ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240924

No Transaction Fees? No Problem! Achieving Fairness in Transaction Fee Mechanism Design

Sankarshan Damle[a,1], **Varul Srivastava**[a,2] **and Sujit Gujar**[a,3]

[a]International Institute of Information Technology (IIIT), Hyderabad
ORCID (Sankarshan Damle): https://orcid.org/0000-0003-1460-6102, ORCID (Varul Srivastava): https://orcid.org/0000-0002-5662-0386, ORCID (Sujit Gujar): https://orcid.org/0000-0003-4634-7862

Abstract. The recently proposed Transaction Fee Mechanism (TFM) literature studies the strategic interaction between the miner of a block and the transaction creators (or users) in a blockchain. In a TFM, the miner includes transactions that maximize its utility while users submit fees for a slot in the block. The existing TFM literature focuses on satisfying standard incentive properties – which may limit widespread adoption. We argue that a TFM is "fair" to the transaction creators if it satisfies specific notions, namely Zero-fee Transaction Inclusion and Monotonicity. First, we prove that one generally cannot ensure both these properties and prevent a miner's strategic manipulation. We also show that existing TFMs either do not satisfy these notions or do so at a high cost to the miners' utility. As such, we introduce a novel TFM using on-chain randomness – `rTFM`. We prove that `rTFM` guarantees incentive compatibility for miners and users while satisfying our novel fairness constraints.

1 Introduction

Transaction Fee Mechanism (TFM) design, introduced in the seminal work by Roughgarden [24], considers the allocation problem of adding *transactions* to a *block* in blockchains such as Bitcoin [21] and Ethereum [6]. More concretely, the *miner* of the block adds transactions to its block from the pool of outstanding transactions (aka "mempool"). Transaction creators (henceforth *users*) optionally send a *transaction fee* as a commission to the miners to incentivize them to add their transactions.

TFM: Framework. The miner-user *strategic* interaction in a TFM is analogous to an auction setting. Indeed, Bitcoin implements a "first-price" auction with a miner maximizing its revenue by greedily adding transactions to its block from the mempool. A user's transaction fee captures its *valuation* for its transaction's inclusion. From [24], TFMs comprise (i) *allocation rule*, adding transactions from the mempool to a block, (ii) *payment rule*, for the payment to the miner, and (iii) *burning rule*[4]. Unlike classic auction settings, in TFMs, the miners have complete control over the transactions they add. Consequently, Roughgarden [24] introduces *miner incentive compatibility* (MIC) in addition to the standard *user incentive compatibility* (UIC). MIC states that the proposed TFM must incentivize miners to follow the intended allocation rule truthfully. UIC ensures that users offer their transaction's valuation as a transaction fee. Next, we have *off-chain collusion proofness* (OCAP) to curb miner-user off-chain collusion. Roughgarden [24] studies popular TFMs like first-price, second-price, and Ethereum's new dynamic posted-price mechanism, namely EIP-1559 [5], in terms of the properties they satisfy. Subsequent works [7, 9] enrich the TFM literature by proposing a dynamic posted-price mechanism and providing significant foundational results, respectively.

TFM: Challenges with Incentives. To satisfy UIC, MIC, and OCAP, TFMs introduce payment and burning rules based on transaction fees. However, we believe that (and as originally intended in Bitcoin [29]) TFMs must also support including transactions with zero fees. In practice, the fees are also higher than recommended [18]. Supporting zero-fee transactions will also benefit the adoption of currencies like Bitcoin and Ethereum. **First**, commission-based digital payment networks (e.g., VISA/MasterCard) are losing ground to commission-less networks (e.g., UPI) [28]. Commission-less payment networks admit ≈ 7.5 times *higher* transaction volume compared to their commission-based counterparts (rbi.org.in). **Second**, networks such as VISA/MasterCard charge the merchant a constant fraction of the transaction amount. This charge is *unlike* Bitcoin/Ethereum, whose transaction fees are independent of the transaction amount and paid by the user. For micropayments (e.g., paying for your morning coffee), these fees are unreasonable [20].

Our Approach and Contributions

Fairness Notions. We introduce (i) *Zero-fee Transaction Inclusion* (ZTi) and (ii) *Monotonicity* (Section 4). A TFM satisfies ZTi if it ensures that zero-fee transactions have a non-zero probability of getting included in the block.[5] However, guaranteeing ZTi must still ensure that the probability of a transaction's inclusion increases with an increase in its fee. E.g., randomly including transactions trivially ensures ZTi but may be unfair for a company that desires swift confirmation to meet the scheduled launch or if the transaction fixes a critical bug. To capture this, we introduce Monotonicity, which states

[1] Email: `sankarshan.damle@research.iiit.ac.in`

[2] Email: `varul.srivastava@research.iiit.ac.in`

[3] Email: `sujit.gujar@iiit.ac.in`

[4] Burning refers to removing tokens from the cryptocurrency's supply forever. E.g., by transferring them to unspendable addresses that can only receive tokens, thus making the tokens inaccessible.

[5] We assume that miners/users are myopic [7, 9, 24], i.e., they only consider their utility from the next block. Thus, ZTi deals with a transaction's probability of inclusion for the next block and *not* "eventual" confirmation. The myopic assumption is reasonable as pending transactions are typically never confirmed (e.g., in Ethereum).

that a TFM must ensure that transactions with a higher transaction fee have a greater probability of getting included in the block. Such a notion allows for *priority-based* transaction confirmation. Our two fairness notions combined imply that *every* transaction in the mempool has a non-zero probability of getting included in the block!

Given the impossibility of satisfying UIC, MIC, and OCAP simultaneously [7], we say a TFM is *fair* if it meets the above two notions, UIC and MIC. That is, fairness in TFMs w.r.t. the transaction creators (or users). Intuitively, as TFM design generally focuses on maximizing the miner's utility, it fails to satisfy ZTi. Moreover, we show that existing TFMs either do not satisfy our fairness notions or do so at a high cost to the miner's utility (Section 4.2). As such, we introduce Randomized TFM (`rTFM`), a TFM that satisfies our fairness notions, and study its incentive properties.

Randomized TFM (`rTFM`). We propose `rTFM` (Section 5), a TFM that satisfies our fairness notions while guaranteeing MIC (for an appropriate payment rule). In `rTFM`, we introduce a novel allocation rule that requires the miner to create two sets of transactions. In the first set, the miner optimally selects the transactions to add to its block (i.e., exactly like it currently does in Bitcoin). In the second set, the miner *uniformly* adds transactions from the mempool to its block but crucially receives *no* fee for these transactions. That is, the miner has no incentive to deviate from the uniform allocation in this set. The miner broadcasts both these sets, and we show that the blockchain network can randomly confirm one of the two sets through a *trusted* coin-flip mechanism (Section 5.1). Intuitively, such an allocation gives a non-zero probability of inclusion for zero-fee transactions due to the uniform sampling in the second set. As the miner has no control over the confirmed set, `rTFM` satisfies MIC for an appropriate payment rule, e.g., Bitcoin's first-price auction (Section 5.2).

2 Related Work

We now place our work concerning the existing literature for (i) TFM design and (ii) fairness in the context of blockchain.

Transaction Fee Mechanism (TFM) Design. Roughgarden [24] presents the seminal work that describes the "inclusion of transactions in a block" in the language of mechanism design. The author shows that EIP-1559 satisfies UIC and MIC and is OCAP (under some constraints on the base fee). Ferreira et al. [9] present a novel dynamic posted-price TFM with an equilibrium characterization of the posted-price. Most recently, Chung and Shi [7] provide several foundational results for TFM design based on underlying incentives and allocation rules. While the works [7, 9, 24, 30] are complementary, they do not focus on transaction fairness in TFMs.

Parallely, works also empirically analyze TFMs to optimize transaction fees [15, 27]. Tedeschi et al. [27] suggest a Deep Neural Network-based approach to predict miners' behavior in terms of including transactions in their blocks. The authors show that their approach reduces transaction fees and improves the confirmation time.

Fairness in Blockchain. Fairness is studied in various contexts, including network latency [12, 16], transaction ordering [2, 10, 13, 14, 22, 26] and price of transaction consumption [3, 25].

Fairness in transaction order focuses on the latency in transaction confirmation. E.g., miners may discriminate among specific transaction creators or only include transactions of the creators they know prior. This line of work [2, 10, 13, 14, 22, 26] does not model game-theoretic interactions and focuses on verifiable methods of ensuring "fairness" using cryptographic primitives. Moreover, there is no provision for the inclusion of zero-fee transactions. E.g., Sokolik and Rottenstreich [26] present a fair approach that prioritizes transactions with significant waiting time. Orda and Rottenstreich [22] provide techniques that enforce that transactions are allocated randomly to each block.

BitcoinF's [25] allocation rule splits the block with dedicated sections for standard and low-fee transactions. The authors argue that this allows miners to maximize their utility (through the standard section) while also processing low-fee transactions. With a strong assumption that transaction influx equals the cryptocurrency's throughput, they empirically argue that BitcoinF provides a lower consumption price. Also, they do not provide any theoretical guarantees for strategyproofness or fairness.

3 Preliminaries

We now summarize (i) the TFM and user model, (ii) relevant game-theoretic definitions, (iii) existing TFMs, and (iv) required blockchain preliminaries.

3.1 TFM Model

TFM design for public blockchains such as Bitcoin [21] and Ethereum [6] considers the following model. The blockchain's public ledger maintains the *state* and orders the sequence of *transactions* $t_1, t_2, \ldots, t_n, n \in \mathbb{N}_{\geq 1}$ that update the state. Let $s_i \in \mathbb{R}_{>0}$ be the size[6] of a transaction t_i. Each user i broadcasts its transaction t_i with a bid (per unit size) $b_i \in \mathbb{R}_{\geq 0}$. That is, the total bid is $s_i \cdot b_i$. The bid represents the amount user i is willing to pay for t_i, given its (per unit size) *private* valuation $\theta_i \in \mathbb{R}_{\geq 0}$. For security and practical reasons, each block has a *finite* capacity (denoted by $C \in \mathbb{R}_{>0}$). Miners create blocks, maintain a *mempool* of outstanding transactions ($M := \{t_1, \ldots, t_n\}$), and add a subset of these transactions to their blocks. Generally, the set of outstanding transactions is larger than the block size.

Transaction Fee Mechanism (TFM). Consider $\mathcal{H} = B_1, \ldots, B_{k-1}$ as the sequence of blocks denoting the on-chain history, current block B_k and mempool M. Designing a TFM involves defining (i) an *allocation* rule, which decides the transactions that get added to B_k, (ii) a *payment* rule describing the fraction of each transaction's bid that gets paid to the miner, and (iii) a *burning* rule, that is, the fraction of the amount that is removed from the supply, forever. An idiosyncrasy of blockchain involves *randomization* in transaction allocation. More concretely, with a "deterministic" TFM, we imply that a miner can include transactions in its block using any deterministic function. Whereas a "randomized" TFM implies that the miner selects the transactions to include through a random function[7]. To the TFM definition proposed in [24], we explicitly add the provision of TFMs being randomized.

Definition 1 (Transaction Fee Mechanism (TFM))**.** For a given on-chain history $\mathcal{H}$, the mempool M and the current block B_k with size C, a TFM is the tuple $\mathcal{T}^{TFM} = (\mathbf{x}, \mathbf{p}, \mathbf{q}, \tau)$ in which,

1. $\mathbf{x}$ is a feasible block allocation rule, i.e., $\sum_{t \in M} s_t \cdot x_t(\mathcal{H}, M) \leq C$ where $x_t(\cdot) \in \{0, 1\}, \forall t \in M$.
2. $\mathbf{p}$ is the payment rule with the payment for each transaction $t \in B_k$ denoted by $p_t(\mathcal{H}, B_k) \geq 0$.

[6] E.g., Ethereum transactions may be token transfers (smaller size) or sophisticated smart contract calls (larger size).

[7] TFMs may also use trustless on-chain randomness for transaction inclusion [7].

3. $\mathbf{q}$ is the burning rule with the amount of burned coins for each transaction $t \in B_k$ denoted by $q_t(\mathcal{H}, B_k) \geq 0$.
4. $\tau \in \{\tau_D, \tau_R\}$ is the mechanism's type – either deterministic (τ_D) or randomized (τ_R).

3.2 User Model and Incentive Properties

We now define the relevant incentive properties introduced in [24] for a TFM. We assume that the miners and bidding users are myopic [7, 9, 24, 30] – they are only concerned with their utility from the next block. For each user i, we have its (per unit size) valuation θ_i, bid b_i, and transaction size s_i. Let the vector $\mathbf{b}$ comprise all bids with $\mathbf{b}_{-i}$ representing all bids without user i. Given $\mathcal{T}^{TFM} = (\mathbf{x}, \mathbf{p}, \mathbf{q}, \tau)$ with $\mathcal{H}, M,$ and B_k, an user i's *quasi-linear* utility u_i is,

$$u_i(\mathbf{b}) := \begin{cases} (\theta_i - p_i(\cdot) - q_i(\cdot))\, s_i \textbf{ if } x_i = 1 \\ 0 \qquad \textbf{otherwise}. \end{cases} \tag{1}$$

User Incentive Compatibility (UIC). A strategic user i will select b_i such that it maximizes its utility defined in Eq. 1. As such, we now define UIC for a TFM.

Definition 2 (UIC [24]). *A TFM $\mathcal{T}^{TFM} = (\mathbf{x}, \mathbf{p}, \mathbf{q}, \tau)$ with $\mathcal{H}$, M, and B_k is UIC if – assuming the miner follows the allocation rule $\mathbf{x}$ – bidding θ_i for each user i maximizes u_i (Eq. 1), irrespective of the remaining bids. That is, $\forall i,\ u_i(b_i^\star = \theta_i, \mathbf{b}_{-i}) \geq u_i(b_i, \mathbf{b}_{-i}),\ \forall b_i$ and $\forall \mathbf{b}_{-i}$.*

Informally, UIC states that it is the best response for a user to submit its valuation as its transaction fee.

Myopic Miner Incentive Compatibility (MIC). In TFMs, the miner of block B_k has complete control over the set of transactions to add to B_k (i.e., implement an alternate allocation rule over the intended one). To deviate from the intended rule $\mathbf{x}$, a miner typically adds "fake" transactions to the mempool. For the set of fake transactions F (i.e., $F \subset M$) and for any $\mathcal{T}^{TFM} = (\mathbf{x}, \mathbf{p}, \mathbf{q}, \tau)$ with $\mathcal{H}$, M, and B_k we can write miner's utility u_{M} as follows [24]. Given $B_k = \{t \in M \mid x_t = 1\}$, we have

$$u_{\mathsf{M}}(B_k, F) := \sum_{t \in B_k \cap M \setminus F} s_t \cdot p_t(\cdot) - \sum_{t \in B_k \cap F} s_t \cdot q_t(\cdot). \tag{2}$$

The first term represents the miner's revenue, and the second term represents the fee burned from the miner's fake transactions. To maximize its utility, the miner performs the following optimization.

$$\begin{cases} \max_{\mathbf{x}'} & \sum_{t \in B_k \cap M \setminus F} x_t \cdot s_t \cdot p_t(\cdot) - \sum_{t \in B_k \cap F} x_t \cdot s_t \cdot q_t(\cdot) \\ \text{s.t.} & \sum_{t \in M} s_t \cdot x_t \leq C \text{ and } x_t(\mathcal{H}, M) \in \{0,1\}, \forall t \end{cases} \tag{3}$$

Given the possibility of a miner's strategic deviation, Roughgarden [24] introduces MIC.

Definition 3 (MIC [24]). *A TFM $\mathcal{T}^{TFM} = (\mathbf{x}, \mathbf{p}, \mathbf{q}, \tau)$ with $\mathcal{H}$, M, and B_k is MIC, if a miner maximizes u_{M} (Eq. 3) by not creating any fake transactions, $F = \emptyset$ and following the rule $\mathbf{x}$.*

Let `OPT` denote the miner's optimal utility from Eq. 3 (i.e., with $p_t = b_t$ and $q_t = 0,\ \forall t \in B_k$). Note that computing the optimal feasible set, say $\mathbf{x}^\star$, in Eq. 3 is NP-Hard since it reduces to KNAPSACK auctions [1]. Miners may instead adopt a greedy-based approach [24].

Off-chain Collusion Proof (OCAP). Another desirable property in TFM is OCAP, which deals with the off-chain collusion of the miner and a set of $c \in \mathbb{N}_{\geq 1}$ users. A TFM is c-OCAP if any coalition between the miner and set of users with cardinality c Pareto improves the intended allocation $\mathbf{x}$. As stated earlier, Chung and Shi [7] prove the impossibility of simultaneously satisfying UIC and 1-OCAP; thus, in this work, we focus only on MIC and UIC.

3.3 Popular TFMs and Their Properties

We now summarize some popular TFMs in literature.

First-price (FPA) TFM. Bitcoin employs a first-price TFM which can be expressed in the language of Definition 1 with $\mathcal{T}^{\text{FPA}} = (\mathbf{x}^{\text{FPA}}, \mathbf{p}^{\text{FPA}}, \mathbf{q}^{\text{FPA}}, \tau^{\text{FPA}})$. Here, $\mathbf{x}^{\text{FPA}}$ follows Eq. 3. For each $t_i \in B_k$ we have, $p_i^{\text{FPA}} = b_i$, $q_i^{\text{FPA}} = 0$ and $\tau^{\text{FPA}} = \tau_D$. FPA does not satisfy UIC but satisfies MIC [24].

Second-price (SPA) TFM. We denote the second-price TFM with $\mathcal{T}^{\text{SPA}} = (\mathbf{x}^{\text{SPA}}, \mathbf{p}^{\text{SPA}}, \mathbf{q}^{\text{SPA}}, \tau^{\text{SPA}})$. Here, $\mathbf{x}^{\text{SPA}}$ follows Eq. 3. Assuming $\bar{b}$ as the lowest winning bid, for each $t_i \in B_k$, we have[8], $p_i^{\text{SPA}} = \bar{b}$, $q_i^{\text{SPA}} = 0$ and $\tau^{\text{SPA}} = \tau_D$. SPA approximately satisfies UIC but does not satisfy MIC [24].

EIP-1559 [5]. Denoted with $\mathcal{T}^{1559} = (\mathbf{x}^{1559}, \mathbf{p}^{1559}, \mathbf{q}^{1559}, \tau^{1559})$, in EIP-1559, for each $t_i \in B_k$, we have $p_i^{1559}(\mathcal{H}, B_k) = b_i - \lambda$ where λ is the (dynamic) base fee[9], $q_i^{1559} = \lambda$ and $\tau^{1559} = \tau_D$. The miner maximizes its utility such that $\mathbf{x}^{1559}$ follows Eq. 3.

EIP-1559 satisfies UIC *only* if λ is *not* "excessively low" [23, Definition 5.6]. The base fee λ is excessively low if λ is small enough so that the number of transactions with a valuation greater than λ exceeds the block size. EIP-1559 also satisfies MIC.

BitcoinF [25]. We denote BitcoinF as $\mathcal{T}^B = (\mathbf{x}^B, \mathbf{p}^B, \mathbf{q}^B, \tau^B)$. Each user i creates *two* transactions offering a public constant fee $\delta \in \mathbb{R}_{>0}$ and $\delta + \hat{b_i}, \hat{b_i} \in \mathbb{R}_{>0}$ as fees. If one gets added, the other is nullified. The allocation rule $\mathbf{x}^B$ splits the block into $\alpha \in (0, 1]$ and $1 - \alpha$ fractions. The miner must first fill the $1 - \alpha$ section through FIFO collecting transactions with δ, after which it can greedily fill the α section. Let C_α and $C_{1-\alpha}$ denote the capacity of the α and $1 - \alpha$ sections, i.e., $C = C_\alpha + C_{1-\alpha}$. For each t_i in the α section, we have $p_i^B = \hat{b}_i + \delta$ and $q_i^B = 0$. Likewise, for each i in the $1 - \alpha$ section, we have $p_i^B = \delta$ and $q_i^B = 0$. Lastly, $\tau^B = \tau_D$. BitcoinF's optimization is as follows.

$$\left.\begin{array}{ll} \max_{\mathbf{x}^B} & \sum_{i \in M} x_i^B \cdot p_i^B(\mathcal{H}, B_k) \cdot s_i \\ \text{s.t.} & \sum_{t \in M, b_t \neq \delta} s_t \cdot x_t^B(\mathcal{H}, M) \leq C_\alpha \\ & \sum_{t \in M, b_t = \delta} s_t \cdot x_t^B(\mathcal{H}, M) = C_{1-\alpha} \text{ and} \\ & x_t^B(\mathcal{H}, M) \in \{0,1\}, \forall t \in M. \end{array}\right\} \tag{4}$$

As a warm-up result, we show that strategic miners in $\mathcal{T}^B$ may deviate, i.e., miners may include fake transactions in the $1-\alpha$ section of the block to increase their utility from the α section. Remark 1 captures this result.

Remark 1. *BitcoinF ($\mathcal{T}^B$) does not satisfy MIC.*

[8] Generally, SPAs require users to pay the highest losing bid. As payments cannot depend on transactions not part of a block, [24] suggests using the lowest winning bid as a proxy.

[9] λ is dynamic and depends on the network congestion. If the block size $> C$, then the congestion is higher, and λ is incremented by 12.5%. If the block size is $\leq C$, λ is decremented by 12.5% [24].

Proof. Consider the following example, where each transaction is of the same size. Let $n = 5$ such that the current block B_k can hold up to 8 transaction. Further, we have $\alpha = 3/4$. The miner must add (any) 2 transactions to the $1 - \alpha$ section first before greedily adding transactions to the α section. Whichever transactions from M the miner chooses to add to the $1 - \alpha$ section, it can strictly increase its utility by adding 2 fake transactions instead. That is, by adding these fake transactions, the miner can add the real transactions of M to the α section. Thus, BitcoinF's allocation rule does not satisfy MIC. □

Section 5 presents a novel TFM – namely, `rTFM` – that leverages specific blockchain and cryptographic primitives, as outlined next.

3.4 Blockchain and Cryptographic Preliminaries

Hash Functions. Given a security parameter $\lambda \in \mathbb{N}_{\geq 1}$, cryptographic hash functions are one-way functions defined as $\textsc{Hash} : \{0,1\}^* \rightarrow \{0,1\}^\lambda$. A hash function is (i) *collision-resistant* if the probability of any two distinct inputs x, y map to the same output with negligible probability, i.e., $\Pr[\textsc{Hash}(x) = \textsc{Hash}(y) | x \neq y] \leq \mathsf{negl}(\lambda)$ and (ii) *pre-image resistant* if the probability of inverting $\textsc{Hash}(x)$ is less than $\mathsf{negl}(\lambda)$. Here, $\mathsf{negl}(\lambda)$ denotes a negligible function in λ. E.g., SHA-256 [11].

Merkle Tree (MT) [17]. These are complete binary trees where every parent node is a hash of its children. In blockchains like Bitcoin, each block comprises an MT such that the parents are hashes of transactions that are included in the block. More concretely, the value of a parent node a is the hash of the concatenation of its two children nodes b, c, i.e. $a = \textsc{Hash}(b||c)$. The Merkle root root is the hash value of the root node of MT.

Proof-of-Work (PoW) [21]. In blockchains like Bitcoin [21], PoW is a protocol to propose new blocks. Here, miners use the blockchain's history $\mathcal{H}$ (comprising previously mined blocks, say up till B_{k-1}) and root of the set of transactions to be included in their block, B_k. The block header of B_k is made up of the hash of the parent block B_{k-1}, root, and a randomly generated nonce. The block is considered mined if the miner finds a nonce such that the hash value of the block $h = \textsc{Hash}(B_k)$ is lesser than *target difficulty* (TD) as decided by the system, i.e., $h < TD$.

On-chain Trustless Randomness. Micali et al. [19] introduce *verifiable random functions*, which take inputs and generate pseudorandom outputs that can be publicly verified. In the blockchain context, this often implies functions whose randomness depends on the information available to the blockchain (aka verifiable or trustless on-chain randomness). E.g., Chung and Shi [7] propose a randomized second-price TFM that uses such randomness to confirm transactions added to its block by the miner.

4 Fairness in TFMs

This section (i) presents our novel fairness notions, (ii) proves the impossibility of simultaneously maximizing the miner's utility and ZTi, (iii) studies the fairness guarantees of BitcoinF when $\delta = 0$, and (iv) discusses Softmax TFM (`STFM`).

4.1 Fairness Notions

We propose the following fairness notions to tackle the challenges due to transaction fees in TFMs.

① **Zero-fee Transaction Inclusion (ZTi).** In Bitcoin, a TFM requires a user to pay transaction fees, even for micropayments. Furthermore, there is an unbounded waiting time for transactions with marginal fees in Bitcoin [25]. As such, we introduce *Zero-fee Transaction Inclusion* (ZTi) as a critical fairness notion for a TFM to satisfy. That is, our first fairness notion ensures that a transaction with zero fees must have a non-zero probability of getting included in the block.

Definition 4 (Zero-fee Transaction Inclusion (ZTi))**.** *A TFM $\mathcal{T}^{TFM}$ satisfies ZTi if the probability with which a transaction t with transaction fee $b_t = 0$ gets included in a block B_k is strictly non-zero, i.e.,* $\Pr(t \in B_k) > 0$.

As the users and miners are myopic, ZTi only considers a transaction's probability of being included in the next block.

② **Monotonicity.** This notion focuses on the probability of the inclusion of a bidding user's transaction being proportional to the transaction fee. Naturally, a user would expect a higher probability of its transaction being included if it increases the transaction's fee. Such a scenario is also desirable in practice, e.g., startups/applications may want faster transaction acceptance to meet launch dates, deployment targets, or critical bug fixes.

Definition 5 (Monotonicity)**.** *A TFM $\mathcal{T}^{TFM}$ satisfies Monotonicity if the probability with which a transaction t gets accepted in a block B_k increases with an increase in its transaction fee b_t, given the remaining bids $\mathbf{b}_{-t}$ are fixed. That is,* $\Pr(t \in B_k \mid \mathbf{b}_{-t}, b_t + \epsilon) > \Pr(t \in B_k \mid \mathbf{b}_{-t}, b_t)$ *for any $\epsilon > 0$ and fixed $\mathbf{b}_{-t}$.*

We remark that most existing TFMs satisfy monotonicity. However, designing TFMs that satisfy monotonicity and ZTi simultaneously is non-trivial. Trivially, a TFM satisfying both our fairness notions ensures that each transaction has a non-zero probability of getting accepted!

4.2 Impossibility of Simultaneously Maximizing Miner Utility and Satisfying ZTi

Before presenting the main impossibility, we first analyze the fairness guarantees for EIP-1559 [5].

Remark 2. EIP-1559 satisfies (i) Monotonicity but does not satisfy (ii) ZTi. As each transaction must at least pay λ, no honest/strategic miner will include zero-fee transactions to preserve the validity of their blocks, i.e., if $b_t = 0 \implies \Pr(t \in B) = 0$. EIP-1559 satisfies Monotonicity as increasing the payment $b_t - \lambda$ increases the chance of the transaction being part of the optimal set in Eq. 3.

Theorem 1 adds to Remark 2 by showing that any TFM that allows a strategic miner complete control over which transactions to add cannot satisfy ZTi, for any non-trivial payment rule. A *trivial payment rule* is $p_t = 0, \forall t \in B_k$. For the proof, in the full version [8], we provide a counterexample s.t. $\forall t \in M,\ b_t = 0 \implies \Pr(t \in B_k = 0)$.

Theorem 1. *No $\mathcal{T}^{TFM}$ with a non-trivial payment rule which provides a strategic miner complete control over the transactions to add to its block, satisfies Zero-fee Transaction Inclusion (ZTi).*

4.3 BitcoinZF: BitcoinF with Zero Fees

We tweak the block allocation rule in BitcoinF [25] to introduce a provision for transactions with zero fees. We set $\delta = 0$ so that the miner *randomly* adds zero-fee transactions to fill the $1-\alpha$ section, followed by *greedily* adding transactions with bid b to the α section. The formal optimization can be derived by fixing $\delta = 0$ in Eq. 4.

Furthermore, with base fee λ, for each i in the α section we have $p_i^{BZ} = b_i - \lambda$ and $q_i^{BZ} = \lambda$. For each i in the $1-\alpha$ section we have $p_i^{BZ} = q_i^{BZ} = 0$. In summary, BitcoinZF is denoted by the tuple $\mathcal{T}^{BZ} = (\mathbf{x}^{BZ}, \mathbf{p}^{BZ}, \mathbf{q}^{BZ}, \tau_D)$.

Fairness Notions. Theorem 2 shows that BitcoinZF satisfies the two fairness notions if each zero-fee transaction's size is less than $C_{1-\alpha}$. In other words, BitcoinZF satisfies ZTi if none of the zero-fee transactions are of significant size.

Theorem 2. *BitcoinZF ($\mathcal{T}^{BZ}$) satisfies (i) Zero-fee Transaction Inclusion and (ii) Monotonicity only if $\forall\, t_i \in M$ with $b_i = 0$, we have $s_i \leq C_{1-\alpha}$.*

We defer Theorem 2's proof to the full version [8]. Informally, let a user i increase its b_i. At the same time, if the other bids remain unchanged, user i's chances of being included in the "α" section increase, satisfying Monotonicity. Furthermore, since the miner receives no increase in utility from any transaction in the "$1-\alpha$" section, it can uniformly include zero-fee transactions.

Cost of Fairness (CoF). Unfortunately, there is a "cost" to the fairness guarantees in BitcoinZF. Ensuring ZTi *hurts* the miner's utility. To this end, consider the following definition.

Definition 6 (CoF). *We define (CoF) of $\mathcal{T}^{TFM} = (\mathbf{x}, \mathbf{p}, \mathbf{q}, \tau)$ as CoF$_{TFM}$ $= \max_{\mathbf{b}\neq 0} \frac{OPT}{u_{\mathsf{M}}^{TFM}}$. Here, u_{M}^{TFM} is the miner's utility from the intented allocation $\mathbf{x}$ and $\mathtt{OPT}$ its utility from Eq. 3 with $p_t = b_t$ and $q_t = 0$, $\forall t \in B_k$.*

Trivially, *lesser* the CoF, *greater* the miner's utility from following $\mathcal{T}^{TFM}$. Claim 1 presents the CoF for BitcoinZF for the specific case when for every $t_i, t_j \in M$ s.t. $i \neq j$, we have $s_i = s_j$. That is, all transactions are of the same size. The proof follows from algebraic manipulations.

Claim 1. *For every $t_i, t_j \in M$ s.t. $i \neq j$, if we have $s_i = s_j$, then CoF$_{BZ}$ $= \frac{OPT}{u_{\mathsf{M}}^{BZ}} = 1/\alpha$ where $\alpha \in (0, 1]$.*

Proof. W.l.o.g., let the optimal set of bids (sorted in non-decreasing order) which maximizes the miner's utility in Eq. 3 with $p_t = b_t$ and $q_t = 0$, $\forall t$ be $\{b_1, \ldots, b_c\}$. Then with $\alpha = \frac{k}{c}$ s.t. $k \leq c$, we can write BitcoinZF's bid set as $\{b_1, \ldots, b_k\}$ (since the miner will maximize utility in the "α" section of the block). Observe that,

$$\frac{OPT}{u_m^{BZ}} = \frac{b_1 + \ldots + b_c}{b_1 + \ldots + b_k} = 1 + \frac{b_{c-k+1} + \ldots + b_c}{b_1 + \ldots + b_k}$$
$$\leq 1 + \frac{(c-k)b_k}{k \cdot b_k} \leq 1 + \frac{c}{k} - 1 \leq \frac{c}{k} = 1/\alpha.$$

This completes the claim. □

Challenges with BitcoinZF. Despite satisfying our fairness notions, BitcoinZF has the following challenges. First, Claim 1 only holds when each transaction's size is equal. With different transaction sizes, $\frac{\mathtt{OPT}}{u_{\mathsf{M}}^{BZ}}$ can be arbitrarily bad. E.g., if the size of the transaction with the highest bid in M is greater than C_α, $\mathtt{OPT}/u_{\mathsf{M}}^{BZ} \to \infty$. Second, when $1-\alpha$ is small, zero-fee transactions of sufficient size will deterministically never get included in the block. Formally, if $\exists\, t_i \in M$ s.t. $b_i = 0$ and $s_i > C_{1-\alpha}$, we have $\Pr(t_i \in B_k) = 0$.

Algorithm 1 Softmax TFM (STFM) Allocation

```
Input: Block Size $C$, Mempool $M$, History $\mathcal{H}$, Temperature $\gamma$
Output: Set of allocated transactions in $B_k$, i.e., $\mathcal{X}_k$
procedure STFMAllocation($C, M, \mathcal{H}$)
    $S = 0, \mathcal{X}_k = \emptyset$
    $\Gamma_k = \left[\frac{\exp(b_t/\gamma)}{\sum_{t' \in M} \exp(b_{t'}/\gamma)}\right]_{\forall t \in M}$    ▷ Generate the Softmax distribution
    while $C - S > 0$ do
        $t \sim \Gamma_k$    ▷ Sample a transaction
        $S \leftarrow S + s_t$    ▷ Add to the current block consumption
        $\mathcal{X}_k \leftarrow \mathcal{X}_k + \{t\}$
        $\Gamma_k = \left[\frac{\exp(b_t/\gamma)}{\sum_{t' \in M \setminus \mathcal{X}_k} \exp(b_{t'}/\gamma)}\right]_{\forall t \in M \setminus \mathcal{X}_k}$    ▷ Re-generate the Softmax distribution
    end while
    return $\mathcal{X}_k$
end procedure
```

4.4 STFM: First Approach to Achieve Fairness Through Randomization

A straightforward approach to satisfy our fairness notions is through an allocation rule wherein a miner samples transactions from a distribution generated by applying the *softmax* function [4] to the set of outstanding transactions in the mempool. The exponential function used to generate the softmax distribution trivially gives a non-zero probability of inclusion for zero-fee transactions and also retains monotonicity. Algorithm 1 formally describes STFM allocation rule.

Unfortunately, STFM does not satisfy MIC, as a strategic miner can always maximize its revenue by optimally selecting transactions instead of following STFM's randomized allocation. We provide the formal mechanism and other results for STFM in the full version [8].

5 rTFM: Fairness in Transaction Fees Mechanism using Randomization

We now propose rTFM: a TFM that uses trustless on-chain randomness to guarantee both our fairness constraints, namely (i) *ZTi* (Zero-Fee Transaction inclusion) and (ii) *Monotonicity*. In addition to this, rTFM is both Miner Incentive Compatible (MIC) and User Incentive Compatible (UIC).

We next (i) formally introduce rTFM and (ii) show that – for appropriate payment rules – rTFM preserves desired incentive guarantees while simultaneously satisfying Monotonicity and ZTi.

5.1 rTFM: Randomized TFM

We denote rTFM as the tuple $\mathcal{T}_\phi^{\text{rTFM}} = \left(\mathbf{x}_\phi^{\text{rTFM}}, \mathbf{p}, \mathbf{q}, \tau_{\mathbf{R}}\right)$. At its core, rTFM comprises a novel allocation rule, $\mathbf{x}^{\text{rTFM}}$, and can be paired with any payment and burning rule. The allocation rule uses two sub-procedures: (i) *transaction sampling* and (ii) *biased coin-toss*. We first introduce these procedures and subsequently use them to formally define $\mathbf{x}_\phi^{\text{rTFM}}$.

Transaction Sampling. An honest miner of a block adds transactions from the mempool M to its block using the following rules.

- Rule 1: The miner uniformly adds transactions from the mempool M to its block B_k. But, for each transaction $t \in B_k$, the miner receives zero fees. That is, $\forall t \in B_k, p_t = 0$. Denote the Merkle tree constructed using these transactions as $\mathsf{MT}_{\mathtt{rand}}$ with the Merkle root, $\mathsf{root}_{\mathtt{rand}}$.

- <u>RULE 2</u>: The miner selects the transactions optimally, i.e., using Eq. 3. Denote the Merkle tree constructed using these transactions according to $\mathsf{MT}_{\texttt{opt}}$ with the Merkle root, $\mathsf{root}_{\texttt{opt}}$.

While mining a block, the miner selects transactions and constructs Merkle trees according to Rule 1 and Rule 2. Let the transaction selection rule, given M, be represented as $\textsc{sample}(M) := ((\mathsf{root}_{\texttt{rand}}, \mathsf{MT}_{\texttt{rand}}), (\mathsf{root}_{\texttt{opt}}, \mathsf{MT}_{\texttt{opt}}))$.

Trustless Biased Coin Toss. `rTFM`'s allocation rule selects one out of the two sets of transactions created from Rules 1 and 2. To select between the two sets, we now introduce an *on-chain-based* biased coin toss method. Let $\phi \in [0, 1]$ denote the probability of heads (or 0) and $1 - \phi$ denote the probability of tails (or 1).

From Section 3, a miner mines its block B_k at height k using the hash of the parent block $\textsc{Hash}(B_{k-1})$, the random nonce `rand`, the block height k, and the two Merkle roots $\mathsf{root}_{\texttt{rand}}$ and $\mathsf{root}_{\texttt{opt}}$. If the block is mined, i.e., $\textsc{Hash}(B_k) < TD$ for target difficulty TD, then the toss' outcome is considered as follows:

$$O\left(\textsc{Hash}(B_k), \phi\right) := \begin{cases} 0 & \text{if } \textsc{Hash}(B_k) < \phi \cdot TD \\ 1 & \text{otherwise} \end{cases} \quad (5)$$

Remark 3 shows that Eq. 5 is equivalent to a biased coin toss; refer to the full version [8] for the formal proof.

Remark 3. *Invoking $O(\textsc{Hash}(B_k), \phi)$ for a mined block B_k is equivalent to a biased coin toss with ϕ as the probability of heads.*

Given this, Algorithm 2 provides the procedural outline of $\mathbf{x}_\phi^{\text{rTFM}}$. The procedure is summarized as follows:

- <u>STEP 1.</u> Miner samples two Merkle trees $\mathsf{MT}_{\texttt{rand}}$ and $\mathsf{MT}_{\texttt{opt}}$ by invoking $\textsc{sample}(M)$ and includes both Merkle roots $\mathsf{root}_{\texttt{rand}}$ and $\mathsf{root}_{\texttt{opt}}$ in block header B_k.
- <u>STEP 2.</u> Miner selects a random nonce for the block header B_k until the block is mined; i.e. $\textsc{Hash}(B_k) < TD$.
- <u>STEP 3.</u> Miner invokes biased coin toss $O(\textsc{Hash}(B_k), \phi)$ (Eq. 5). If the outcome is 1, then $\mathsf{MT}_{\texttt{opt}}$ (optimally selected transactions) is considered part of the blockchain. If the outcome is 0, then $\mathsf{MT}_{\texttt{rand}}$ (Merkle tree with uniformly sampled transactions) is considered part of the blockchain.

To summarize, Definition 7 formally defines $\mathbf{x}^{\text{rTFM}}$.

Definition 7 (`rTFM` Allocation Rule)**.** *Given $\mathcal{H}, M$ and B_k, let x_ϕ^{rTFM} denote a feasible allocation rule generated using Algorithm 2. Formally, the set of allocated transactions $x^{rTFM}(\mathcal{H}, M, B_k, C, \phi) = \mathcal{X}_k$ for block B_k is obtained from $(\mathcal{X}_k, B_k) \leftarrow \textsc{MineBlock}(C, M, p, \mathcal{H})$.*

rTFM Payment and Burning Rule. The allocation rule $\mathbf{x}_\phi^{\text{rTFM}}$ can be coupled with any payment $(\mathbf{p})$ and burning $(\mathbf{q})$ rules to define $\mathcal{T}_\phi^{\text{rTFM}}$. E.g., similar to FPA, we can create $\mathcal{T}_\phi^{\text{rTFM}}$ such that each bidding user i whose $t_i \in \mathcal{X}_k$ for $(\mathcal{X}_k, B_k) \leftarrow \textsc{MineBlock}(C, M, p, \mathcal{H})$ has $p_i^{FPA} = b_i$ else $p_i^{FPA} = 0$. In both cases, $q_i^{\text{rTFM}} = 0$.

5.2 *`rTFM`: Properties*

We now discuss the incentive properties of $\mathcal{T}_\phi^{\text{rTFM}}$ with payment rules of (i) First Price Auction (FPA) and (ii) EIP-1559. First, we show that `rTFM` satisfies our fairness properties, namely (1) *ZTi* and (2) *Monotonicity*. Following this, we also show that `rTFM` satisfies MIC for both FPA and EIP-1559 payment rules. Moreover, `rTFM` is UIC when the payment rule is EIP-1559. The formal proofs for the results presented in this subsection are available in the full version [8].

Algorithm 2 Randomized TFM (`rTFM`) Allocation Rule

```
Input: Block Size $C$, Mempool $M$, Zero-Fees probability $\phi$, parent Block $B_{k-1}$, Target difficulty $TD$
Output: $(\mathsf{MT}_k, B_k)$ Merkle Tree $\mathsf{MT}_k$ of selected transactions and Mined block $B_k$
procedure $\textsc{MineBlock}(C, M, \phi, B_{k-1})$
    $((\mathsf{root}_{\texttt{rand}}, \mathsf{MT}_{\texttt{rand}}), (\mathsf{root}_{\texttt{opt}}, \mathsf{MT}_{\texttt{opt}})) \leftarrow \textsc{sample}(M)$
    $r \leftarrow \textsc{random}()$                                    ▷ Select a random nonce
    $B_k \leftarrow (B_{k-1}, \mathsf{root}_{\texttt{rand}}, \mathsf{root}_{\texttt{opt}}, r)$          ▷ Construct block $B_k$
    while $\textsc{Hash}(B_k) \geq TD$ do
        $r \leftarrow \textsc{random}(\cdot)$
        $B_k \leftarrow (B_{k-1}, \mathsf{root}_{\texttt{rand}}, \mathsf{root}_{\texttt{opt}}, r)$
    end while
    if $\textsc{Hash}(B_k) \leq \phi \cdot TD$ then                            ▷ Biased coin-toss
        return $(\mathsf{MT}_{\texttt{rand}}, B_k)$
    else
        return $(\mathsf{MT}_{\texttt{opt}}, B_k)$
    end if
end procedure
```

`rTFM` with FPA. The payment rule for FPA for any selected transaction t_i with bid b_i is $p_i^{FPA} = b_i$ if $t_i \in B_k$ and $p_i^{FPA} = 0$ otherwise. In both cases, the burning rule is $q_i^{FPA} = 0$. Trivially, `rTFM` with FPA is not UIC, while (later) Theorem 5 proves that it satisfies MIC.

`rTFM` with EIP-1559. The EIP-1559 payment rule implies that for each bidding user i whose $t_i \in B_k$ and $b_i \neq 0$ has $p_i^{EIP-1559} = b_i - \lambda$ and $q_i^{EIP-1559} = \lambda$. Here, λ is the posted price determined by the network (refer to Footnote 9).

`rTFM`: Fairness Guarantees. TFM with `rTFM`'s allocation rule and EIP-1559's payment rule satisfies both Monotonicity and Zero-Fee Transaction Inclusion.

Theorem 3. *`rTFM` with EIP-1559 satisfies (i) Monotonicity and (ii) Zero-Fee Transaction Inclusion for any $\phi \in (0, 1)$.*

Note that, Theorem 1 does not apply to `rTFM` as the miner does not have complete control over which set of transactions are selected with $\mathbf{x}^{\text{rTFM}}$. We can trivially extend Theorem 3 to show that `rTFM` with FPA also satisfies both of our novel fairness notions.

`rTFM` : Incentive Properties. First, Theorem 4 shows that `rTFM` with EIP-1559 is UIC.

Theorem 4. *`rTFM` with EIP-1559's payment rule satisfies* Dominant Strategy Incentive Compatibility *(UIC), if λ is not excessively low.*

Theorem 4 follows by observing that a user's strategy does not depend on `rTFM`'s allocation but only on the payment and the burning rule. Thus, the UIC guarantee of EIP-1559 carries over for `rTFM` with EIP-1559.

Next, unlike `STFM` (Section 4.4), `rTFM` also satisfies MIC.

Theorem 5. *`rTFM` is Miner Incentive Compatible (MIC) when the allocation rule is $\mathbf{x}_\phi^{rTFM}$ and payment rule $\mathbf{p}^{rTFM}$ and burning rule $\mathbf{q}^{rTFM}$ are either (1) First Price Auction or, (2) EIP-1559.*

Proof. To show that the TFMs satisfy MIC, we remark that the selecting between the optimal and zero-fee transactions (refer Algorithm 2) is carried out by the blockchain in a trustless manner (Eq. 5). As the miner has no control over the random outcome of $O(\textsc{Hash}(B_k), \phi)$ (Remark 3), its strategy involves (i) optimally selecting the transactions and (ii) either adding the zero-fee transactions or keeping them empty. For (i), we know that both EIP-1559 and FPA payment rules satisfy MIC. For (ii), both strategies result in zero utility for the miner; that is, rTFM is MIC for the miner. □

To summarize, for appropriate payment and burning rules, rTFM satisfies MIC and our novel fairness notions (refer to Table 1).

Table 1: Summary of our results.

TFM	**UIC**	**MIC**	**Monotonicity**	**ZTi**
EIP-1559	✓*	✓	✓ (Rem. 2)	✗ (Rem. 2)
Uniform TFM	✗	✗	✗ [8]	✓ [8]
BitcoinZF	✗	✗‡ (Clm. 1)	✓ (Thm. 2)	✓† (Thm. 2)
STFM + FPA	✗ [8]	✗ [8]	✓ [8]	✓ [8]
STFM + EIP-1559	✓* [8]	✗ [8]	✓ [8]	✓ [8]
rTFM + FPA	✗	✓ (Thm. 5)	✓	✓
rTFM + EIP-1559	✓* (Thm. 4)	✓ (Thm. 5)	✓ (Thm. 3)	✓ (Thm. 3)

†: Only if $\forall\, t_i \in M$ with $b_i = 0$ we have $s_i \leq C_{1-\alpha}$.
⋆: Only if λ is not excessively low

5.3 *rTFM: Choosing ϕ*

rTFM's allocation rule is parameterized by the probability ϕ of mining a block where each transaction t_i has bid $b_i = 0$. We now discuss the impact of ϕ on CoF and the variation in the miner's revenue.

Cost of Fairness (CoF). From Definition 6, CoF is the ratio of the utilities u_{opt} (refer to Eq. 3) and u_{rTFM} (i.e., miner's utility when the transactions are selected according to $\mathbf{x}_\phi^{\text{rTFM}}$).

The miner's utility in rTFM is dependent on the output of random variable $O(\textsc{Hash}(B_k), \phi)$. If $O(\mathsf{Hash}(B_k), \phi) = 0$ (occurs with probability ϕ), then each selected transaction t_i has $b_i = 0$ resulting in zero revenue for the miner. In contrast, with probability $1 - \phi$, we have $O(\textsc{Hash}(B_k), \phi) = 1$, such that the optimal transactions are selected. Here, the miner's revenue is equal to u_{opt}. That is,

$$\mathbb{E}_\phi[u_{\text{rTFM}}] = \phi \cdot 0 + (1 - \phi) \cdot u_{\text{opt}}.$$

This implies that, $\mathsf{CoF}_{\text{rTFM}} = \frac{u_{\text{opt}}}{\mathbb{E}_\phi[u_{\text{rTFM}}]} = \frac{1}{1-\phi}$.

Impact of ϕ on CoF. Trivially, increasing ϕ increases ZTi. Doing so also increases CoF, reducing the miner's revenue. However, since rTFM (with an appropriate payment rule) is MIC, we believe that the system designers must choose an appropriate ϕ which (i) incentivizes the miner to not abstain from the system and (ii) allows for a desirable percentage of zero-fee transactions that may lead to greater adoption.

Coefficient of Variation (CoV). An increase in ϕ not only decreases the miner's expected revenue but will also *increase* its variance. More concretely, denote σ_{opt} as the standard deviation and π_{opt} as the miner's expected utility when it optimally selects the transactions. Likewise, σ_{rTFM} and π_{rTFM} are the standard deviation and expectation in the miner's utility from rTFM. We know that the Coefficient of Variation (CoV) is given by $\frac{\sigma}{\mu}$. By trivial arguments, we see:

$$CoV_{\text{opt}} = \frac{\sigma_{\text{opt}}}{u_{\text{opt}}} = 1 \;\&\; CoV_{\text{rTFM}} = \frac{\sigma_{\text{rTFM}}}{\mathbb{E}_\phi[u_{\text{rTFM}}]} = \left(\frac{1-\phi}{\phi}\right)^{1/2}$$

We want to choose ϕ such that $CoV^2_{\text{opt}}/CoV^2_{\text{rTFM}}$ is maximized. Observe that, as $\phi \to 0$, the CoV ratio increases monotonically.

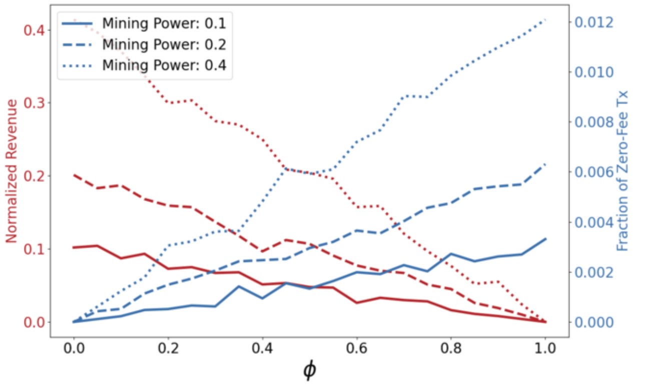
rTFM: Empirical Analysis

Setup. To simulate rTFM, we need to configure the size of the mempool M, block size C, ϕ, sample each user's bid b, and their sizes. In our experiments, we consider transactions of the same size ($s_i = s_j = 1$). We set the mempool size as $n = 1000$, block size $C = 100$, and each user's bid is sampled from *Truncated Gaussian* distribution, $b \sim \mathcal{N}(4, 3)$.

Measures. We vary $\phi \in [0, 1]$ and report the (i) *Normalized Miner Revenue*, ratio of miner's revenue from rTFM with OPT and (ii) *Fraction of Zero-fee Txs*, ratio of zero-fee transactions accepted in rTFM with the mempool size. The results reported are averaged across 1000 runs.

Results. Figure 1 depicts our results. As expected, an increase in ϕ increases the zero-fee transactions included and decreases the miner's revenue. In the full version [8], we also show that the trends depicted in Figure 1 remain the same when $b \sim \text{U}[0, 1]$ and $b \sim \text{Exp}(\lambda = 1.5)$.

Figure 1: rTFM: Effect of ϕ



Choosing ϕ. In summary, the trade-off between (1) CoF, (2) CoV, and (3) Fraction of Zero-fee Txs is such that as ϕ increases, CoF increases, CoV-ratio decreases, and ZTi increases. If we wish to increase the number of zero-fee transactions accepted, then we must compromise with utility and suffer higher variance. Figure 1 depicts the said trade-off empirically.

6 Conclusion

In this paper, we focused on the need for fairness in TFMs regarding the transaction fees for the transaction creators. We argued that including zero-fee transactions is necessary for the widespread adoption of TFMs. We introduced two novel fairness notions: Zero-fee Transaction Inclusion (ZTi) and Monotonicity. We showed that existing TFMs do not satisfy at least one of these notions or do so for smaller transaction sizes and at a high cost to the miner's utility. To resolve these limitations, we first introduced STFM which samples transactions through the distribution generated from the softmax with temperature (γ) function. We showed that while STFM is a fair TFM, it is not MIC. To this end, we introduced rTFM which simultaneously satisfies MIC and our fairness notions.

Future Work. We believe that these fair TFMs may further democratize TFMs by contributing to their broader accessibility and enhancing their adoption in the market. Future work can further study the role of ϕ in rTFM towards striking a desirable balance between a miner's revenue and the fraction of zero-fee transactions included. Last, as aforementioned, one can also explore extending rTFM for Proof-of-Stake blockchains.

References

[1] G. Aggarwal and J. D. Hartline. Knapsack auctions. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, page 1083–1092, 2006.

[2] A. Asayag, G. Cohen, I. Grayevsky, M. Leshkowitz, O. Rottenstreich, R. Tamari, and D. Yakira. A fair consensus protocol for transaction ordering. In *IEEE 26th International Conference on Network Protocols (ICNP)*, pages 55–65, 2018.

[3] S. Basu, D. Easley, M. O'Hara, and E. G. Sirer. Towards a functional fee market for cryptocurrencies. *arXiv preprint arXiv:1901.06830*, 2019.

[4] J. S. Bridle. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing*, pages 227–236. Springer, 1990.

[5] V. Buterin, E. Conner, R. Dudley, M. Slipper, I. Norden, and A. Bakhta. Eip-1559: Fee market change for eth 1.0 chain. eips.ethereum.org/EIPS/eip-1559, 2019.

[6] V. Buterin et al. A next-generation smart contract and decentralized application platform. *White Paper*, 3(37):2–1, 2014.

[7] H. Chung and E. Shi. Foundations of transaction fee mechanism design. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2023.

[8] S. Damle, V. Srivastava, and S. Gujar. No transaction fees? no problem! achieving fairness in transaction fee mechanism design. *arXiv preprint arXiv:2402.04634*, 2024.

[9] M. V. X. Ferreira, D. J. Moroz, D. C. Parkes, and M. Stern. Dynamic posted-price mechanisms for the blockchain transaction-fee market. In *ACM Conference on Advances in Financial Technologies (AFT)*, pages 86–99, 2021.

[10] A. Gervais, S. Capkun, G. O. Karame, and D. Gruber. On the privacy provisions of bloom filters in lightweight bitcoin clients. In *Annual Computer Security Applications Conference (ACSAC)*, pages 326–335, 2014.

[11] H. Gilbert and H. Handschuh. Security analysis of sha-256 and sisters. In M. Matsui and R. J. Zuccherato, editors, *Selected Areas in Cryptography*, pages 175–193, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. ISBN 978-3-540-24654-1.

[12] A. Jain, S. Siddiqui, and S. Gujar. We might walk together, but i run faster: Network fairness and scalability in blockchains. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, page 1539–1541, 2021.

[13] M. Kelkar, F. Zhang, S. Goldfeder, and A. Juels. Order-fairness for byzantine consensus. In *Annual International Cryptology Conference (CRYPTO)*, pages 451–480, 2020.

[14] K. Kursawe. Wendy, the good little fairness widget: Achieving order fairness for blockchains. In *ACM Conference on Advances in Financial Technologies (AFT)*, pages 25–36, 2020.

[15] A. Laurent, L. Brotcorne, and B. Fortz. Transactions fees optimization in the ethereum blockchain. *Blockchain: Research and Applications*, page 100074, 2022.

[16] Y. Mao and S. B. Venkatakrishnan. Less is more: Fairness in wide-area proof-of-work blockchain networks. *arXiv preprint arXiv:2204.02461*, 2022.

[17] R. C. Merkle. A digital signature based on a conventional encryption function. In *Conference on the theory and application of cryptographic techniques*, pages 369–378. Springer, 1987.

[18] J. Messias, M. Alzayat, B. Chandrasekaran, and K. P. Gummadi. On blockchain commit times: An analysis of how miners choose bitcoin transactions. In *The Second International Workshop on Smart Data for Blockchain and Distributed Ledger (SDBD2020)*, 2020.

[19] S. Micali, M. Rabin, and S. Vadhan. Verifiable random functions. In *40th annual symposium on foundations of computer science (cat. No. 99CB37039)*, pages 120–130. IEEE, 1999.

[20] D. Z. Morris. Bitcoin's unfinished business: Why micropayments still matter. https://www.coindesk.com/layer2/2022/04/28/bitcoins-unfinished-business-why-micropayments-still-matter/, 2022.

[21] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, page 21260, 2008.

[22] A. Orda and O. Rottenstreich. Enforcing fairness in blockchain transaction ordering. *Peer-to-peer Networking and Applications*, 14(6):3660–3673, 2021.

[23] T. Roughgarden. Transaction fee mechanism design. *CoRR*, abs/2106.01340, 2021.

[24] T. Roughgarden. Transaction fee mechanism design. In *ACM Conference on Economics and Computation (ACM EC)*, page 792, 2021.

[25] S. Siddiqui, G. Vanahalli, and S. Gujar. Bitcoinf: Achieving fairness for bitcoin in transaction fee only model. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 2008–2010, 2020.

[26] Y. Sokolik and O. Rottenstreich. Age-aware fairness in blockchain transaction ordering. In *IEEE/ACM 28th International Symposium on Quality of Service (IWQoS)*, pages 1–9. IEEE, 2020.

[27] E. Tedeschi, T.-A. S. Nordmo, D. Johansen, and H. D. Johansen. On optimizing transaction fees in bitcoin using ai: Investigation on miners inclusion pattern. *ACM Trans. Internet Technol.*, 22(3), 2022.

[28] "Visa and Mastercard". Visa and mastercard are losing fast to indian alternatives. https://d3.harvard.edu/platform-digit/submission/visa-and-mastercard-are-losing-fast-to-indian-alternatives/, 2020.

[29] B. Wikipedia. Historic rules for free transactions. https://en.bitcoin.it/wiki/Miner_fees, 2022.

[30] Z. Zhao, X. Chen, and Y. Zhou. Bayesian-nash-incentive-compatible mechanism for blockchain transaction fee allocation. In *Crypto Economics Security Conference (CESC)*, 2022.