From CoWs to Multi-Chain AMMs
A Strategic Optimization Model for Enhancing Solvers

Shi, Zeshun; Sweck, Sydney; Zaki, Omar

DOI
10.1109/Blockchain62396.2024.00022

Publication date
2024

Document Version
Final published version

Published in
Proceedings of the 2024 IEEE International Conference on Blockchain (Blockchain)

From CoWs to Multi-Chain AMMs: A Strategic Optimization Model for Enhancing Solvers

Zeshun Shi*, Sydney Sweck†, and Omar Zaki†
*Cyber Security Group, Delft University of Technology, the Netherlands
†Composable Foundation, Zug, Switzerland
Email: z.shi-2@tudelft.nl, sydney@composable.finance, 0xbrainjar@composable.finance

2024 IEEE International Conference on Blockchain (Blockchain) | 979-8-3503-5159-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/BLOCKCHAIN62396.2024.00022

***Abstract*—In the rapidly evolving decentralized finance (DeFi) ecosystem, ensuring efficient and interoperable transaction mechanisms is a critical challenge. To address this issue, this paper introduces a strategic optimization model for a blockchain-based token exchange platform. By leveraging the principles of Coincidence of Wants (CoWs) and multi-chain Automated Market Makers (AMMs), our model enhances interoperability and efficiency of token exchange in the DeFi space. Users specify their transaction intents and solvers compete to find the most efficient execution pathways, considering factors such as available liquidity and market constraints. This approach not only facilitates seamless cross-chain transaction flows, but also optimizes the efficiency of existing solvers and reduces the reliance on centralized mechanisms. To validate the effectiveness of the proposed model, we conducted extensive simulation experiments assessing the model's performance with various order inputs and AMM constraints. The results show that our optimization model significantly increases the transaction completion rate. This improvement ranges from 26.1% to 46.1% compared to the CoWs-only model under different experimental settings. This enhances user welfare and market fairness. The proposed optimization model has broad applicability to efficient and interoperable cross-chain token transactions. Thus, it has a significant potential impact on the DeFi landscape.**

***Index Terms*—Solvers, coincidence of wants, automated market makers, user welfare optimization, cross-chain liquidity**

I. Introduction

A blockchain is a ledger or database of digital transactions, shared among nodes of a computer network [1]. This computing solution has risen in popularity over the years since the first blockchain, Bitcoin, was created in 2008 [2]. For example, the banking and financial services sector of the blockchain market alone grew from $1.89 billion in 2022 to $3.07 billion in 2023 (a compound annual growth rate of 62.1%) [3]. Cryptocurrencies ("crypto") and other digital assets that operate in blockchain networks have been a significant factor in this growth, with the total crypto market capitalization doubling in 2023 [4]. Together, crypto and blockchain technology have enabled a robust market for decentralized finance (DeFi), an industry of crypto-based transactions, exchanges, and other financial services.

Within DeFi, there are a number of mechanisms that facilitate the settlement of crypto transactions. Automated market makers (AMMs) play a particularly important role. AMMs are institutions that stand ready to buy or sell assets automatically. They do so by allowing traders to place orders within the AMM using algorithmic pricing [5]. These entities are an improvement of traditional market makers, which are not automated and thus required more intensive means of establishing prices.

While AMMs are important in crypto transactions, it is possible to settle transactions without them. Notably, transactions can be settled via the Coincidence of Wants (CoWs) principle [6]: an economic phenomenon where two or more parties coincidentally hold an item or asset that the other wants. Thus, these parties can exchange directly with one another without an intermediary exchange like an AMM. In the case of crypto, a CoW occurs when transactions are coincidentally the opposite of one another (i.e. a transaction swapping asset A for asset B and another to swap asset B for asset A form a CoW) [7]. By removing the need for a market maker, settling a crypto transaction via a CoW requires less information to be processed on-chain. This is often more time- and cost-efficient, and thus considered preferable to AMM settlement.

Despite its many advancements such as AMMs, DeFi is still an evolving industry [8]. One persisting limitation is the lack of interoperability between blockchains. Specifically, interacting with DeFi processes across multiple blockchains is quite complex; users must navigate between many apps and wallets, perform multiple transactions, and identify opportunities quickly before they disappear. Moreover, there are many barriers between different blockchains, meaning users and subsequently liquidity cannot seamlessly transact between ecosystems. Existing crypto transaction settlement tools do not address these issues. For example, there is a lack of cross-chain AMM solutions.

As a result, in the blockchain and DeFi space, popular areas of research and development include cross-chain interoperability [9], [10] and advanced intents/solvers [11], [12]. However, these ideas have yet to be combined; there is not a cross-chain intent settlement framework available on the market. Such a framework would enable users to simultaneously take advantage of the benefits of both intents/solvers and cross-chain interoperability.

To this end, this paper aims to propose a strategic optimization model for enhancing solvers that compete to facilitate crypto transaction intents using CoWs and multi-chain AMMs. We further outline the expected impacts and contributions of our model to the DeFi ecosystem. When integrated with cross-chain infrastructure (like the Picasso Network [13] and Mantis [14] developed by the Composable Foundation [15]) our opti-

2834-9946/24/$31.00 ©2024 IEEE
DOI 10.1109/Blockchain62396.2024.00022

mization model simplifies cross-chain token intent settlements and token exchanges, thereby improving completion rates and delivering significant benefits to users.

The remainder of this paper is organized as follows: Section II (Background and Related Work) defines and describes relevant concepts. Section III (Proposed Optimization Model) details a system designed to optimally solve intents within a cross-chain context. The experimental results are presented in Section IV (Experimental Results). Finally, Section V (Conclusion and Future Work) summarizes the findings and discusses areas for future research.

II. Background and Related Work

This section outlines the foundational concepts and prior research pertinent to our study, encompassing topics such as intents and solvers, CoWs, types of market makers, and cross-chain interoperability. These areas form the theoretical underpinnings of our proposed models and are critical for understanding the subsequent discussions.

A. Intents and Solvers

In the evolving blockchain and DeFi ecosystem, "intents" and "solvers" have become pivotal concepts, exemplified by platforms like Anoma [11], SUAVE by Flashbots [16], and CoW Swap by CoW Protocol [12]. An *intent* generally refers to a user-defined set of constraints for cryptocurrency transactions, outlined as an off-chain signed message indicating desired state transitions [11]. Unlike fixed transactions, intents allow for flexibility in the execution path, aiming for optimal outcomes such as cost savings.

Solvers, on the other hand, are entities that devise the most efficient execution pathway for these intents. They compete to provide the best price or terms for a user's order. The successful solver executes the transaction and potentially receives rewards [17]. This competition ensures users receive favorable transaction terms. Moreover, the use of uniform clearing price batch auctions within this system helps prevent miner extractable value (MEV) attacks by eliminating disparities in order fulfillment prices, thereby safeguarding against front-running and similar strategies [7].

B. Coincidence of Wants

Coincidence of Wants (CoWs) is an economic phenomenon wherein each party possess an item or items that the other party desires, and thus are able to exchange these items directly to meet their wants [7]. When applying this concept to blockchain and cryptocurrency, a CoW enables two users' orders to be matched for settlement without the need for an external market maker or liquidity provider. In the case of intents, this principle means that a user's intent can coincidentally be the opposite of another user's intent (e.g. one intent to swap A for B and another to swap B for A form a CoW).

In the intent settlement framework discussed presently, CoWs are one means in which solvers may settle user intents [17]. This is arguably the optimal means for intent settlement whenever CoWs are available, due to the lack of a need for an external market or liquidity (which thus eliminates any fees associated with use of external markets). Therefore, in the optimization model presented in this paper, CoWs are a prioritized form of intent settlement.

C. AMMs, CFMMs, and Cross-Chain Interoperability

Automated market makers automate the traditional market-making process, which typically requires maintaining a constant presence to buy or sell assets. By enabling users to place orders at algorithmically determined prices, AMMs facilitate efficient trading in decentralized markets [5]. By contrast, *constant function market makers* (CFMMs) are a specialized subset of AMMs that operate through smart contracts. Liquidity providers contribute capital to CFMMs, which then continuously facilitate trades based on a predefined trading strategy articulated through a function of its asset reserves [18]. This model ensures liquidity and price stability within DeFi platforms.

Several notable types of CFMMs include the constant product model, exemplified by Uniswap v2 [19], which maintains a constant product of the reserves of two assets to ensure liquidity regardless of market size. The constant sum model [5] trades assets such that the sum of the reserves remains unchanged. This model is suitable for markets with stable asset values. Meanwhile, the constant mean model, used by Balancer [20], employs a weighted average of several assets to define the trading function, accommodating diversified liquidity pools.

Cross-chain interoperability is crucial in blockchain technology, enabling asset and information flows between independent blockchains. It involves asset exchanges and communication between disparate ledgers, extending to synchronized transactions across multiple ecosystems [9]. This capability unlocks new functionalities in cryptocurrency, such as efficient asset bridging, unified governance, and enhanced cross-chain DeFi activities like liquidity pooling, lending, and yield farming. These functions significantly broaden potential user engagement and returns within the DeFi sector [21]. Prominent technologies enabling this include the Inter-Blockchain Communication Protocol, exemplifying progress in the field [10]. Despite advancements, there remains a lack of comprehensive frameworks for cross-chain intent settlement. This paper proposes an optimization model for solvers to establish a robust framework for this purpose.

III. Proposed Optimization Model

In this section, we first present a system overview of the proposed intent settlement framework. Then, we detail our optimization model, which consists of a main CoWs-based model and a CFMM-based nested optimization model. Finally, we discuss other constraints and considerations for refining the integrated model.

A. System Overview

The system presented in this section is an intent settlement framework designed to optimize the settlement of cryptocurrency transactions across multiple blockchain networks. The

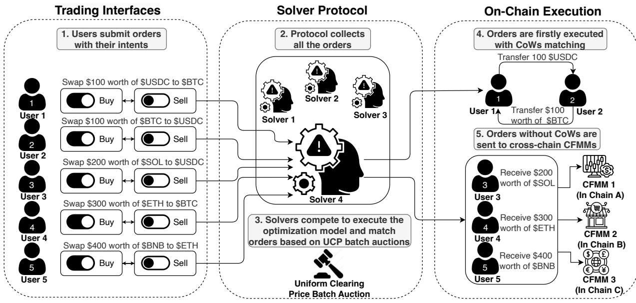

Fig. 1. System overview of the proposed intent settlement framework.

system's goal is to enhance the DeFi ecosystem's efficiency and interoperability.

The system overview is depicted in Fig. 1. The process begins when users submit their intents. The solver protocol then collects these intents. Each intent specifies parameters of the desired transaction, such as swaps between different cryptocurrencies. At the core of the model is a solver, which executes an optimization algorithm to match orders. In practice, to avoid a centralized single point of failure, the distributed solvers compete to solve the optimization model separately. The solver that provides the most user welfare will be selected as the winner and granted the right to settle the intent.

In this system, solvers make an initial attempt to ensure that direct matches between opposite user intentions are prioritized using the principle of CoWs, thus reducing transaction costs and increasing speed. This matching is facilitated through a uniform price batch auction system where transactions are aggregated and executed at a uniform price. This makes transaction order irrelevant within the block, undermining the ability for MEV bots to extract value [7]. For intents that cannot be directly matched via CoWs, the solver routes these orders to CFMMs located on various blockchain networks. This ensures that liquidity is utilized efficiently and that transactions are completed even when direct counterparties are not immediately available. The architecture supports multiple CFMMs across different chains, reflecting a robust approach to handling a diverse set of transaction scenarios and user demands. The model's integration of CoWs with cross-chain CFMM routing optimizes the flow of digital assets across disparate blockchain environments. Accordingly, the model significantly enhances user experience by providing faster, cheaper, and more reliable transaction settlement.

B. Main Optimization Model for CoWs

In this section, we discuss the CoWs optimization model that dictates the decision-making process among users. This is designed to maximize trading utility. The utility function integrates the individual utilities of a set of trading orders, factoring in the specifics of each order type—either limit-buy or limit-sell—and relevant parameters.

The utility functions for limit-buy and limit-sell orders are designed to accommodate both full and partial executions at a specified limit exchange rate π. This ensures adherence to quantity and price constraints. For a limit-buy order involving the purchase of x units of token k and the sale of y units of token j at rate π, the utility is $(x \cdot \pi - y) \cdot p_{b,k}$. Here, $x \cdot \pi$ is the ideal payment in token j for x units of token k, and y is the actual payment. The utility captures the net benefit, factoring in the internal clearing price $p_{b,k}$. Similarly, the utility for a limit-sell order which entails selling y units of token k to buy x units of token j at rate π, is calculated as $\left(x - \frac{y}{\pi}\right) \cdot p_{b,j}$. This formula evaluates the trade's efficiency when fully or partially executed. $\frac{y}{\pi}$ represents the received equivalent in token j. The difference between the ideal and actual tokens received, multiplied by the internal price $p_{b,j}$, quantifies the utility derived from the transaction. Thus, we can define objective as:

1) Objective: The main goal of the CoWs optimization model is to maximize the total trading utility u_{CoWs}:

$$\max \left[u_{CoWs} = \sum_{i=0}^{N_o - 1} (u_i - g_i) \right]$$

where u_i, g_i, and N_o are defined as follows:

- If order i is a limit-buy order, then $u_i = (x_i \cdot \pi - y_i) \cdot p_{b,k}$.
- If order i is a limit-sell order, then $u_i = \left(x_i - \frac{y_i}{\pi}\right) \cdot p_{b,j}$.
- g_i is the constant cost incurred by the order execution.
- N_o is the total order number.

2) Constraints: The trading rules for CoWs are adapted from the CoW Swap model [12]. CoW Swap uses the following definition of a set of valid trading constraints for limit-buy and limit-sell orders based on exchange rates and order size limits: "the order either is fully executed and the limit price is respected, or it is partially executed and is traded at its limit price." [22]. To simulate this scenario, we can set a binary variable b_i to enforce these constraints. $b_i = 1$ corresponds to one set of conditions and $b_i = 0$ corresponds to another.

Trading volume limit: We first set the maximum values of x_i and y_i. For all $i \in N_0$, we set:

$$x_i \leq x_{max}$$
$$y_i \leq y_{max}$$

Trading rules for limit-buy orders: For all $i \in N_0$ with limit-buy order, we set $\mathrm{M} = x_{\max} + 0.001$, where 0.001 serves as an arbitrary small adjustment value to ensure the constraints are properly implemented.

$$x_i \cdot (\pi - 0.001) \cdot b_i \leq y_i \leq x_i \cdot \pi$$
$$x_i + 0.001 - M \cdot (1 - b_i) \leq x_{\max} \leq x_i + M \cdot b_i$$

Trading rules for limit-sell orders: Similarly, for all $i \in N_0$ with limit-sell order, we set $\mathrm{N} = y_{\max} + 0.001$:

$$x_i \cdot (\pi - 0.001) \cdot b_i \leq y_i \leq x_i \cdot \pi$$
$$y_i + 0.001 - N \cdot (1 - b_i) \leq y_{\max} \leq y_i + N \cdot b_i$$

Here, we employ the Big-M formula, using large constants M and N for limit-buy and limit-sell orders respectively. This introduces a linear relaxation of binary constraints, simplifying the solution process. For both order types, setting $b_i = 1$ ensures that orders are executed exactly at the specified limit exchange rate ($\frac{y_i}{x_i} = \pi$), with $x_i < x_{\max}$ for buys and $y_i < y_{\max}$ for sells, respectively. Conversely, when $b_i = 0$, the constraint $\frac{y_i}{x_i} \leq \pi$ maintains that no orders exceed the limit exchange rate, with x_i set at $x_{\max}$ for buys and y_i at $y_{\max}$ for sells.

These constraints effectively capture the dynamics of trading with limit orders, accurately reflecting the strategic decision-making based on traders' risk tolerance and market expectations. Resultantly, we enhance the realism and applicability of our trading model.

C. Combined Volume Constraints

To execute asset swaps, we initially use the CoWs method due to its efficiency in facilitating direct trades among users. If a trade cannot be fully executed using CoWs, we employ CFMMs to process the remaining portion. This approach ensures swift and streamlined execution, optimizing the trading process and minimizing potential delays and slippage. In cases where CoWs is insufficient, CFMMs serve as an alternative to complete the transaction.

1) Partial Execution with CoWs: Given the available volume limit for the CoWs model:

$$x_i \leq V_{\text{CoWs},x_i}$$
$$y_i \leq V_{\text{CoWs},y_i}$$

Here, V_{CoWs,x_i} and V_{CoWs,y_i} are maximum volumes set by the solver, adjusted dynamically based on operational needs and strategy to optimize trading efficiency. Conversely, $x_{\max}$ and $y_{\max}$ (which are discussed in the previous section) are user-set upper limits on the order sizes, defining the maximum tokens to be traded. The distinction is critical as V_{CoWs,x_i} can be altered to facilitate more transactions within the limits of $x_{\max}$ and $y_{\max}$, or to manage risk and adapt to market conditions. This ensures transactions under CoWs operate within user constraints and maintain flexibility in execution.

2) Remaining Trades in CFMMs: When a trade surpasses the available volume in CoWs, the residual trade for assets is represented by Δx_i and Δy_i and is redirected to the CFMMs:

$$\Delta x_i = \max(0, x_i - V_{\text{CoWs},x_i})$$
$$\Delta y_i = \max(0, y_i - V_{\text{CoWs},y_i})$$

Here, the max function ensures the residual trade amounts (Δx_i and Δy_i) are not less than zero. Specifically, if the volume of the trade exceeds the available volume in CoWs orders, the surplus is calculated using this function and redirected to CFMMs. This ensures that any excess demand is properly managed.

CFMMs, with their inherent constant functions, are equipped to handle these trades. Taking the constant product CFMM as an example, for a liquidity pool with assets $e_{1,i}$ and $e_{2,i}$, the constant product is:

$$k_i = e_{1,i} \times e_{2,i}$$

For a given trade volume Δx_i in asset 1, the corresponding trade in asset 2 is:

$$\Delta y_i = \frac{k_i}{e_{1,i} + \Delta x_i} - e_{2,i}$$

To maintain pool liquidity, we can set:

$$\Delta x_i \leq \alpha \times e_{1,i}$$
$$\Delta y_i \leq \alpha \times e_{2,i}$$

where α is the pool liquidity coefficient.

D. Nested Optimization Model for CFMMs

To optimize a trade in CFMMs, we seek to maximize a utility function, $U(\Psi)$, that accounts for the total value of the trade and subtracts any fixed transaction costs. The variables and constraints encapsulate the trade volumes, liquidity considerations, and fees associated with the CFMMs.

To model an optimization problem for CFMM-based optimal routing across multiple blockchains, we need to incorporate an additional dimension that represents each blockchain platform. Let us denote the set of blockchain platforms as B and its cardinality (the number of blockchains) as n.

1) Objective: The main goal is to maximize the utility derived from trading while accounting for fixed transaction costs that may vary across different blockchains.

$$\max \left[u_{CFMMs} = U(\Psi) - \sum_{j=0}^{n-1} g'_j \eta_j \right]$$

where $U(\Psi)$, Ψ, g'_j, and $\eta_{i,j}$ are defined as follows:

- $U(\Psi)$ is an equation mapping the number of traded tokens to utility, which can be defined as an arbitrary function.
- Ψ is the total net number of tokens tendered to the CFMMs in the network. Specifically:

  $$\Psi = \sum_{j=1}^{n} \sum_{i=1}^{m} A_{i,j} (\Delta y_{i,j} - \Delta x_{i,j})$$

  where $A_{i,j}$ is the coefficient matrix that shows the weights of trade on CFMM i on blockchain j.
- g'_j is the constant cost incurred by the order execution.
- $\eta_{i,j}$ is a binary decision variable indicating whether or not to trade on CFMM i on blockchain j.

2) Constraints: The constraints considered for multi-chain CFMM transactions include:

Liquidity Constraint: This ensures the liquidity of any CFMM does not go below its initial state after a trade.

$$\varphi_{i,j}(R_{i,j} + \gamma_{i,j} \Delta x_{i,j} - \Delta y_{i,j}) \geq \varphi_{i,j}(R_{i,j}) \quad \forall i, j$$

where $\varphi_{i,j}$ is the trading function of CFMM i on blockchain j. Examples of trading functions are the product function, the sum function, and the weighted geometric mean function [23]. $R_{i,j}$ is the current reserves and $\gamma_{i,j}$ is the commission fee.

Trading Volume Limit: This bounds the trade volume based on the liquidity and decision to trade on a specific CFMM on a blockchain.

$$0 \leq \Delta x_{i,j} \leq \eta_{i,j} \Delta_{x,i,j}^{\max} \quad \forall i, j$$

$$\Delta y_{i,j} \geq 0 \quad \forall i, j$$

E. Other Considerations

1) Uniform Clearing Prices: For each batch (for CoWs and CFMMs), a uniform price is applied to constrain market volatility. Uniform pricing ensures all transactions in a batch are executed at the same price, enhancing market fairness and stability.

- $\forall i \in \{0, 1, \ldots, N_o - 1\}, x_i \cdot p_{b,j} = y_i \cdot p_{b,k}$
- $\forall i \in \{0, 1, \ldots, N_c - 1\}, a_{1,i} \cdot p_{b,j_{u,i}} = a_{2,i} \cdot p_{b,k_{u,i}}$

2) Gas Cost Incorporation: In the optimization of trading models, accounting for gas costs is critical for achieving realistic assessments of net gains and losses. Each order incurs not only a straightforward gas cost g_i in the CoWs model, but also complex and varied costs when routed through CFMMs. Specifically, CFMM gas costs g_i' can be broken down into three main components: $g_{\text{swap},i}$, $g_{\text{delay},i}$, and $g_{\text{network},i}$.

$$g_i' = g_{\text{swap},i} + g_{\text{delay},i} + g_{\text{network},i}$$

$g_{\text{swap},i}$ covers transaction fees for asset swaps facilitated by the market maker's smart contracts. $g_{\text{delay},i}$ accounts for costs associated with execution delays, which can arise from blockchain congestion or sequential processing, reflecting price slippage and opportunity costs. Lastly, $g_{\text{network},i}$ includes fees paid to the blockchain network, varying with network demand, which ensure prompt trade execution. These components together encapsulate the primary expenses incurred during CFMM routing.

The total gas cost for order i is then: $g_{total,i} = g_i + g_i'$. The constraints related to the total gas costs include:

$$\sum_{i=1}^{N_o} g_{total,i} \leq G$$

Based on the whole model presented in this section, the flow of transaction processing can be described as an integrated optimization algorithm (as shown in Algorithm 1).

Algorithm 1 Integrated Optimization Algorithm for Solvers

Require: N_0 (total number of orders), N_c (total number of CFMMs), $\mathbf{x}$ and $\mathbf{y}$ (trade volume vectors), π (exchange rate), $p_{b,k}$ and $p_{b,j}$ (buy and sell clearing prices), g_i (gas costs for CoWs), g_i' (gas costs for CFMMs)
Ensure: Optimized trade execution strategy via CoWs and CFMMs
Initialize binary variables b_i for CoWs to
Define utility functions u_i for orders and a global utility maximization goal
for $i = 0$ to $N_0 - 1$ **do** ▷ Iterate through each order
  if Order i is limit-buy **then**
    Compute utility $u_i = (x_i \cdot \pi - y_i) \cdot p_{b,k}$
  else if Order i is limit-sell **then**
    Compute utility $u_i = \left(x_i - \frac{y_i}{\pi}\right) \cdot p_{b,j}$
  end if
  Apply CoWs trading rules to attempt full execution
  if order i not fully executed **then**
    Calculate unexecuted volumes Δx_i, Δy_i
    Route remaining volume to CFMMs for execution
  end if
end for
for $i = 0$ to $N_c - 1$ **do** ▷ Optimize trade execution in CFMMs
  Apply CFMM trading constraints for liquidity and execution
end for
Compute total gas costs $g_{total,i} = g_i + g_i'$ for each order
Apply gas cost constraints and ensure net utility exceeds $g_{total,i}$
return Comprehensive trade execution strategy

IV. Experimental Results

This section shows the experimental results for our proposed optimization model. We first describe the settings and then present the experimental results for CoWs, CFMMs, and the integrated optimization model.

A. Experiment Settings

1) Solver Configuration: The experiments were conducted using the Gurobi Optimizer version 9.5.1. The optimization model is implemented using Python.

2) Order Details: Orders are represented as tuples consisting of buy/sell token indices j and k, buy and sell limits x_m and y_m, exchange rate π, and order type t (where 'b' indicates a limit-buy order and 's' indicates a limit-sell order). Below are the details of the simulated five orders:

TABLE I
Detailed information of each simulated order

Order	j	k	x_m	y_m	π	t
1	0	1	11	74	6.73	b
2	0	1	65	10	0.15	b
3	0	1	57	89	1.56	b
4	1	0	73	100	1.37	b
5	1	0	35	56	1.6	s

3) CFMM Configuration: Experiments involved five simulated CFMMs. Each was on a different blockchain with fixed transaction costs, fees, token indices, and specific reserves:

TABLE II
CFMM transaction costs, fees, reserves, and token indices

CFMM	Transaction Cost	Fee	Token Indices	Reserves
1	0.1	2%	[0, 1, 2]	$[3, 0.2, 1]$
2	0.2	1%	[0, 1]	$[10, 1]$
3	0.15	4%	[1, 2]	$[1, 10]$
4	0.25	3%	[0, 2]	$[20, 50]$
5	0.1	1%	[0, 2]	$[10, 10]$

B. Optimization Model of CoWs

The four figures (Section III-E2) in the first row of Fig. 2 depict variations in objective values in relation to order buy and sell limits, under different exchange rate scenarios. Notably, the first figure illustrates a declining trend in objective values as the buy limit increases for a limit-buy order scenario. This suggests that higher buy limits may lead to less favorable outcomes under certain conditions, potentially due to diminishing returns on increased buy limits. Conversely, the second figure in the row shows an ascending trend in objective values with increasing sell limits, indicating that higher sell limits can enhance outcomes by facilitating better matching opportunities or more favorable sell conditions. The third and fourth diagrams in the sequence further investigate the complex interplay between order limits and objective values. Within the context of a limit-sell order, the objective value is shown to escalate swiftly. This value reaches a peak as both the buy limit and sell limit are increased, then declines and subsequently stabilizes. The time required to reach this peak correlates with the value of π. These observations underscore the intricacies involved in optimizing trading strategies within a simulated CoWs environment.

The second row of Fig. 2 presents figures that explore the relationship between clearing prices and various factors such as order limits and pricing strategies. The plots illustrate how clearing prices adjust in response to changes in the order buy and sell limits across different settings. In scenarios involving both limit-buy and limit-sell orders, the variation in the clearing price exhibits greater stability upon incrementing the sell limit. Here, the clearing price of token 1 consistently remains marginally higher, albeit with minimal fluctuation. Conversely, when the buy limit is elevated, the clearing prices of the two tokens exhibit more pronounced shifts and may even intersect. These dynamics may be attributed to the intricacies inherent in the settings of the order parameters and the constraints of the model.

The final row of Fig. 2 provides a detailed examination of the number of tokens transacted, exploring how this metric adapts to varying buy and sell limits as well as exchange rates. The analysis reveals that for limit-buy orders, as the buy limit increases, the number of tokens initially rises to a peak and then sharply declines. This trend suggests that, within a certain range of increased buy limits, initial increments can enhance economic benefits such as average pricing. However, surpassing a specific threshold may lead to increased costs due to constraints imposed by other market conditions. For limit-sell orders, the number of tokens remains more stable with increasing sell limits, indicating that the model is less sensitive to changes in sell limits.

C. Optimization Model of CFMMs

The diagrams in Fig. 3 offer a comprehensive analysis of dynamics across three different CFMM types: 1) constant

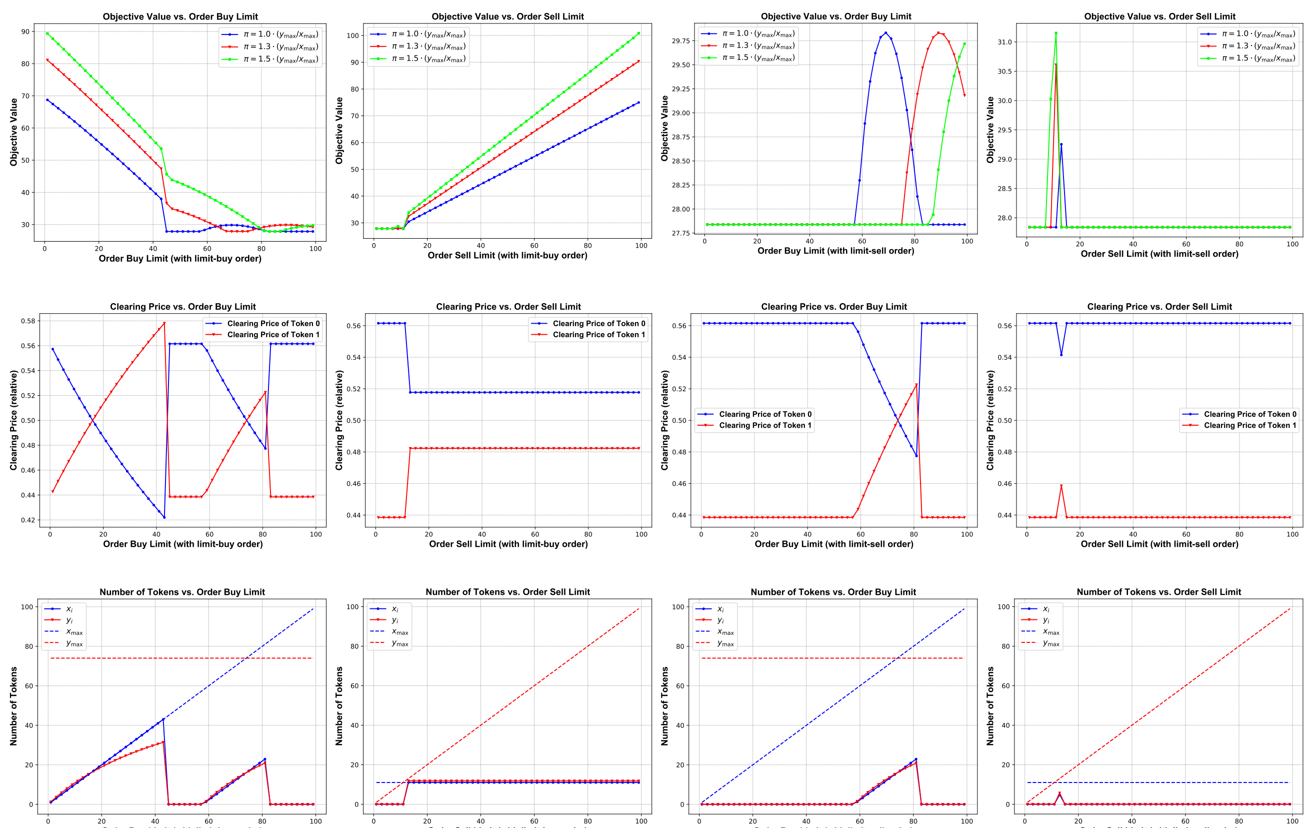

Fig. 2. Results of different simulation experiments for the optimization model of CoWs (fully executed).

product (using Uniswap v2 [19] as an example); 2) constant sum [5]; and 3) constant mean (using Balancer [20] as an example). Each set of four figures within this section explores the impact of varying parameters—blockchain transaction costs, trade volume limitations, CFMM commission fees, and reserves—on the objective value in relation to the number of token 1. Below are analyses for each CFMM type based on the figures:

The first set of figures (the first row in Fig. 3) illustrates the behavior of Uniswap v2 under different market conditions. Notably, the impact of blockchain transaction costs on the objective value shows a pronounced decrease as costs increase. This reflects the sensitivity of Uniswap v2 to on-chain transaction fees. Trade volume limitations reveal that tighter constraints lead to diminished objective values, indicating the critical role of liquidity in optimal market functioning. Similarly, variations in CFMM commission fees highlight the trade-offs between transaction costs for users and revenue for liquidity providers. Here, higher fees leading to lower objective values. Lastly, increasing the reserves demonstrates a positive effect on the objective value, emphasizing the importance of ample liquidity in fostering a robust trading environment.

For Constant Sum and Balancer CFMMs, the change in objective value follows a similar trend. Comparatively, the

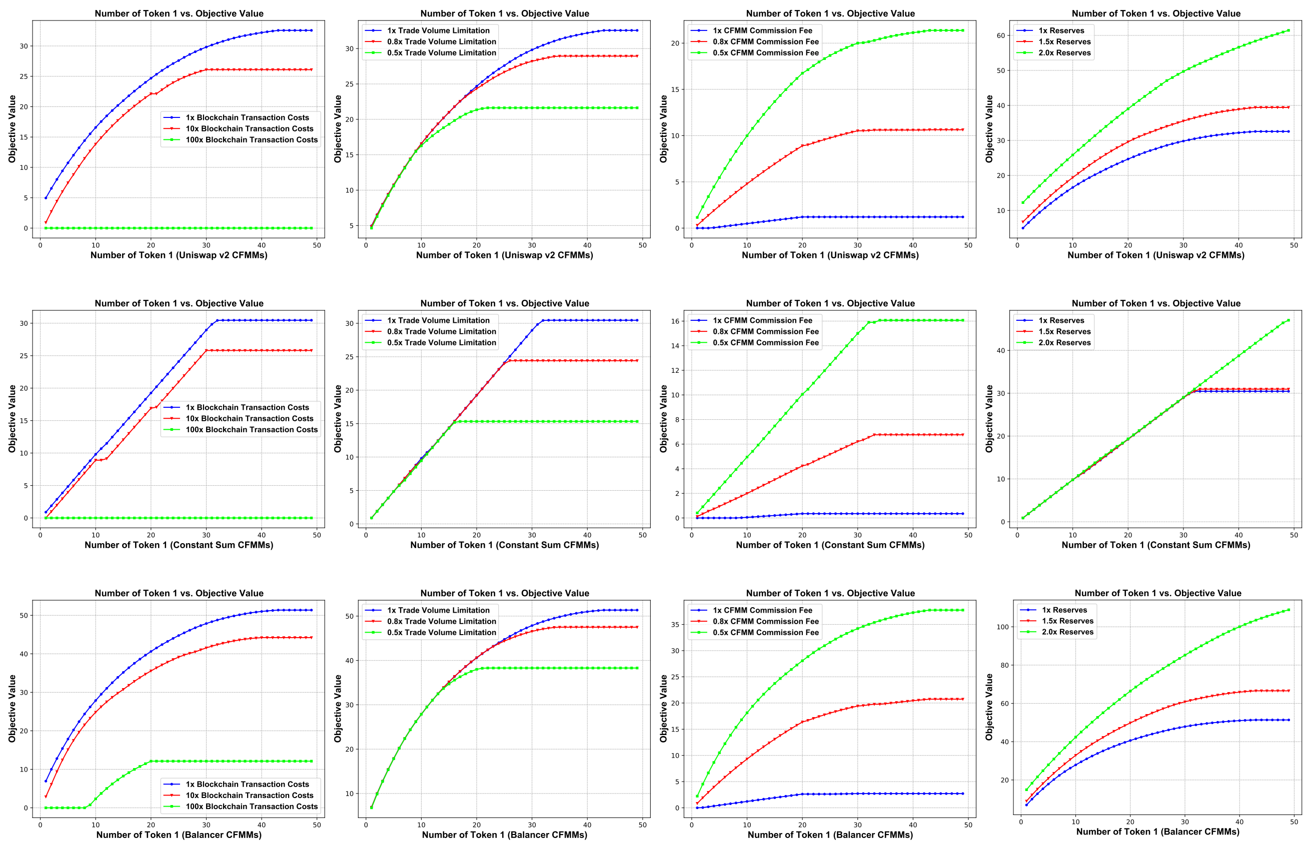

Fig. 3. Results of different simulation experiments for the optimization model of CFMMs routing (fully executed).

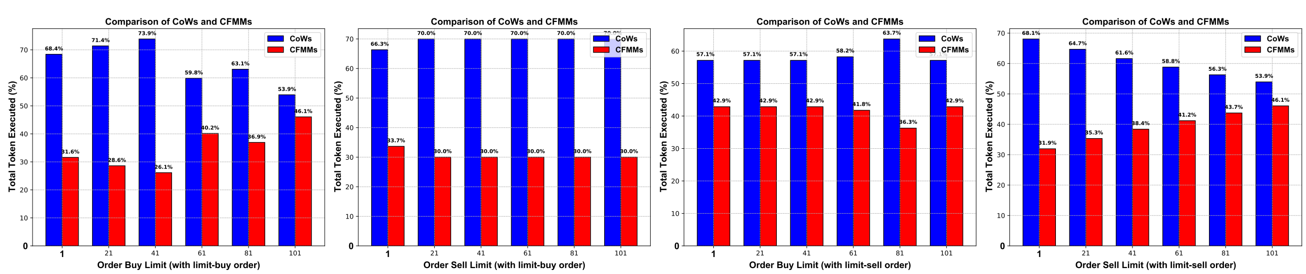

Fig. 4. Results of different simulation experiments with CoWs and CFMM routing integration optimization models.

growth trends of the three CFMMs vary significantly with increases in the number of token 1. Both Uniswap v2 and Balancer follow a logarithmic growth in objective value, whereas Constant Sum CFMMs exhibit initial linear growth before stabilizing. Overall, under similar data settings, Balancer achieves the highest objective values, followed by Uniswap v2, with Constant Sum CFMMs trailing. This comparison underscores the distinct operational dynamics and efficiency of each CFMM, highlighting Balancer's superior ability to optimize trading outcomes through strategic liquidity management.

D. Integrated Optimization Model

In Fig. 4, we analyze and compare the completion rates of CoWs and CFMMs for executing trades under various order constraints. We include both buy and sell orders with limit-buy and limit-sell conditions. Through a series of graphs, it becomes evident that the completion rate of CoWs and CFMMs varies significantly depending on the order type and limit. In general, the percentage of total tokens executed through CoWs surpasses that of CFMMs, indicating a higher efficiency in certain market conditions. Conversely, CFMMs have a higher execution ratio in the case of limit-sell orders. This is due to their flexible liquidity pools, which more effectively match sell orders with buy orders. Moreover, from a holistic perspective, the results show that our integrated optimization model results in a significant improvement in the order completion rate (26.1% to 46.1%) compared to the CoWs-only model under different experimental settings. Thus, our model improves user welfare and market fairness significantly.

V. Conclusion and Future Work

In this paper, we presented an intent settlement framework and a novel strategic optimization model for enhancing solvers that facilitate cryptocurrency transaction intents using the principles of CoWs and multi-chain AMMs. This model aims to overcome interoperability and efficiency challenges in DeFi ecosystems. By integrating cross-chain capabilities, our framework enables seamless interactions across different blockchain platforms, thereby enhancing liquidity and transaction flexibility. Our experimental results demonstrate that the proposed model effectively optimizes solver operations, ensuring transactions are settled efficiently and at minimal costs. By leveraging multi-chain AMMs and the CoWs principle, the model reduces the dependency on centralized mechanisms and enhances the privacy and security of transactions. The implementation of this model, combined with cross-chain bridging infrastructure developed by Composable, validates our approach and aligns with the projected impacts on the DeFi ecosystem.

Our future work will focus on two main areas: enhancing the solver algorithms and extending the cross-chain functionality. For the solver algorithms, we plan to incorporate more sophisticated decision-making capabilities that can dynamically adjust to varying market conditions. Additionally, we aim to broaden the cross-chain functionality to include more blockchains and improve the integration with existing and emerging DeFi applications.

References

[1] Z. Shi, V. Ivankovic, S. Farshidi, J. Surbiryala, H. Zhou, and Z. Zhao, "Awesome: an auction and witness enhanced sla model for decentralized cloud marketplaces," *Journal of Cloud Computing*, vol. 11, no. 1, p. 27, 2022.

[2] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[3] "Global projections 2023: Blockchain utilization in finance sector," 2024. [Online]. Available: https://www.reportlinker.com/p06244972/Blockchain-In-Banking-And-Financial-Services-Global-Market-Report.html?utm_source=GNW

[4] "2024 crypto market outlook," 2023. [Online]. Available: https://www.coinbase.com/institutional/research-insights/research/market-intelligence/2024-crypto-market-outlook

[5] V. Mohan, "Automated market makers and decentralized exchanges: a defi primer," *Financial Innovation*, vol. 8, p. 20, 2022. [Online]. Available: https://link.springer.com/article/10.1186/s40854-021-00314-5

[6] A. E. Roth, T. Sönmez, and M. U. Ünver, "Efficient kidney exchange: Coincidence of wants in markets with compatibility-based preferences," *American Economic Review*, vol. 97, no. 3, pp. 828–851, 2007.

[7] *Coincidence of Wants*. Accessed: Nov. 6, 2023. [Online]. Available: https://docs.cow.fi/overview/coincidence-of-wants

[8] Z. Shi, S. Farshidi, H. Zhou, and Z. Zhao, "An auction and witness enhanced trustworthy sla model for decentralized cloud marketplaces," in *Proceedings of the Conference on Information Technology for Social Good*, 2021, pp. 109–114.

[9] S. Ou, Weiand Huang, J. Zheng, Q. Zhang, G. Zeng, and W. Han, "An overview on cross-chain: Mechanism, platforms, challenges and advances," *Computer Networks*, vol. 218, p. 109378, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128622004121

[10] C. Goes, "The interblockchain communication protocol: An overview," *CoRR*, vol. abs/2006.15918, 2020. [Online]. Available: https://arxiv.org/abs/2006.15918

[11] C. Goes, A. Sun Yin, and A. Brink. (2022) *Anoma: a unified architecture for full-stack decentralised applications*. Accessed: Nov. 6, 2023. [Online]. Available: https://github.com/anoma/whitepaper/blob/main/whitepaper.pdf

[12] *CowSwap: First CoW Protocol UI*. Accessed: Nov. 6, 2023. [Online]. Available: https://github.com/cowprotocol/cowswap

[13] C. Foundation. Picasso network. Accessed: May 18, 2024. [Online]. Available: https://www.picasso.network/

[14] ——. Mantis whitepaper. Accessed: May 18, 2024. [Online]. Available: https://assets.website-files.com/65b28e756a8eda2e91e76ca4/6656289f21123d6215091555_MANTIS%20Whitepaper.pdf

[15] *Composable Foundation*. Accessed: Mar. 19, 2024. [Online]. Available: https://www.composablefoundation.com/

[16] R. Miller. (2023) *The MEVM, SUAVE Centauri, and Beyond*. Accessed: Nov. 20, 2023. [Online]. Available: https://writings.flashbots.net/mevm-suave-centauri-and-beyond

[17] *Introduction: Cow Protocol*. Accessed: Nov. 6, 2023. [Online]. Available: https://docs.cow.fi/solvers/solvers

[18] G. Ramseyer, M. Goyal, A. Goel, and D. Mazières, "Augmenting batch exchanges with constant function market makers," 2023.

[19] H. Adams, N. Zinsmeister, and D. Robinson. Uniswap v2 core. Accessed: March 15, 2024. [Online]. Available: https://uniswap.org/whitepaper.pdf

[20] F. Martinelli and N. Mushegian. A non-custodial portfolio manager, liquidity provider, and price sensor. Accessed: March 15, 2024. [Online]. Available: https://balancer.fi/whitepaper.pdf

[21] J. Zheng, D. K. C. Lee, and D. Quian, "An in-depth guide to cross-chain protocols under multi-chain world," *World Scientific Annual Review of Fintech*, 2023. [Online]. Available: https://ssrn.com/abstract=4476061

[22] T. Walther. (2018) *Multi-token Batch Auctions with Uniform Clearing Prices: Features and Models*. Accessed: Oct. 18, 2023. [Online]. Available: https://github.com/gnosis/dex-research/releases/download/v0.1.0/Multi-token.Batch.Auctions.with.Uniform.Clearing.Prices.pdf

[23] G. Angeris, A. Evans, T. Chitra, and S. Boyd, "Optimal routing for constant function market makers," in *Proceedings of the 23rd ACM Conference on Economics and Computation*, 2022, pp. 115–128.