Transaction Fee Mechanism For Order-Sensitive Blockchain-based Applications

Mohammad Sadegh Nourbakhsh[1], Feng Hao[1], and Arshad Jhumka[1]

University of Warwick, Coventry, United Kingdom
{Mohammad-Sadegh.Nourbakhsh, feng.hao, H.A.Jhumka}@warwick.ac.uk

Abstract. Demand for blockchains such as Bitcoin and Ethereum far exceeds supply, thereby requiring a selection mechanism that, from a transaction pool, chooses a subset of transactions to be included "on-chain". Historically, every transaction submitted to the pool is associated with a bid (in the blockchain's native currency). A miner then decides which set of transactions from the pool should be included in a block. When the block is published, the bid of each included transaction is transferred from its creator to the miner. However, in newer applications such as *decentralised finance (DeFi)*, transaction inclusion in a block is no longer sufficient. In fact, the order in which the transaction is executed is of paramount importance. While research exists on mitigating transaction ordering manipulations, there is a lack of work on transaction fee mechanisms (TFMs) that are order-robust. This paper investigates order-robust TFMs from a mechanism design perspective and shows several impossibility results. For instance, we demonstrate that the recent EIP-1559 TFM is not incentive-compatible for order-sensitive transactions. On the other hand, we present and prove a necessary condition for an order-robust TFM.

Keywords: Blockchain · Transaction fee mechanism · Ordering · Incentive compatibility · Utility maximization.

1 Introduction

Blockchain and distributed ledger technologies (DLT) have gained immense popularity since the development of Bitcoin in 2009 [10]. These technologies offer an open, distributed, trustless, and tamper-resistant ledger that serves as the foundation for a new generation of financial services. Beyond facilitating transactions, blockchain enables the design and deployment of complex financial services and tools through smart contracts, particularly on platforms like Ethereum. The market capacity of blockchain technology reached a record-breaking 3 trillion dollars in November 2021 [1].

Ordering is a crucial aspect of traditional financial markets, and its significance extends to blockchain-based financial services. In these services, the order of transactions holds equal importance and can significantly impact users' profitability. To illustrate the transaction ordering problem, we present the following use case:

Example 1 (Eggs in sandwich shop). In a small town with limited egg availability, a sandwich shop operates by buying eggs from suppliers at the same price they sell them. The price of sandwiches containing eggs fluctuates based on the quantity of remaining eggs. Customers place orders on-site, and there is a maximum queue size at the shop, with the owner deciding which subset of customers can enter when the queue exceeds its capacity. In this scenario, Alice desires an omelette, Rachel wants a hotdog, Kevin orders a tuna sandwich, and Bob requests an egg mayo sandwich. Charlie is the supplier of eggs to the shop. Assuming all customers can stand in line, the order in which they join the queue becomes significant because the price of their sandwiches is influenced by the number of remaining eggs. For instance, if Charlie sells his eggs before Alice and Bob, he will receive a lower payment compared to selling them after the customers who specifically desire eggs. Likewise, if Alice stands behind Bob in the queue, she will pay a higher price for her sandwich compared to if she stands before Kevin. The positions of Kevin and Rachel in the queue are inconsequential as the prices of their sandwiches are unaffected by the number of remaining eggs.

Ordering's impact on users' outcomes can lead them to strategically execute transactions before or after a specific set of transactions in order to maximize their profits or utility. In traditional finance, front-running involves leveraging early access to market information for personal gain [6]. Similarly, in the blockchain system, front-running refers to submitting a transaction that is executed ahead of certain pending transactions, while back-running involves submitting a transaction to be executed after a particular set of transactions. These actions can introduce instability in order-sensitive blockchain applications.

Unlike traditional finance, where front-running typically requires exclusive and sometimes illicit access to stock data, the transparent nature of blockchain allows anyone to observe recently mined transactions and monitor the mempool, which serves as a shared buffer for transactions awaiting inclusion in blocks. This transparency creates opportunities for front-running, and the potential profitability of such opportunities attracts various users to the blockchain ecosystem.

Users exploit reordering opportunities to increase profits, resulting in competitive priority gas auctions (PGA) and higher transaction fees [5]. The mempool becomes a vulnerable space where reordering attacks can occur, known as the "dark forest" [12]. However, PGAs can negatively impact blockchain throughput and transaction fees for other transactions in blocks.

Our research primarily centres on Nakamoto-style consensus blockchain systems, wherein a randomly chosen miner is tasked with selecting and ordering transactions within each block. Consequently, we examine the incentives of agents, including both users and miners, at the block-level timescale. While it is conceivable for a group of agents to collude and employ sophisticated order manipulation strategies, the likelihood of such collusion is relatively low, mainly due to the random selection of miners for each block.

Miners in blockchain systems are incentivized through block rewards and transaction fees to maintain the blockchain's security. Various mechanisms have been proposed for transaction allocation and payment methods to miners since

the emergence of Bitcoin. These mechanisms fall under the Transaction Fee Mechanism (TFM), which determines user fees for block inclusion and miners' rewards for transaction allocation. However, existing TFMs are mainly tailored for cryptocurrency systems like Bitcoin that overlook the significance of transaction order in blockchain applications such as *decentralized finance* (DeFi). As a result, there are currently no TFMs specifically designed to address the needs of order-sensitive blockchain applications used in financial services.

This paper introduces a generalized model for TFM analysis that incorporates the impact of transaction ordering on agents' utilities. Currently, there is no existing TFM model that accounts for the state of transactions within a block. Our novel model is general enough to allow formalization and analysis of existing TFMs, helping to establish correctness conditions based on incentive compatibility. By applying this framework, we prove that the current TFM in Ethereum, namely *EIP-1559*, is not user incentive-compatible (UIC) for order-sensitive blockchain-based applications. Additionally, we prove that there is no deterministic TFM that satisfies both off-chain agreement proof and UIC. On the positive side, we identify a necessary condition for a TFM to be UIC.

2 Related Works

The TFM holds a pivotal role in blockchain systems, defining transaction costs and network security. Initially, the prevalent TFM in Bitcoin and Ethereum was the first price auction (FPA) [10]. However, Ethereum shifted to a posted-price TFM [15].

Prior TFM research mainly concentrated on the Bitcoin blockchain. Lavi et al. proposal [7] introduced a monopolistic auction for the Bitcoin TFM, where users in a block pay the minimum bid. However, it was shown by Andrew et al. [19] that in this auction, strategic bidding leads to zero revenue as users increase. Another TFM by Lavi et al. [7] is the Random Sampling Optimal Price (RSOP), addressing the user incentive compatibility (UIC) issue, but not myopic miner incentive compatible (MMIC).

To transcend TFM limitations, Basu et al. proposed a generalized second price auction-based TFM [3] aiming for UIC. This TFM is not MMIC, but as users increase, deviation revenue converges to zero. Moreover, Basu et al. [2] proposed Stablefees, an alternative TFM grounded in the second price auction. In a different vein, Chung et al.'s burning second price auction [4] confirms a random set of transactions in a block.

EIP-1559 marks a significant departure from the FPA and serves as the pioneering TFM implemented in a large-scale blockchain system. Comparative analyses have extensively examined the features of EIP-1559 in relation to FPA and other potential TFMs for Ethereum [14]. Furthermore, research has focused on assessing the stability of EIP-1559 and the influence of its base fee [8, 13]. Empirical studies have been conducted to investigate transaction fees, consensus security, and the impact of EIP-1559 on the Ethereum blockchain ecosystem [9]. Although EIP-1559 has introduced positive attributes such as predictable fees

and enhanced user experience, it has overlooked the impact of transaction ordering on agents' utilities. Notably, empirical evidence reveals that the extraction of miner extractable value (MEV) remains unaffected following the implementation of EIP-1559 [9].

Despite the efforts to improve TFMs, none of the proposed mechanisms has considered the effect of transaction orders on users' revenue by assuming a constant value for a transaction.

Several research papers in the field of blockchain systems have tackled the issue of order manipulation, considering it a problem in the context of changing transaction orders. For instance, [6] introduced the concept of front-running in blockchain systems and provided insights into the front-running behaviour exhibited by miners within the Ethereum blockchain.

The emergence of users actively seeking front-running opportunities has led to the rise of a competitive phenomenon known as the priority gas auction (PGA) within blockchain systems [5]. Additionally, the concept of miner-extractable value (MEV) is introduced by Daian et al. [5], which quantifies the potential profit that miners can obtain by manipulating the order of transactions in the blockchain. To assess the extent of extracted value within the Ethereum blockchain, empirical studies have been conducted [12, 17].

Exclusive mining services have emerged as a response to the challenges posed by order manipulation and MEV extraction in blockchain systems [16]. These services involve collusion between users and a specific miner, where users transmit their transactions directly to the service through a private channel instead of broadcasting them across the network. However, recent research has revealed that exclusive mining services not only facilitate the extraction of MEV but also provide significant benefits to the participating miners, amplifying their advantage in order manipulation and MEV extraction [11].

3 System Model

In this paper, we consider the blockchain system as responsible for maintaining the current state and executing a precisely ordered sequence of transactions that both read from and modify this state. There are four main components in the process, namely (i) users, (ii) miners, (iii) mempool and (iv) blockchain.

Users: Users are represented as agents responsible for generating transactions to be executed within the blockchain system. Each user is identified by a unique label c_i, ranging from 1 to N. Transactions generated by user c_i are denoted as tx_i and can encompass various actions such as buying stocks or selling shares.

Mempool: The mempool in the blockchain serves as a buffer where verified transactions are stored. When a transaction is generated by a user, it undergoes verification and is then added to the mempool. We assume that each user has at most one transaction in the mempool at any given time. Hence, a transaction tx_i generated by user c_i is represented as a 3-tuple: $\langle b_i, g_i, v_i \rangle$, where:

- b_i denotes the bid per unit size, indicating the amount the user c_i is willing to pay for executing the transaction.

– g_i represents the visible size of the transaction.
– v_i corresponds to the valuation of the transaction, indicating the maximum value that user c_i is willing to pay for the execution of tx_i. It is also referred to as the private value of the transaction.

We will revisit this transaction model when considering transaction ordering.
Miners: Miners select a subset of verified transactions from the mempool to form a block, which has a maximum size denoted by G. The transactions within the block are executed in the specified order, leading to updates in the blockchain's state.

When a block is added to the blockchain, the miner receives a payment determined by the bids associated with each transaction in the block. Therefore, we assume that miners are rational and will select transactions in a manner that maximizes their payment.
Blockchain: A blockchain is a sequence of blocks denoted as $B_1, B_2, \ldots, B_{k-1}$, with the initial block called the *genesis* block (B_1). The current block being added to the blockchain is B_k, and the entire block history is denoted as H.

4 Background

In this section, we provide a brief overview of the TFMs from an unordered set of transactions in a block. TFM models proposed by [4, 15] focused on the allocation of transactions to the block. Historically, a TFM is formalized as a 3-tuple of rules: (i) Allocation rule, (ii) payment and (iii) miner revenue rule.

Definition 1 (Allocation rule). *The allocation rule is a function x from the on-chain history H and mempool M to a binary value $x_i(H, M)$ for each pending transaction $tx_i \in M$. $x_i(H, M) = 1$ means that transaction tx_i is allocated to the current block B_k.*

A trivial allocation will be to allocate all transactions to a block. However, this may exceed the maximum permissible size of a block (G). Thus, we have the notion of a feasible allocation.

Definition 2 (Feasible allocation rule). *An allocation rule is feasible if, for every possible history H and mempool M:*

$$\sum_{tx_i \in M} g_i \cdot x_i(H, M) \leq G \tag{1}$$

where G is the maximum permissible size of a block.

A feasible allocation rule is responsible for assigning transactions to a block while respecting the maximum block size. The payment rule determines how transaction fees are transferred when a transaction is added to the blockchain.

Definition 3 (Payment rule). *A payment rule is a function p from the on-chain history H and allocated transactions in B_k to a non-negative number $p_i(H, B_k)$ for each transaction tx_i in B_k.*

$p_i(H, B_k)$ indicates the cost (per unit size) of allocating tx_i in block B_k that user c_i should pay.

Definition 4 (Miner revenue rule). *A miner revenue rule is a function Mr from the on-chain history H and allocated transactions in B_k to a non-negative number $Mr_i(H, B_k)$ for each transaction tx_i in B_k.*

$Mr_i(H, B_k)$ indicates the price (per unit size) of allocating tx_i in block B_k for the miner.

Definition 5 (Transaction fee mechanism (TFM)). *A transaction fee mechanism (TFM) is a triple (x, p, Mr) in which x is a feasible allocation rule, p is a payment rule, and Mr is a miner revenue rule.*

The definitions provided earlier assume a constant value for transactions in a block, regardless of their order. However, in decentralized financial systems such as decentralized exchanges, the order of transactions within a block can greatly affect the utility and profit of users. To address this, a new model is needed that takes into account the impact of transaction ordering on agent utilities.

5 Theory

In this section, we now revisit some previous definitions in the context of transactions (re)ordering in a block.

5.1 Order oriented private value

Previous studies on TFMs have often assumed that a user's valuation of a transaction remains constant once it is included in a block. However, this assumption may not hold true for order-sensitive applications like DeFi. To address this, we introduce the concepts of order-robustness and order-sensitivity, which classify transactions based on whether their valuation depends on the presence and order of other transactions in the block.

Definition 6 (T-order robust transaction). *Transaction tx_i is order-robust in sequence T if, changing tx_i's position in the sequence T without changing $T \backslash \{tx_i\}$, the private value of tx_i remains constant, i.e., $T = T_p \cdot tx_i \cdot T_s$ and $T' = T_{p'} \cdot tx_i \cdot T_{s'}$, where $(T = T' \wedge T_p \neq T_{p'} \wedge T_s \neq T_{s'})$, the valuation of tx_i in T is the same as in T'.*

Remark 1. If tx_i is T-order robust, tx_i is T'-order robust, for every $T' \sqsubseteq T$, where $\sqsubseteq$ denotes subsequence.

We denote the set of all possible sequences of a set X by $\bar{X}$.

Definition 7 (Order-robust transaction). *Given a set of transactions in block B_k, a transaction $tx_i \in B_k$ is said to be order-robust in B_k if it is T-order robust for every possible sequence T of the transactions in B_k, i.e, $\forall\ T \in \bar{B_k}$, tx_i is T-order robust.*

Definition 8 (Order-sensitive transaction). *A transaction $tx_i \in B_k$ is said to be order-sensitive in a transactions block B_k if tx_i is not order-robust in B_k.*

Definition 9 (Biggest order-sensitive subsequence (BOS)). *The biggest order-sensitive subsequence of a transaction tx_i (BOS_i) is the biggest subsequence $T \sqsubseteq \bar{B}_k$ such that removing each $tx_j \in T$ changes tx_i's private value.*

Example 2. In the sandwich shop Example 1, assume the maximum queue size is 3 and Alice's transaction tx_A (omelette):

- *Given $B_k = \{Alice, Rachel, Kevin\}$, tx_A is T-order-robust in the sequence $T = \langle Rachel, Alice, Kevin\rangle$ as the private value of tx_A is constant in every place in $T \backslash \{Alice\} = \{Rachel, Kevin\}$.*

- *Given $B_k = \{Alice, Rachel, Bob\}$, tx_A is order-sensitive as there exists $T' = \langle Rachel, Bob, Alice\rangle$, tx_A is not T'-order-robust & $BOS_A = \{Bob\}$ in T'.*

- *Kevin's transaction tx_K is order-robust in ($B_K \subseteq \{Alice, Bob, Charlie, Kevin, Rachel\} \wedge |B_k| = 3 \wedge tx_K \in B_k$), as it is T-order robust for every possible $T \sqsubseteq \bar{B}_k$.*

Definition 10 (Sensitive mempool). *We say that a mempool $\mathcal{M}$ is a sensitive mempool if there is at least one order-sensitive transaction in the mempool $\mathcal{M}$, i.e., $\exists B_k \subseteq \mathcal{M}, \exists tx_i \in B_k$ such that tx_i is order-sensitive in B_k.*

5.2 Generalized TFM modelling

A TFM is a crucial component of the blockchain protocol that determines which transactions are included in a block and their order within the block. It also specifies the transaction fees users need to pay for inclusion and the revenue received by the miner. To capture the impact of transaction ordering, we modify the TFM by introducing a *placement* rule, which considers the order of transactions. The placement rule serves as an allocation rule that incorporates transaction order. Placement rule captures the effect of the order of transactions in a block. For the same subset of transactions, different placement sequences may have different outcomes for both users (with order-sensitive transactions) and the miner. Additionally, we generalize the payment rule and miner revenue rule to account for transaction orders. The modified rules are as follows:

Definition 11 (Placement rule). *The placement rule is a function X from the on-chain history H and mempool M to a binary vector $X_i(H, M)$ for each pending transaction $tx_i \in M$.*

- *$X_i(H, M)[s]$ indicates the value of the placement vector $X_i(H, M)$ for the order s. $X_i(H, M)[s] = 1$ if tx_i is found in order s in B_k, 0 otherwise.*
- *We denote by s_i, the rank of transaction tx_i in the block B_k, i.e, $X_i(H, M)[s_i] = 1$. If tx_i is not in the block, then $s_i = 0$ and $X_i(H, M)[s_i] = 0$.*

– *$X(H, M)$ is the placement matrix such that each row is a placement vector of a transaction in the mempool.*

$$X(H,M) = \begin{bmatrix} \cdots \\ \cdots \\ X_i(H,M) \\ \cdots \\ \cdots \end{bmatrix}$$

– *Block B_k is then given as:*

$$B_k = X^T \cdot M \tag{2}$$

However, the placement rule may attempt to place a transaction in more than one place. Thus, we need to constrain the placement rule, to give rise to a feasible placement rule.

Definition 12 (Feasible placement rule). *A placement rule is feasible if, for every possible history H and mempool M:*

1. *The placement rule should place each transaction tx_i in at most one place.*

$$\forall tx_i \in M : \sum_{\forall s} X_i(H,M)[s] \leq 1 \tag{3}$$

2. *The placement rule should assign each place in the block sequence to one transaction. Denoting the number of transactions in B_k by S, none of the places before S should remain empty, i.e., all ranks are allocated.*

$$\forall\ j, 1 \leq j \leq S, \exists tx_k \in M \cdot X_k[H,M][j] = 1 \tag{4}$$

3. *Two transactions cannot have the same rank in the block.*

$$\forall\ tx_i, tx_j \in B_k, i \neq j \cdot s_i \neq s_j. \tag{5}$$

4. *The placement rule should place transactions in a block with a cumulative size smaller or equal to the size of the block.*

$$\sum_{tx_i \in M} g_i \cdot x_i(H,M)[s_i] \leq G \tag{6}$$

While a placement rule is proposed by a TFM, miners still have complete control over the placement of transactions in the block.

Definition 13 (Payment rule). *A payment rule is a function P from the on-chain history H and ordered transactions in B_k through the placement matrix X, to a non-negative number $P_i(H, B_k, X)$ for each transaction tx_i in order s_i of B_k.*

$P_i(H, B_k, X)$ indicates the cost (per unit size) of an included transaction tx_i through placement matrix X in block B_k that user i should pay.

Definition 14 (Miner revenue rule). *A miner revenue rule is a function Mr from the on-chain history H and placed transactions, through a feasible placement rule, in B_k to a non-negative number $mr_i(H, B_k, X)$ for each transaction tx_i with placement vector x_i in B_k. $mr_i(H, B_k, x_i)$ indicates the prize (per unit) of placing tx_i in order s_i of block B_k for the miner.*

Example 3 (Payment and (miner) revenue in sandwich shop). In the sandwich shop Example 1, Alice will pay $P_A(H, B_k, X)$ which is a function of the number of eggs in the shop (H), and placed people (X) in the current queue (B_k).

Alice's payment $P_A(H, B_k, X)$ where X is placing transactions in $B_k = \{Alice, Rachel, Bob\}$ is different from $P_A(H, B_k, X')$ where X' is placing transactions in $B'_k = \{Charlie, Alice, Bob\}$.

The sandwich shop's revenue from serving an omelette for Alice is $Mr_A(H, B_k, X)$. Same as above, the revenue is a function of placing people in the queue.

Definition 15 (Generalized Transaction Fee Mechanism (GTFM)). *A generalized transaction fee mechanism (GTFM) is a triple (X, P, Mr) in which X is a feasible placement rule, P is a payment rule, and Mr is a miner revenue rule.*

Based on the proposed Generalized TFM modelling, we aim to model several previously known and proposed TFMs using the rules proposed in the previous section. Our objective is to demonstrate that our proposed model can accurately represent these existing TFMs.

Example 4 (First price auction (FPA) [10]). The (intended) placement rule for FPA is to include a feasible subset of transactions that maximizes the sum of the size-weighted bids. The payment rule is equal to the miner revenue rule and both are independent of blockchain history and other transactions in the block.

- $P_i(B_k) = Mr_i(B_k) = b_i$
-
$$\begin{aligned} \arg\max_{X_i} \quad & \sum_{tx_i \in M} X_i(H, M)[s_i].b_i.g_i \\ \text{s.t.} \quad & \sum_{tx_i \in M} X_i(H, M)[s_i].g_i \leq G \end{aligned} \tag{7}$$

FPA's lack of UIC is evident [15], even without accounting for the impact of transaction ordering on users' utilities.

Example 5 (EIP-1559 [18]). EIP-1559 TFM is the current TFM of the Ethereum blockchain. The (intended) placement rule for EIP-1559 is to include a feasible subset of transactions that maximizes the sum of the size-weighted bids condition on which they bid at least the base fee. The base fee is "burning" (similar to giving away a small fee) and the miner receives the difference between users' payments and the base fee.

- $r = f(H)$ is the base fee which is a function of the blockchain history.

– $P_i(H, B_k, X) = min\{C_i, r+t_i\}$ which C_i is the fee cap and t_i is the maximum tip the user tends to pay the miner.
– $Mr_i(H, B_k, X) = P_i(H, B_k, X) - r = min\{C_i - r, t_i\}$
–

$$\begin{aligned} \max_{x_i} \quad & \sum_{tx_i \in M} \sum_{\forall s_i} x_i(H, M).min\{C_i - r, t_i\}.g_i \\ \text{s.t.} \quad & \sum_{tx_i \in M} \sum_{\forall s_i} x_i(H, M).g_i \leq G \end{aligned} \tag{8}$$

In EIP-1559, the mechanism designer assumes that the base fee is usually not excessively low. It means that the cumulative size of transactions in the mempool whose private value is more than the base fee is not bigger than the maximum block size.

$$\sum_{tx_i \in M: v_i \geq r} g_i \leq G \tag{9}$$

5.3 Agents' utilities and incentive compatibility

Based on our proposed generalized TFM modelling, considering the order-based private value of the transactions, we present formal definitions of the utilities of miners and users, who act as rational agents in the blockchain system and aim to optimize their utilities by adhering to the rules of the TFM. Our focus is on agents' incentives at the level of a single block. We assume that the addition of a transaction to the block has a common marginal cost μ, known to all users. Given these assumptions, rational users aim to optimize their utility by bidding for the intended placement of their transactions.

Miners In this paper, we consider a simplified model of miners' behaviour in which they have the ability to add fake transactions to the mempool without incurring any cost. We assume that miners are myopic and their utility is intended for the current block. This simplified model allows us to analyze the incentives for miners in the short term and examine how they can optimize their utility through TFM rules. The utility function of a miner (Eq. 10) shows that the utility of a miner depends on two arguments that the miner can control to maximize its utility. The first argument is the selection of transactions and their ranks in a block. The second part is adding fake transactions and their ranks to the block. A miner maximizes its utility by controlling both mentioned arguments based on the miner revenue rule of a TFM. Though a TFM is specified by the protocol designer based on a given placement rule, miners may choose to deviate from that rule. Thus, to ensure that miners are truthful, we require the TFM to be incentive-compatible.

Definition 16 (Myopic miner utility). *For a $TFM(X, P, Mr)$, on-chain history H, mempool M, and fake transactions F, utility of a myopic miner is:*

$$
\begin{aligned}
U_{miner}(B_k, F) := & \underbrace{\sum_{tx_i \in B_k \cap M} Mr_i(H, B_k, X_i) \cdot g_i}_{miner's\ revenue} \\
& - \underbrace{\sum_{tx_i \in B_k \cap F} P_i(H, B_k, X_i) \cdot g_i}_{fee\ for\ miner's\ fake\ transactions} \\
& - \underbrace{\mu \sum_{tx_i \in B_k} g_i}_{marginal\ costs} .
\end{aligned} \tag{10}
$$

The first term shows all ordered real transactions revenues. The second term indicates the cost to the miner of adding its fake transactions to the block. The last sum indicates the marginal cost of adding transactions.

Definition 17 (Incentive compatibility for myopic miner (MMIC)). *A $TFM(X, P, Mr)$ is incentive-compatible for a myopic miner (MMIC) if for every on-chain history H and mempool M, a myopic miner maximizes its utility (10) following the placement rule without creating any fake transactions, i.e., $F = \emptyset$. Then, $B_k = X^T \cdot M$*

Users Assuming rationality, users bid in order to maximize their utility. In light of the order-robust and order-sensitive classifications, we can formally define users' utility as a function of their private value and the payments made through the TFM's payment rule. Specifically, as noted earlier, the private value of a given transaction may depend on the presence and order of other transactions in the block. In fact, the transaction's private value is a function of its BOS in the block. By incorporating the placement matrix into the utility function, we can account for the impact of transaction order on users' valuations and thereby more accurately model their behaviour within the TFM framework.

Definition 18 (User utility function). *For a $TFM(X, P, Mr)$, on-chain history H and mempool M, the utility of user i, the owner of transaction tx_i, with private value v_i and bid b_i is:*

$$
u_i(b_i) := \begin{cases} (v_i - b_i).g_i, & s_i \neq 0 \\ 0 & otherwise \end{cases} \tag{11}
$$

In this paper, we do not assume that the private value is a constant. Rather, the private value v_i of a transaction tx_i depends on the expected private value V_i of tx_i, based on the last block, and on the (preceding) transactions that affects its value in the current block B_k, i.e., $v_i(V_i, BOS_i)$

Users are rational agents who aim to maximize their utilities by adhering to TFM rules. It is important to consider that miners also prioritize their own

utilities when placing transactions within a block. Therefore, we must examine whether utility maximization is feasible through TFM rules. For TFM rules to be effective, they must be incentive-compatible for users. In order to define this incentive compatibility, we first define the symmetric Ex Post Nash Equilibrium.

Definition 19 (Symmetric Ex Post Nash Equilibrium (Symmetric EPNE)). *Fix a $TFM(X, P, Mr)$ and block history H. A bidding strategy $b^*(\cdot)$ is a symmetric ex-post Nash equilibrium (symmetric EPNE), if for every mempool M, bidding $b^*(v_i)$ maximizes the utility (11) of user c_i conditioned on all $c_j \neq c_i$ following strategy $b^*(v_j)$.*

The existence of a symmetric EPNE in a TFM means that, if all users c_j follow b^*, c_i will not have any incentive to deviate. We can define incentive compatibility for users based on the existence of symmetric EPNE bidding strategy.

Definition 20 (Incentive compatibility for users (UIC)). *A $TFM(X, P, Mr)$ is user incentive compatible (UIC) if, for every on-chain history H, and mempool M, there is a symmetric EPNE bidding strategy.*

5.4 Off-chain agreements

In a blockchain environment, it is plausible for a miner and a group of users to form an off-chain coalition, with the goal of increasing the collective utility of the coalition. Thus, a TFM should be devised taking this possibility into account.

Definition 21 (Off-chain agreement). *For set C of transactions and miner m, an off-chain agreement (OCA) between C's creators and m specifies:*

1. *A bid matrix b^C, with b_i^C vector to be submitted with the transaction $tx_i \in C$.*
2. *A placing matrix X^C, indicating the transactions that the miner m will place in its block.*
3. *An agreement price B^C, with β_i from transaction tx_i's owner to miner m.*

In an OCA, each transaction tx_i's owner agrees to submit tx_i with an on-chain bid of b_i^C while transferring $\beta_i.g_i$ to the miner m off-chain; the miner, in turn, agrees to mine a block with the agreed-upon sequence of transactions C.

Definition 22 (Joint utility). *for an on-chain history H, the joint utility of a miner and some users with a set of transactions C with placement vector X^C in block B_k is:*

$$u_{joint}(m, C, X^C) = \sum_{tx_i \in C} (v_i - P_i(H, B_k, X) + Mr_i(H, B_k, X)).x_i^T.g_i \qquad (12)$$

Definition 23 (OCA-Proof). *A $TFM(X, P, Mr)$ is OCA-proof if, for every on-chain history H, no off-chain agreement between a miner m and any number of users with the transaction set C can increase their joint utility by deviation from TFM rules.*

6 Results

The author of [15] has shown that the EIP-1559 TFM is UIC when the base fee is not excessively low (9) and $min\{v_i, r + \mu\}$ is the users' symmetric EPNE. However, the incentive compatibility is satisfied assuming constant private values for transactions. However, for an order-sensitive mempool, the TFM cannot satisfy incentive compatibility for users.

Theorem 1 (EIP-1559 is not UIC for order-sensitive mempool.). *Fix an on-chain History H, a base fee $r = f(H)$, and a marginal cost μ. There is no symmetric EPNE bidding strategy in an order-sensitive mempool.*

Proof. We assume that the base fee is not excessively low, which means that the block size is sufficient to accommodate all transactions present in the mempool where the optimal fee for each transaction exceeds the base fee, i.e., $b^*(.) > r$. The author of [15] has shown that for excessively low base fee, EIP1559 is not UIC. We assume that there is a symmetric EPNE for all users: $b^* = min\{C_i^*, r + t_i^*\}$.

The myopic miner places transactions in the block in order to maximize its revenue (Eq. 8). The not excessively low assumption satisfies the feasible placement condition. Therefore, the miner places all transactions with $b^*(.) > r$ in the block.

As the places of transactions in the block do not affect the miner's revenue (Eq. 8), without loss of generality, we assume that the miner places the transactions in B_k based on their bid b^* in descending order.

As in order-sensitive mempool, there is a subset of order-sensitive transactions, We assume there is an order-sensitive transaction tx_i which is order-sensitive in a sequence $T \subseteq B_k$. The transaction owner can change his bid $b_i > b^*$ to get a higher rank (smaller BOS_i) in T to enhance his utility by increasing its private value $v_i(V_i, BOS_i)$. Therefore, b^* is not an EPNE for an order-sensitive transaction tx_i. Subsequently, there is no symmetric EPNE for users in EIP-1559 and it is not UIC. □

In light of the fact that EIP-1559 does not satisfy incentive compatibility for users in an order-sensitive mempool, our focus is on identifying a TFM that satisfies incentive compatibility for users. The rationale for prioritizing user incentive compatibility over miner incentive compatibility stems from the fact that miners, as rational agents, have access to the mempool and can maximize their utility by selecting the most profitable set of transactions to include in a block.

Theorem 2 (Impossibility theorem). *For an order-sensitive mempool, there is no deterministic TFM that satisfies both UIC and OCA-proof.*

Proof. In a deterministic TFM, a miner decides about the placement of transactions in the block. If we assume that the order of the transactions is forced by an external protocol, such as an order-fairness protocol. Therefore, the miner is at least responsible for the inclusion of transactions in a block. In both cases, we show that a TFM cannot satisfy UIC and OCA-proof

Assuming two order-sensitive transactions (tx_i, tx_j) exist in the mempool. The miner can make a block B_k in one of the following ways:

1. $S_1 = \langle tx_i, tx_j \rangle \sqsubseteq \bar{B_k}$
2. $S_2 = \langle tx_j, tx_i \rangle \sqsubseteq \bar{B_k}$
3. $S_3 = S_1 \backslash tx_i = \langle tx_j \rangle \sqsubseteq \bar{B_k}$

As we have assumed that the TFM is UIC, there is a symmetric EPNE for both users i and j. Without loss of generality, we can assume that based on the users' bids, the miner makes a block such as (i) following the placement rule. Now the following may happen:

1. The user j increase its transaction's bid convincing the miner to change the block such as (ii). Therefore, as the user can increase its revenue by deviation from the EPNE, the assumption of the existence of a symmetric EPNE and subsequently UIC is incorrect.
2. The user j makes an off-chain agreement with the miner to make the block such as (ii) or (iii). The difference in revenue increase can divide between the miner and the user. Therefore, as there is a possibility of OCA, the assumption of OCA-proof is not correct. Hence, a contradiction. □

Theorem 3 (Necessary condition for UIC TFM). *A TFM is said to be incentive compatible for users if, for every placed transaction tx_i in block B_k, BOS_i is empty.*

Proof. We consider two cases: (i) if a transaction tx_i is order-robust and (ii) when transaction tx_i is order-sensitive.

- Order-robust: if a transaction tx_i is order-robust, then, by definition, the private value of user u_i is independent of other transactions in the block.
- Order-sensitive: Assume that tx_i is placed at rank $r > s_i$. Since tx_i is order-sensitive, it means that its BOS_i at r is non-empty. This means that the private value of tx_i is lower than the maximum expected private value. To increase the utility of c_i, tx_i is moved higher up. However, when BOS_i is empty when tx_i is ranked at s_i, tx_i is order-robust for that given sequence. Hence, the private value of user c_i is constant, equal to the expected private value, when BOS_i is empty. □

Corollary 1 (UIC TFM). *If a TFM is UIC for order-robust transactions and if for every order-sensitive transaction tx_i in a block and BOS_i is empty, then TFM is UIC.*

7 Discussion

One of the primary motivations behind the replacement of the first price auction (FPA) with EIP-1559 in the Ethereum blockchain was to achieve incentive compatibility for both users and miners [15]. However, our analysis using the generalized TFM framework reveals that EIP-1559 is not UIC when considering an order-sensitive mempool. This finding is supported by the growing adoption of exclusive mining services such as Flashbots, which indicates a lack of UIC for

order-sensitive transactions [9]. Moreover, by disregarding the impact of transaction ordering, existing TFMs fail to satisfy the conditions of MMIC, UIC, and OCA-proof. Moreover, we have extended our results to demonstrate the impossibility of a deterministic TFM that can simultaneously satisfy both user incentive compatibility and OCA-proof conditions.

We have established that the necessary condition for a TFM to be UIC is the absence of any order-sensitive sub-sequence (BOS) within a block. This condition ensures that all transactions in the block are order-robust, providing users with confidence in the value they will receive. Alternatively, a UIC TFM can be achieved by having an order-robust mempool, meaning that services or functionalities within the blockchain system do not rely on transaction orders. This condition was implicitly present in previously considered UIC TFMs.

In this study, we have primarily operated under the assumption that all involved agents, encompassing users and miners, exhibit rational behaviour, driven by their desire to maximize their revenue within the context of a TFM. We acknowledge that this perspective excludes the influence of non-rational agents, including altruistic or malicious actors whose behaviour might deviate from incentive-based rationale. Nonetheless, it is our conjecture that over time, users' behaviour will tend towards rationality. Additionally, we have presumed that miners adhere to myopic decision-making. However, it is pertinent to note that in cases where a substantial coalition of miners collaborates, they might prioritize long-term utility optimization over immediate block-specific gains. Another dimension of limitation in our model pertains to our assumption of each user having a maximum of one transaction within a mempool per block. This assumption disregards scenarios where users may possess multiple transactions, each with potentially varying effects on their revenue contingent upon distinct placements within the block.

8 Conclusion and Future works

This paper has made significant contributions towards understanding the effect of transactions (re)ordering on blockchain transactions from agents' utility perspectives. By defining the private value of transactions as a function of other transactions in the block, we have captured the impact of order sensitivity. Additionally, we have proposed a generalized TFM modelling approach that considers the placement of transactions in the block. Through our analysis, we have shown that no deterministic TFM can satisfy both UIC and OCA-proof. Furthermore, we have identified and proved a necessary condition for a TFM to satisfy UIC. These findings can help improve the design of TFMs in blockchain systems to prevent order manipulation attacks.

As there exists no deterministic TFM that is UIC for order-sensitive mempool, we will investigate weaker notions of UIC and also a stochastic version of TFMs that are UIC, i.e., the pricing mechanism is varying rather than deterministic.

Our future research will delve into the exploration of alternative forms of UIC within the context of order-sensitive mempools. This involves investigating weaker variations of UIC and exploring stochastic versions of TFMs, where pricing mechanisms become variable rather than deterministic. We plan to address the limitations outlined in Section 7, including the incorporation of non-rational agent dynamics. Additionally, we plan to refine our model to account for scenarios where users may have multiple transactions within a mempool for a single block. This refinement will provide a more comprehensive understanding of how transaction order impacts revenue outcomes.

Acknowledgements

The second author is supported by EPSRC (EP/T014784/1).

References

1. Crypto total market cap 2010-2022. https://www.statista.com/statistics/730876/cryptocurrency-maket-value/, accessed: 2022-2-1
2. Basu, S., Easley, D., O'Hara, M., Sirer, E.G.: Stablefees: A predictable fee market for cryptocurrencies. Management Science (2023)
3. Basu, S., Easley, D.A., O'Hara, M., Sirer, E.G.: Towards a functional fee market for cryptocurrencies. CoRR **abs/1901.06830** (2019), http://arxiv.org/abs/1901.06830
4. Chung, H., Shi, E.: Foundations of Transaction Fee Mechanism Design, pp. 3856–3899 (2023). https://doi.org/10.1137/1.9781611977554.ch150, https://epubs.siam.org/doi/abs/10.1137/1.9781611977554.ch150
5. Daian, P., Goldfeder, S., Kell, T., Li, Y., Zhao, X., Bentov, I., Breidenbach, L., Juels, A.: Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In: 2020 IEEE Symposium on Security and Privacy (SP). pp. 910–927 (2020). https://doi.org/10.1109/SP40000.2020.00040
6. Eskandari, S., Moosavi, S., Clark, J.: Sok: Transparent dishonesty: Front-running attacks on blockchain. In: Bracciali, A., Clark, J., Pintore, F., Rønne, P.B., Sala, M. (eds.) Financial Cryptography and Data Security. pp. 170–189. Springer International Publishing, Cham (2020)
7. Lavi, R., Sattath, O., Zohar, A.: Redesigning bitcoin's fee market. ACM Trans. Econ. Comput. **10**(1) (may 2022). https://doi.org/10.1145/3530799, https://doi.org/10.1145/3530799
8. Leonardos, S., Monnot, B., Reijsbergen, D., Skoulakis, E., Piliouras, G.: Dynamical analysis of the eip-1559 ethereum fee market. In: Proceedings of the 3rd ACM Conference on Advances in Financial Technologies. p. 114–126. AFT '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3479722.3480993, https://0-doi-org.pugwash.lib.warwick.ac.uk/10.1145/3479722.3480993
9. Liu, Y., Lu, Y., Nayak, K., Zhang, F., Zhang, L., Zhao, Y.: Empirical analysis of eip-1559: Transaction fees, waiting times, and consensus security. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. p. 2099–2113. CCS '22, Association for Computing Machinery,

New York, NY, USA (2022). https://doi.org/10.1145/3548606.3559341, https://doi.org/10.1145/3548606.3559341
10. Nakamoto, S., et al.: Bitcoin. A peer-to-peer electronic cash system (2008)
11. Piet, J., Fairoze, J., Weaver, N.: Extracting godl [sic] from the salt mines: Ethereum miners extracting value. arXiv preprint arXiv:2203.15930 (2022)
12. Qin, K., Zhou, L., Gervais, A.: Quantifying blockchain extractable value: How dark is the forest? In: 2022 IEEE Symposium on Security and Privacy (SP). pp. 198–214 (2022). https://doi.org/10.1109/SP46214.2022.9833734
13. Reijsbergen, D., Sridhar, S., Monnot, B., Leonardos, S., Skoulakis, S., Piliouras, G.: Transaction fees on a honeymoon: Ethereum eip-1559 one month later. In: 2021 IEEE International Conference on Blockchain (Blockchain). pp. 196–204. IEEE Computer Society, Los Alamitos, CA, USA (dec 2021). https://doi.org/10.1109/Blockchain53845.2021.00034, https://doi.ieeecomputersociety.org/10.1109/Blockchain53845.2021.00034
14. Roughgarden, T.: Transaction fee mechanism design for the ethereum blockchain: An economic analysis of eip-1559. arXiv preprint arXiv:2012.00854 (2020)
15. Roughgarden, T.: Transaction fee mechanism design. ACM SIGecom Exchanges **19**(1), 52–55 (2021)
16. Strehle, E., Ante, L.: Exclusive mining of blockchain transactions (2020)
17. Torres, C.F., Camino, R., State, R.: Frontrunner jones and the raiders of the dark forest: An empirical study of frontrunning on the ethereum blockchain. In: USENIX Security 21. pp. 1343–1359. USENIX Association (Aug 2021)
18. Vitalik Buterin (@vbuterin), E.C.e.: Eip-1559: Fee market change for eth 1.0 chain (Apr 2019), https://eips.ethereum.org/EIPS/eip-1559
19. Yao, A.C.: An incentive analysis of some bitcoin fee designs. CoRR **abs/1811.02351** (2018), http://arxiv.org/abs/1811.02351