Chair of Software Engineering for Business Information Systems
Department of Computer Science
School of Computation, Information and Technology
Technical University of Munich

# Detecting Arbitrages on a First-Come-First-Served Blockchain

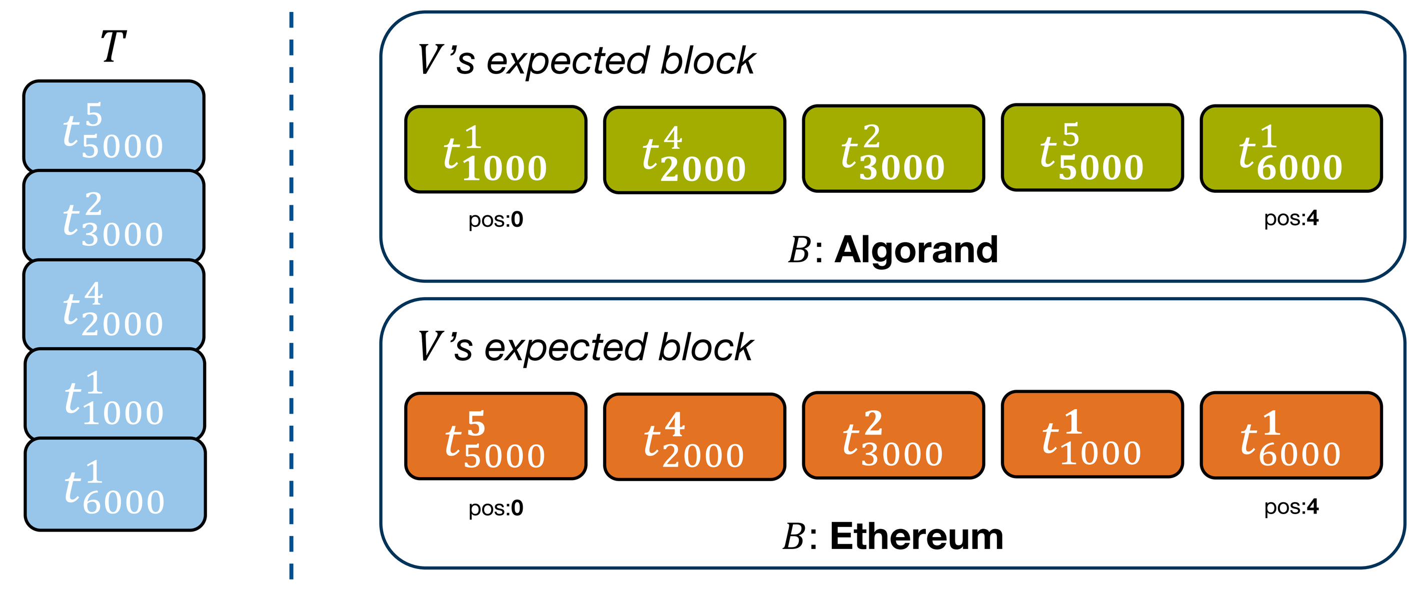
Burak Öz, Jonas Gebele, and Florian Matthes
{burak.oez, jonas.gebele, matthes}@tum.de

## Motivation

- **Decentralized Exchanges'** (DEXs) daily volumes exceed **multi-billion USD** [1].
- To have **efficient prices** on DEXs, **arbitrage trades** are necessary.
- According to EigenPhi [2], the **monthly profits** of arbitrageurs operating on the **Ethereum** blockchain **exceeds 2 million USD**.
- Previous research [3,4] explores arbitrage opportunity discovery and execution on such a network where transaction **fees** can **influence** the **execution order**.
- In **First-Come-First-Served** (FCFS) blockchains, block proposers sequence transactions in the order of appearance in their memory pool (mempool).
- Hence, **obtaining the desired block position** for an arbitrage opportunity requires **precise** transaction **issuance timing** and **propagation** to the network.
- We propose an **algorithmic approach for detecting arbitrage opportunities in an FCFS network** and demonstrate it on the **Algorand** blockchain, extending the work by Öz et al. in [5].

## Simplified First-Come-First-Served

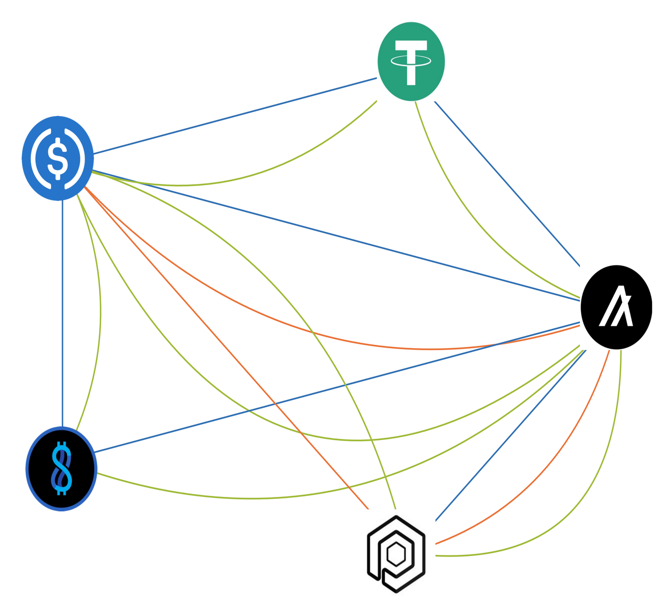
$B$: **blockchain**; $V$: **validator**; $T$: set of **transactions pending** in $V$'s **mempool**
$t_{ts}^{f} \in T$: a **transaction** in $T$ with **fee** $f$ and mempool entry **timestamp** $ts$



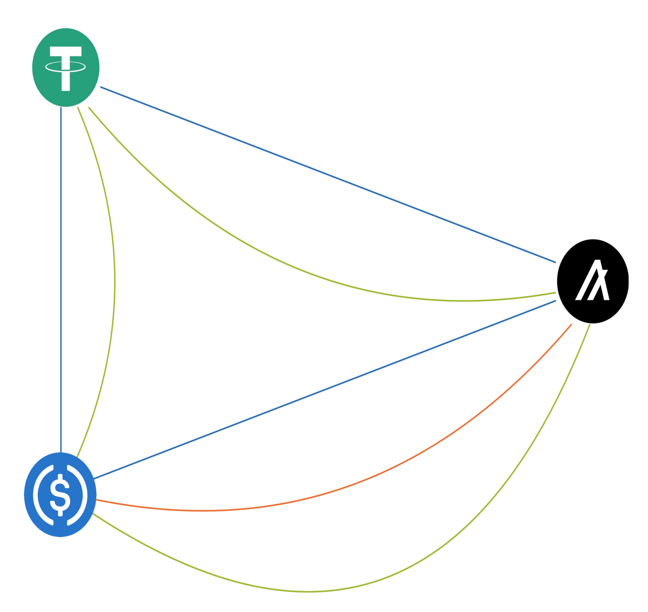

## Algorithmic Arbitrage Discovery

**Toy Model**

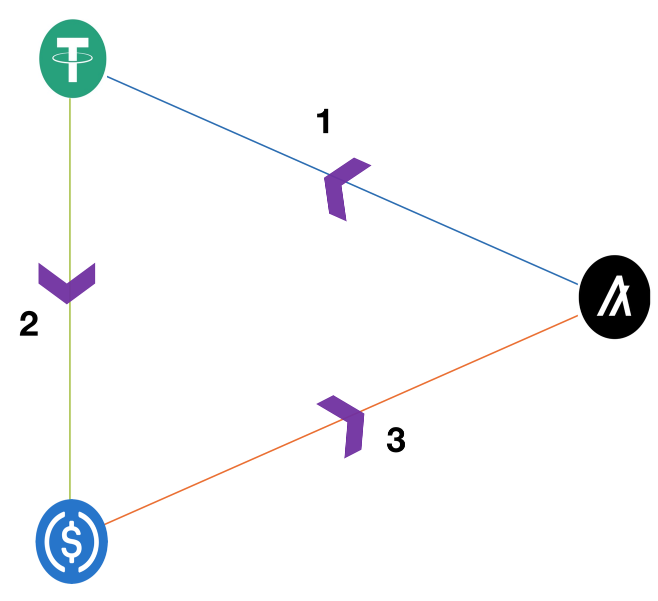
| | |
|---|---|
| Assets: | *ALGO, USDC, USDT, STBL, OPUL* |
| DEXs: | *Tinyman V1*, *Tinyman V2*, *Pact* |



**Exchange Multigraph**



***ALGO-USDC-USDT-ALGO* Cycles**




**Potential *ALGO* Arbitrage Route**

**Algorithm 1** Arbitrage Discovery and Execution

```
hops ← [2, 3, 4]
profit_assets ← ['ALGO', 'USDC', 'USDT']
price_data ← fetch_price_data()
exchange_graph ← build_exchange_graph(price_data)
arb_cycles ← get_arbitrage_cycles()
for each round in rounds do
    affected_arb_cycles ← get_affected_arb_cycles()
    for each arb_cycle in affected_arb_cycles do
        if diagonal_product() > 1 then
            inputs ← optimize_inputs()
            execute_arbitrage()
```

The algorithm can be applied:

i. After a block $round$, to find **block-state arbitrages** on arb cycles that had an affected pool in that $round$,
ii. After a transaction in the mempool that interacts with a relevant pool, to find **network-state arbitrages** on affected arb cycles.

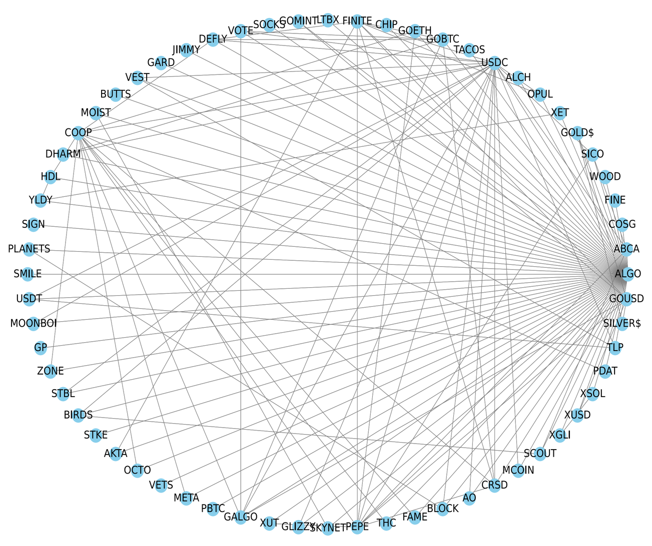
### Experiments

- We apply the algorithmic arbitrage discovery algorithm on **136 assets** exchanged in **255 pools** available on *Tinyman V1*, *Tinyman V2*, and *Pact*.
- The graph $G = (E, V)$ of assets ($V$) and pools ($E$), can be represented as an adjacency matrix, denoted with $A$. Then, the main diagonal of the $\boldsymbol{A^3}$ matrix yields the **3-hop cycles** for each asset.
- For each $l \in L$ length trading cycle $\boldsymbol{c_{pa}^{l}} \in \boldsymbol{C_{pa}^{l}}$ of a profit asset $\boldsymbol{pa} \in \boldsymbol{PA}$, we compute all possible implementations using the pools available for each leg of the cycle, denoted with:

$$\boldsymbol{I_{pa}^{l}} = \cup_{c_{pa}^{l} \in C_{pa}^{l}} I_{c_{pa}^{l}}$$

- The cumulative cycle implementation set of $PA$ is:

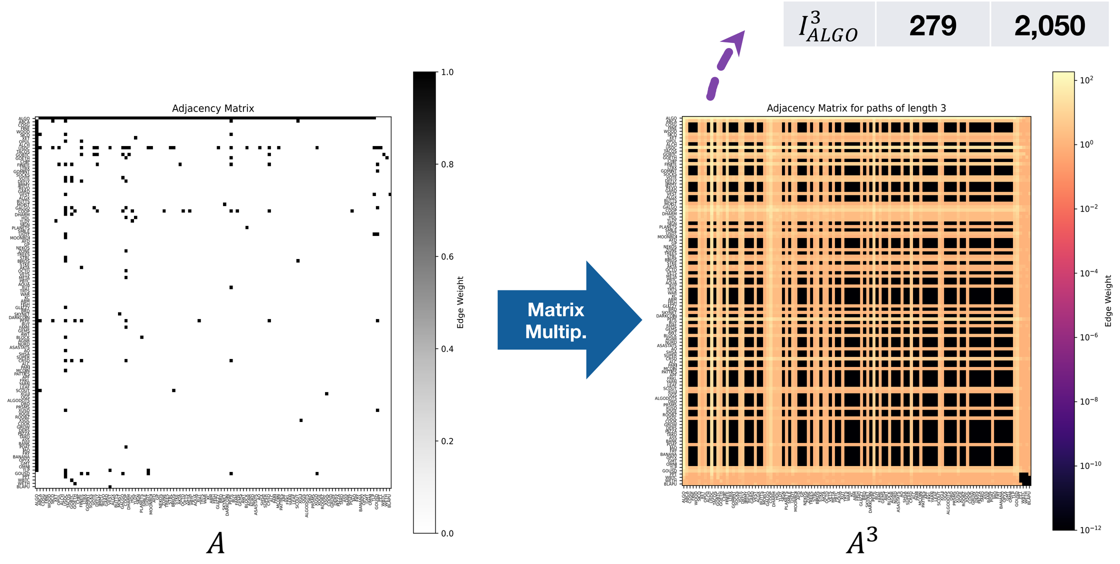
$$\boldsymbol{I_{total}} = \cup_{pa \in PA} \cup_{l \in L} I_{pa}^{l}$$



**Graph Representation**

**Potential Arb. Cycles of *ALGO***

| | 3-Hop | 4-Hop |
|---|---|---|
| $C_{ALGO}^{3}$ | **122** | **896** |
| $I_{ALGO}^{3}$ | **279** | **2,050** |




**Adjacency Matrix Representation**

## First Findings and Future Work

- We analyzed **300,007 blocks** built between 2023-10-05 and 2023-10-16, where **21,624 blocks** had **at least one relevant pool updated**.
- The algorithm finds, on average, **more than one unique block-state arbitrage opportunity** at every block **under 0.2**$s$ **runtime**.
- The **profitability** of the algorithm **depends on the available runtime**. Our experiments on the range [0.2, 19.8] show that 0.2 $s$ runtime discovers approximately **83% less value** than 19.8$s$. This difference **drops to less than 1%** as the runtime approaches the block time (3.3$s$).
- Unfortunately, even with 19.8$s$ runtime, the discovered profit levels are minimal as arbitrage **positions are efficiently closed inside the block they appear**. While the maximum realized profit by an arbitrageur is 167.17 USD, the algorithm finds, at most, a 32.26 USD opportunity that is fully closed in the following six blocks.
- The **future work** will focus on:
  - Collecting mempool data and operating the algorithm on the network level,
  - Optimizing the current prioritization rule utilized when examining arb cycles.

## Runtime Constraint

- In FCFS networks, the arbitrage opportunity detection algorithm's available **runtime depends** on the **first transaction's arrival time**, **changing a relevant pool's state**.
- Our initial experiments on Algorand show that relevant pools' states are **updated every six blocks** (median), providing roughly **19.8**$s$ **available runtime** to the algorithm, as Algorand's block time is around 3.3$s$.
- Interestingly, we observe a **maximum of 294 blocks** (~16min) **without updates** on a relevant pool state.

## Contact

Burak Öz
Technical University of Munich, sebis
burak.oez@tum.de

Technical University of Munich
School of Computation, Information and Technology
Department of Computer Science
Chair of Software Engineering for Business Information Systems (sebis)

References
[1] "Top Decentralized Exchanges Ranked by Volume." URL https://www.coingecko.com/en/exchanges/decentralized.
[2] "MEV Data | EigenPhi | Wisdom of DeFi." (2023) URL https://eigenphi.io/.
[3] Zhou, L., Qin, K., Cully, A., Livshits, B., Gervais, A. "On the Just-In-Time Discovery of Profit-Generating Transactions in DeFi Protocols." In 2021 IEEE Symposium on Security and Privacy (SP) 919–936 (2021) doi:10.1109/SP40001.2021.00113 iSSN: 2375-1207.
[4] Y. Wang, Y. Chen, H. Wu, L. Zhou, S. Deng, and R. Wattenhofer, "Cyclic Arbitrage in Decentralized Exchanges," in Companion Proceedings of the Web Conference 2022, ser. WWW '22. New York, NY, USA: Association for Computing Machinery, Aug. 2022, pp. 12–19. [Online]. Available: https://doi.org/10.1145/3487553.3524201.
[5] B. Öz, F. Rezabek, J. Gebele, F. Hoops, and F. Matthes, "A Study of MEV Extraction Techniques on a First-Come-First-Served Blockchain," Nov. 2023, arXiv:2308.06513 [cs] version: 2. [Online]. Available: http://arxiv.org/abs/2308.06513.