whose respective orders o_i, o_j both involve tokens ν, η , we have

$$x_{i1}x_{j2} - x_{i2}x_{j1} = 0$$

when the tokens ν and η play identical base/quote roles in o_i, o_j , and

$$x_{i1}x_{j1} - x_{i2}x_{j2} = 0$$

if they play opposite roles.

We rewrite this as

$$\begin{aligned} x_{i1}x_{j2} - x_{i2}x_{j1} &= 0 & \forall i, j \in \{1, \dots, n\} : \pi_i = \pi_j \wedge \tau_i = \tau_j \\ x_{i1}x_{j1} - x_{i2}x_{j2} &= 0 & \forall i, j \in \{1, \dots, n\} : \pi_i = \tau_j \wedge \tau_i = \pi_j \end{aligned}$$

The unique clearing price constraint prevents the problem from falling in the domain of linear programming.

Note that limit order constraints as formulated in the DaoCross case also hint at this nonlinearity, as they involve conjunctions with a global clearing price (which DaoSwap must determine).

TODO(gp): explain better, cite... The general solution of the DaoSwap problem must satisfy a set of non-linear, combinatorial constraints, whose solution is NP-complete.

It can still be solved in an effective way with solvers like...

5.6. Additional constraint 2: No arbitrage. Beyond the constraint of having a unique clearing price (if the clearing price is defined), it is desirable to structure trade so that arbitrage opportunities in the realized exchange were not present.

Let Γ denote the directed graph whose vertices V are given by the set of elements T , and whose directed edges E are given by

$$E = \bigcup_{\substack{i \in \{1, \dots, n\} \\ \pi_i, \tau_i \in T}} ((\pi_i, \tau_i) \cup (\tau_i, \pi_i))$$

Each edge $(b, a) \in E$ seek to give the weight $w_{a,b}$. Note that $w_{a,b}$ and $w_{b,a}$ are distinct.

Let C denote the collection of all simple cycles of Γ . For an simple cycle $\gamma \in C$, consecutively enumerate the vertices (with starting place arbitrary) by c_j for $j = 1, \dots, |\gamma|$. Let $c_0 := c_{|\gamma|}$ for

notational convenience. Then the *no arbitrage* constraints states that, for each $\gamma \in C$, we have

$$\sum_{j=0}^{|\gamma|} w_{j,j+1} = 0$$

In the case of a two-token cycle, this constraint reduces to ensuring that there is a unique exchange rate between the pair of tokens (changing the roles of the base and quote tokens in limit orders inverts the exchange rate). For an efficient algorithm for finding all elementary cycles, see [10] as well as [11] for a comparison.

5.7. DaoCross problem as simplified DaoSwap problem. In the DaoCross set-up, the price of the tokens involved in the swap is determined by an external oracle. This allows simplifying the general problem by applying the equivalence principle between orders and removing the non-linearity from the set of inequalities above.

In the case of each order o_i , the boolean condition c_i (which compares the actual price p^* and the limit order price $p(o_i)$) is either verified or not (the solution does not need to consider both branches):

$$\begin{cases} 0 \leq q_{base}^*(o_i) \leq q_{base}^* & \text{if } c_i \\ q_{base}^*(o_i) = 0 & \text{if } \neg c_i \end{cases} . \quad (3)$$

TODO: Comment on quote token amount exchanged. Verify supply meets demand.

The problem then becomes a set of linear inequalities (in fact, if $\neg c_i$ prevails, then order o_i is effectively removed from problem) that can be solved with various efficient methods (e.g., simplex-method).

TODO(gp): check

6. FEES AND TOKENOMICS

DELOC charges a fee on each transaction based on the exchanged tokens. E.g., a transaction requiring

7. IMPLEMENTATION DETAILS

7.1. DaoCross and DaoSwap Architecture.

7.2. Off-chain computation. Currently DELOC offloads some computations to external oracles, due to current computational limitation of blockchains. We do not believe that this is detrimental to the security and decentralization level of DELOC as long as these computations are provably correct.

For instance, although the solution of the DaoSwap problem is NP-hard, verifying that one solution is correct only requires time linear in the number of constraints.

For this reason, DaoSwap stores on-chain the result of the DaoSwap optimization in order to allow independent verification that the off-chain system is not malicious or compromised.

7.3. Ensuring a timely solution. DaoSwap avoids the case where solving the optimization problem becomes intractable by performing a swap when a maximum number of orders and/or currency pairs is reached.

7.4. Performing the swap.

8. ALTERNATIVE SOLUTIONS FOR TOKEN SWAPPING

TODO(gp): Add colors: in green what's good, yellow, red

TODO(gp): Reorg, maybe from best to worst? DaoSwap, LOB, DaoCross, AMM

	<i>LOB</i>	<i>AMM</i>	<i>DaoSwap</i>	<i>DaoCross</i>
Custodial risk	Yes	No	No	No
Transparency	Low	High	High	Medium
Censorship resistance	No	Yes	Yes	No
Intrinsic efficiency	Medium	Low	High	Medium
Support for limit orders	Yes	No	Yes	Yes
Risk of predatory behaviors	High	High	Low	Medium
Impermanent loss	No	Yes	No	No
Multi-coin exchange support	Low	Medium	High	High

8.1. DaoSwap.

- Custodial risk: Low
 - Private keys and funds are always under the control of the users
- Transparency: High
 - Work is done on-chain and the off-chain computations are independently verifiable
- Censorship resistance: Yes
 - The application runs on-chain and even if the web front-end is attacked or disabled. The off-chain computation can be made robust using similar approaches to distributed oracles, or even ported on-chain if gas prices are not an issue (e.g., using Layer2 solutions)
- Intrinsic efficiency: High
 - DaoSwap relies on periodic auctions and it has been argued in previous literature that for small enough intervals (e.g., seconds) periodic auctions result in the same quality of continuous matching without allowing predatory behaviors
 - TODO(gp): Add a reference to Cramton paper
- Support for limit order: Yes
 - Limit orders are supported natively
- Risk of predatory behaviors: Low
 - TODO(gp):
- Impermanent loss: No
 - TODO(gp):
- Support for multi-coin swap: Yes
 - TODO(gp): DaoSwap/Cross pools all the liquidity in a single optimization problem

8.2. DaoCross.

- Custodial risk: Low
 - Same as DaoSwap
- Transparency: Medium
 - DaoCross relies on price discovery carried out on a lit exchange
- Censorship resistance: No
 - The robustness of DaoCross is the same as the lit exchange that it relies on. See comments on the corresponding topic for LOB
- Intrinsic efficiency: Medium
 - The efficiency in matching trades is the same as the limit-order book. See comments on the corresponding topic for LOB
- Support for limit order: Yes
 - Same as DaoSwap
- Risk of predatory behaviors:
 - TODO(gp)

- Impermanent loss: No
 - TODO(gp)
- Support for multi-coin swap:
 - TODO(gp): DaoSwap/Cross pools all the liquidity in a single optimization problem

8.3. Limit Order Book (LOB).

- Custodial risk: High
 - Exchanges own user private keys and funds, creating issues with fraud (e.g., FTX) and hacking (Mt. Gox)
- Transparency: Low
 - Many centralized exchanges can easily exaggerate volume (e.g., through wash trading) to attract ICOs and liquidity
 - There is no guarantee on the
- Censorship resistance: No
 - Users need to link their bank account to CEX and transfer funds, which can take 2-3 days and be subjected to Know-Your-Customer (KYC) and AML (Anti money laundering policies)
 - Governments can censor, interfere with, or even sever the connection between traditional and decentralized finance (see recent interventions of SEC against Coinbase and Binance, and FDIC rescue of Silvergate Bank, Signature Bank)
- Intrinsic efficiency: Medium
 - Limit order books are considered efficient in matching continuously trades, although recently researchers have criticized their time-continuous-time nature as source of latency arbitrage (TODO(gp): Add ref to Compton, IEX, Flash Boys)
 - The same token can be traded on different exchanges causing liquidity and price discovery to be fragmented with detriment on market quality
- Support for limit order: Yes
 - Limit order books naturally support limit orders from users
 - Risk of predatory behaviors: High
 - LOB operators often welcome and incentivize high-frequency traders as way to increase trading revenues and liquidity, at expense of predatory tactics (e.g., front-running, latency arbitrage, spoofing, sniping) which arm
- Impermanent loss: No
 - LOB
- Support for multi-coin swap: No

8.4. Automatic Market Makers (AMM).

- Custodial risk: Low
 - Private keys and the
- Transparency:
- Censorship risk:
- Intrinsic efficiency: Low
 - Arbitrageurs are needed to keep liquidity pool and prices in alignment with other centers for price discovery
- Support for limit order: No
 - Uniswap V3 doesn't support limit orders directly but only in terms of
- Risk of predatory behaviors: High
 - TODO(gp):
- Impermanent loss: Yes

- Uniswap is affected by impermanent loss because liquidity providers and liquidity takers are in general different users operating at different time scales
- Support for multi-coin swap: Medium
 - Uniswap requires multiple swaps and fees for arbitrary tokens

9. COMMENTS ON AMMs

Some of the benefits of AMMs are:

- conceptual simplicity
- low computational requirements
- ability to provide liquidity even for illiquid markets
- ability to function without a reference price

Some of the drawbacks of AMMs are:

- force liquidity providers to trade at worse-than-market prices
- a rarely used building block used mainly in prediction markets rather than in mainstream finance. In fact only recently papers have started analyzing the financial return / risk profile of AMMs [13]
- price needs to be corrected by an arbitrageur, impacting the quality of the provided liquidity
- artificially separates liquidity providers from traders, preventing traders from providing liquidity to each other in the way well-functioning markets do

Some critiques and counterpoints can be made to the benefits of AMMs listed above.

- conceptual simplicity. Although this is a favorable point for users without experience in finance, the evolution of finance practices favors the use of limit orders, as explained in the introduction.
- low computational requirements. This is not necessarily a strict requirement any more due to progress in off-chain computation and improved scalability in blockchain technology (such as layer 2 blockchains). We do not believe that an inefficient solution (from the point of view of exchanged value) should be preferred only because of implementation simplicity.
- ability to function even for illiquid markets. Although this is a valid advantage for many markets (e.g., prediction markets), the vast majority of crypto coins are extremely liquid and do not require trading off trading quality.
- ability to function without a reference price. This feature was certainly appealing to initial researchers (e.g., [6]) in search of a fully-decentralized solution to the problem of token exchange. In reality, the fact that price discovery unquestionably happens on current CEXs turned AMMs into arbitrage generation machines (AGMs), where 80% of the trading volume is due to arbitrageurs keeping AMM prices in sync with the predominant price TODO(gp): Add reference

One of the problems with current state of DeFi is that users look for applications to use their crypto coins, waiting for mainstream adoption of crypto in every day payment. Also a current trend is for crypto holders to find yield to benefit from holding crypto. AMMs satisfied the need for ways to extract yields from holding coins.

We don't think AMM will completely replaced but new mechanisms are needed LPs need to be compensated with revenues to offset the cost of their adverse selection (quantified and characterized by the LVR paper)

10. CPMM AS AN LOB

TODO(Paul): Replace with a reference to the derivation in [18], and focus on the difference in stylized facts between the LOB a CPMM implies versus the LOBs seen in LOB-based markets. Note that some conventions here vs the paper are slightly different.

Here we reinterpret a constant product market maker as a limit order order book that a market participant may transact against (e.g., hit a bid or lift an offer). In a traditional limit order book, one expects frequent limit order cancellation and submission in response to market activity.

Let A and B be two tokens in a CPMM and suppose that the available supply of A and B in the pool at time t_0 are, respectively, a_0 and b_0 . We set $k = a_0 b_0$. Though neither token A nor token B enjoys a distinguished role, for the purposes of this discussion we designate token A the base token and token B the quote token (due to the symmetry, the subsequent analysis continues to hold with the roles reversed). From this perspective, we think of placing orders to either buy or sell token A, with price quoted in token B per unit of A.

10.1. Buy order. Suppose we wish to place an order to buy δa of token A. This corresponds to removing an amount of δa from the pool, which must be compensated for by adding (paying) some amount δb to the pool. The constant product constraint ensures

$$a_0 b_0 = (a_0 - \delta a)(b_0 + \delta b)$$

which upon rearrangement leads to

$$\delta b = b_0 \frac{\delta a}{a_0 - \delta a}$$

The effective cost of buying δa units of A is given by

$$\frac{\delta b}{\delta a} = \frac{b_0}{a_0 - \delta a}$$

If we make the substitution $b_0 = k/a_0$, this becomes

$$\frac{\delta b}{\delta a} = \frac{k}{a_0(a_0 - \delta a)}$$

In the idealized setting of infinite divisibility of tokens, the instantaneous price of buying A at point (a_0, b_0) is

$$\frac{b_0}{a_0} = \frac{k}{a_0^2}$$

For convenience, define p_0 as this price:

$$p_0 := \frac{b_0}{a_0} = \frac{k}{a_0^2}$$

Note that as more of token A is purchased (more A is withdrawn from the pool), the price of token A in the pool (in terms of the quote token) goes up, as expected.

10.2. Supply liquidity. TODO(Paul): Here we use p as effective price of transacting quantity q . Using cumulative quantity up to a given (max) price p (as in [18]) is more natural for some analyses.

Rephrasing, suppose q denotes quantity of token A that we wish to purchase. If the current state of the market is a_0, b_0 , then the price p of purchasing quantity q is

$$p(a_0, b_0, q) = \frac{k}{a_0(a_0 - q)}$$

For brevity, we drop the functional dependence. Reexpressing in terms of quantity, we have

$$q = a_0 - \frac{k}{a_0 p} = a_0 \left(1 - \frac{p_0}{p}\right)$$

which expresses cumulative quantity q available at average price p .

10.2.1. *Slippage and post-trade state.* Measure slippage as the relative difference between effective buy price and market price p_0 . We calculate

$$\frac{p - p_0}{p_0} = \frac{a_0}{a_0 - q} - 1 = \frac{q}{a_0 - q}$$

Note that this only depends upon amount of \mathbf{A} in the pool and the purchase quantity. TODO(Paul): find and cite the earliest references for this. Note that this is approximately linear in q for large a_0 (in other words, market impact is approximately linear when there is sufficient depth). Note that this is independent of volatility and turnover.

When a buy transaction takes places, the market price changes from

$$p_0 = \frac{b_0}{a_0}$$

to

$$p_1 = \frac{b_0 + \delta b}{a_0 - \delta a}.$$

Note that this is greater than the effective purchase price of

$$\frac{b_0}{a_0 - \delta a},$$

which is analogous to what happens when multiple levels of a limit order book are cleared (the new best offer is greater than the effective price of the buy order that cleared multiple levels of offers).

Following a transaction, marginal quantity readjusts given the new position (a_1, b_1) of the book.

REFERENCES

- [1] Hayden Adams, *Uniswap Whitepaper* (2018), available at <https://hackmd.io/s/HJ9jLsfTz>.
- [2] Hayden Adams, Noah Zinsmeister, and Dan Robinson, *Uniswap v2 Core* (March 2020), available at <https://uniswap.org/whitepaper.pdf>.
- [3] Hayden Adams, Noah Zinsmeister, Moody Salem, River Keefer, and Dan Robinson, *Uniswap v3 Core* (March 2021), available at <https://uniswap.org/whitepaper-v3.pdf>.
- [4] Alejandro Bernales, Daniel Ladley, Evangelos Litos, and Marcela Valenzuela, *Alternative Execution Priority Rules in Dark Pools* (July 22, 2022), available at https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4169352.
- [5] Jean-Philippe Bouchaud, Julius Bonart, Jonathan Donier, and Martin Gould, *Trades, Quotes and Prices: Financial Markets Under the Microscope*, Cambridge University Press, 2018.
- [6] Vitalik Buterin, *Let's run on-chain decentralized exchanges the way we run prediction markets* (2017), available at https://www.reddit.com/r/ethereum/comments/55m04x/lets_run_onchain_decentralized_exchanges_the_way.
- [7] CME Group, *2023 FX Product Guide*, available at <https://www.cmegroup.com/trading/fx/files/fx-product-guide-2023-us.pdf>.
- [8] ———, *Euro FX Futures - Contract Specs*, available at <https://www.cmegroup.com/markets/fx/g10/euro-fx.contractSpecs.html>.
- [9] Robin Hanson, *Combinatorial Information Market Design*, Information Systems Frontiers **5** (2003), 107-119, available at <https://doi.org/10.1023/A:1022058209073>.
- [10] Donald B. Johnson, *Finding all the Elementary Circuits of a Directed Graph*, SIAM Journal on Computing **4** (1975), no. 1, 77-84.
- [11] Prabhaker Mateti and Deo Narsingh, *On Algorithms for Enumerating All Circuits of a Graph*, SIAM Journal on Computing **5** (1976), no. 1, 90-99, available at <https://doi.org/10.1137/0205007>.

- [12] Jason Milionis, Ciamac C. Moallemi, and Tim Roughgarden, *Complexity-Approximation Trade-offs in Exchange Mechanisms: AMMs vs. LOBs* (2023), available at <https://arxiv.org/abs/2302.11652>.
- [13] Jason Milionis, Ciamac C. Moallemi, Tim Roughgarden, and Anthony Lee Zhang, *Automated Market Making and Loss-Versus-Rebalancing* (2022), available at <https://doi.org/10.48550/arXiv.2208.06046>.
- [14] Morgan Stanley, *Morgan Stanley Dark Pools*, available at <https://www.morganstanley.com/disclosures/morgan-stanley-dark-pools>.
- [15] ———, *MS Pool ATS-N Filings*, available at <https://www.sec.gov/cgi-bin/browse-edgar?action=getcompany&filenum=013-00117>.
- [16] *Walrasian auction*, available at https://en.wikipedia.org/wiki/Walrasian_auction.
- [17] Mu Xia, Jan Stallaert, and Andrew B. Whinston, *Solving the combinatorial double auction problem*, Journal of Operation Research **164** (2005), 239-251, available at <https://www.sciencedirect.com/science/article/abs/pii/S0377221703008981>.
- [18] Jamie E. Young, *On Equivalence of Automated Market Maker and Limit Order Book Systems* (October 2020), available at https://professorjey.com/assets/papers/AMM_Order_Book_Equivalence_DRAFT_2020_10_16.pdf.