Transaction Fee Mechanism Design in a Post-MEV World

Maryam Bahrani ✉
Ritual, New York, NY, USA

Pranav Garimidi ✉
a16z Crypto, New York, NY, USA

Tim Roughgarden ✉
a16z Crypto , New York, NY, USA
Columbia University, New York, NY, USA

Abstract

The incentive-compatibility properties of blockchain transaction fee mechanisms have been investigated with passive block producers that are motivated purely by the net rewards earned at the consensus layer. This paper introduces a model of active block producers that have their own private valuations for blocks (representing, for example, additional value derived from the application layer). The block producer surplus in our model can be interpreted as one of the more common colloquial meanings of the phrase "maximal extractable value (MEV)."

We first prove that transaction fee mechanism design is fundamentally more difficult with active block producers than with passive ones: With active block producers, no non-trivial or approximately welfare-maximizing transaction fee mechanism can be incentive-compatible for both users and block producers. These results can be interpreted as a mathematical justification for augmenting transaction fee mechanisms with additional components such as order flow auctions, block producer competition, trusted hardware, or cryptographic techniques.

We then consider a more fine-grained model of block production that more accurately reflects current practice, in which we distinguish the roles of "searchers" (who actively identify opportunities for value extraction from the application layer and compete for the right to take advantage of them) and "proposers" (who participate directly in the blockchain protocol and make the final choice of the published block). Searchers can effectively act as an "MEV oracle" for a transaction fee mechanism, thereby enlarging the design space. Here, we first consider a TFM that is inspired by how searchers have traditionally been incorporated into the block production process, with each transaction effectively sold off to a searcher through a first-price auction. We then explore the TFM design space with searchers more generally, and design a mechanism that circumvents our impossibility results for TFMs without searchers. Our mechanism (the "SAKA" mechanism) is incentive-compatible (for users, searchers, and the block producer), sybil-proof, and guarantees roughly 50% of the maximum-possible welfare when transaction sizes are small relative to block sizes. We conclude with a matching negative result: even when transaction sizes are small, no DSIC and sybil-proof deterministic TFM can guarantee more than 50% of the maximum-possible welfare.

2012 ACM Subject Classification Theory of computation → Computational pricing and auctions; Security and privacy → Distributed systems security

Keywords and phrases MEV, Transaction Fee Mechanisms, Auctions

Digital Object Identifier 10.4230/LIPIcs.AFT.2024.29

Related Version *Full Version*: `https://eprint.iacr.org/2024/331` [7]

Funding *Tim Roughgarden*: Research at Columbia University supported in part by NSF awards CCF-2006737 and CNS-2212745, and research awards from the Briger Family Digital Finance Lab and the Center for Digital Finance and Technologies.

6th Conference on Advances in Financial Technologies (AFT 2024).
Editors: Rainer Böhme and Lucianna Kiffer; Article No. 29; pp. 29:1–29:24
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

1.1 Transaction Fee Mechanisms for Allocating Blockspace

Blockchain protocols such as Bitcoin and Ethereum process transactions submitted by users, with each transaction advancing the "state" of the protocol (e.g., the set of Bitcoin UTXOs, or the state of the Ethereum Virtual Machine). Such protocols have finite processing power, so when demand for transaction processing exceeds the available supply, a strict subset of the submitted transactions must be chosen for processing. To encourage the selection of the "most valuable" transactions, the transactions chosen for processing are typically charged a transaction fee. The component of a blockchain protocol responsible for choosing the transactions to process and what to charge for them is called its *transaction fee mechanism (TFM)*.

Previous academic work on TFM design (surveyed in Section 1.5) has focused on the game-theoretic properties of different designs, such as incentive-compatibility from the perspective of users (ideally, with a user motivated to bid its true value for the execution of its transaction), of block producers (ideally, with a block producer motivated to select transactions to process as suggested by the TFM), and of cartels of users and/or block producers. Discussing incentive-compatibility requires defining utility functions for the relevant participants. In most previous works on TFM design (and in this paper), users are modeled as having a private value for transaction inclusion and a quasi-linear utility function (i.e., value enjoyed minus price paid). In previous work – and, crucially, unlike in this work – a block producer was modeled as *passive*, meaning its utility function was the net reward earned (canonically, the unburned portion of the transaction fees paid by users, possibly plus a block reward).

While this model is a natural one for the initial investigation of the basic properties of TFMs, it effectively assumes that block producers are unaware of or unconcerned with the semantics of the transactions that they process – that there is a clean separation between users (who have value only for activity at the application layer) and block producers (who, if passive, care only about payments received at the consensus layer).

1.2 MEV and Active Block Producers

It is now commonly accepted that, at least for blockchain protocols that support a decentralized finance ("DeFi") ecosystem, there are unavoidable interactions between the consensus layer (block producers) and the application layer (users), and specifically with block producers deriving value from the application layer that depends on which transactions they choose to process (and in which order). For a canonical example, consider a transaction that executes a trade on an automated market maker (AMM), exchanging one type of token for another (e.g., USDC for ETH). The spot price of a typical AMM moves with every trade, so by executing such a transaction, a block producer may move the AMM's spot price out of line with the external market (e.g., on centralized exchanges (CEXs) like Coinbase), thereby opening up an arbitrage opportunity (e.g., buying ETH on a CEX at the going market price and then selling it on an AMM with a larger spot price). The block producer is uniquely positioned to capture this arbitrage opportunity, by executing its own "backrunning" transaction (i.e., a trade in the opposite direction) immediately after the submitted trade transaction.

The first goal of this paper is to generalize the existing models of TFM design in the minimal way that accommodates *active* block producers, meaning block producers with a utility function that depends on both the transactions in a block and the net fees earned.

Specifically, in addition to the standard private valuations for transaction inclusion possessed by users, the block producer will have its own private valuation, which is an abstract function of the block that it publishes. We then assume that a block producer acts to maximize its *block producer surplus (BPS)*, meaning its private value for the published block plus any additional profits (or losses) from fees (or burns). In the interests of a simple but general model, we deliberately avoid microfounding the private valuation function of a block producer or committing to any specifics of the application layer. Our model captures, in particular, canonical on-chain DeFi opportunities such as arbitrage and liquidation opportunities, but a block producer's valuation can reflect arbitrary preferences, perhaps derived also from off-chain activities (e.g., a bet with a friend that settles on-chain) or subjective considerations.

The extraction of application-layer value by block producers, in DeFi and more generally, was first studied by Daian *et al.* [17] under the name "MEV" (for "maximal extractable value"). At this point, the term has transcended any specific definition – in both the literature and popular discourse, it is used, often informally, to refer to a number of related but different concepts. For a brief survey see Section 1.5.4. We argue that our definition of BPS captures, in a precise way and in a concrete economic model, one of the more common colloquial meanings of the term "MEV."

1.3 The Block Production Supply-Chain

In the first part of this paper, we treat a block producer as a single entity that publishes a block based on the transactions that it is aware of. This would be an accurate model of block production, as carried out by miners in proof-of-work protocols and validators in proof-of-stake protocols, up until a few years ago. More recently, especially in the Ethereum ecosystem, block production has evolved into a more complex process, typically involving "searchers" (who identify opportunities for extraction from the application layer), "builders" (who assemble such opportunities into a valid block), "relays" (who gather blocks from builders and select the most profitable one for the proposer), and "proposers" (who participate directly in the blockchain protocol and make the final choice of the published block), and several others. One interpretation of a block producer in our model is as a vertically integrated party performing the job of all these entities.

In the second part of the paper, we consider a more fine-grained model of the block production process, in which the role of finding MEV extraction opportunities is decoupled from the proposer's role of participating in consensus and is instead performed by specialized searchers. An interpretation of this model is that the proposer runs an open-source consensus client to collect block rewards, while outsourcing the complicated task of finding MEV opportunies to searchers. This is in the same spirit as mev-geth, which was a widely-used Ethereum client written by Flashbots that proposers could run to allow for the submission of both regular transactions by users and wrapped bundles of transactions by searchers.[1] Prior to mev-geth, searchers and users were treated equally by proposers and competed with each other for inclusion; among other issues, multiple searchers pursuing the same MEV extraction opportunity would often have their extraction transactions included in a block, with the first such transaction capturing the opportunity and the rest failing (but still paying transaction fees for inclusion and wasting valuable blockspace). Mev-geth introduced an explicit auction, upstream from the blockchain's fee mechanism, in which searchers could compete directly with each other to capture MEV extraction opportunities. Our model can be viewed as formalizing this idea by allowing a TFM to treat searchers and users differently, subject to different rules for inclusion and payment.

[1] See https://github.com/flashbots/mev-geth/blob/master/README.md.

1.4 Overview of Results

Our starting point is the model for transaction fee mechanism design defined in [45]. In this model, each user has a private valuation for the inclusion of a transaction in a block, and submits a bid along with its transaction. As in [45], we consider TFMs that choose the included transactions and payments based solely on the bids of the pending transactions (as opposed to, say, based also on something derived from the semantics of those transactions). A block producer publishes any block that it wants, subject to feasibility (e.g., with the total size of the included transactions respecting some maximum block size). A TFM is said to be *dominant-strategy incentive-compatible (DSIC)* if every user has a dominant (i.e., always-optimal) bidding strategy. The DSIC property is often associated with a good "user experience (UX)," in the sense that each user has an obvious optimal bid. In [45], a TFM was said to be *incentive-compatible for myopic miners (MMIC)* if it expects a block producer to publish a block that maximizes the net fees earned (at the consensus layer). Here, we introduce an analogous definition that accommodates active block producers: We call a TFM *incentive-compatible for block producers (BPIC)* if it expects a block producer to publish a block that maximizes its private valuation plus the net fees earned. An ideal TFM would satisfy, among other properties, both DSIC and BPIC.

1.4.1 Vertically Integrated Active Block Producers

We begin with a model in which there are only users and a single (vertically integrated) active block producer, and show that there are fundamental barriers to designing ideal transaction fee mechanisms in this case.

Our first result (Theorem 11) is a proof that with active block producers *no* non-trivial TFM satisfies both DSIC and BPIC, where "non-trivial" means that users must at least in some cases pay a nonzero amount for transaction inclusion[2]. (In contrast, with passive block producers and no MEV, the "tipless mechanism" suggested in [45] is non-trivial and satisfies both DSIC and BPIC; the same is true of the EIP-1559 mechanism of Buterin et al. [12], provided the mechanism's base fee is not excessively low [45].) In particular, the EIP-1559 and tipless mechanisms fail to satisfy DSIC and BPIC when block producers can be active. Intuitively, for these mechanisms, a user might be motivated to underbid in the hopes of receiving an effective subsidy by the block producer (who may include the transaction anyways, if it derives outside value from it).

Our second result (Theorem 13) formalizes the intuition that TFMs that do not charge non-zero transaction fees – and in particular (by Theorem 11), TFMs that are both DSIC and BPIC – cannot guarantee any approximation of the maximum-possible social welfare. Intuitively, the issue is the lack of alignment between the preferences of users and of the block producer: If a block producer earns no transaction fees from any block, it might choose a block with non-zero private value but only very low-value transactions over one with no private value but very high-value transactions.

1.4.2 TFMs with Competitive Searchers

We then consider a more fine-grained model of block production that more accurately reflects current practice, where we distinguish the roles of "searchers" (who actively identify opportunities for value extraction from the application layer and compete for the right to

[2] We distinguish this result from surface level connections to previous impossibility theorems in mechanism design in Section 1.5.5

take advantage of them) and “proposers” (who participate directly in the blockchain protocol and make the final choice of the published block). Searchers can effectively act as an “MEV oracle” for a transaction fee mechanism, thereby enlarging the mechanism design space.

In this model, we first consider a TFM that is inspired by how searchers have traditionally been incorporated into the block production process, and specifically by mev-geth (see Section 2.5). Intuitively, this mechanism runs a first-price auction for each transaction among the interested searchers; the winning bid then acts as estimate of the transaction’s MEV, which the TFM can then use to charge prices to users in a way that recovers the DSIC property for users (Theorem 16).

We then explore the TFM design space with searchers more generally, with a focus on good approximate welfare guarantees. Our main contribution here is a mechanism, which we call the SAKA mechanism, which is DSIC for users, DSIC for searchers, BPIC, sybil-proof, and guarantees roughly 50% of the maximum-possible welfare when transaction sizes are small relative to block sizes (as they are in practice); see Theorems 19 and 20. In particular, this combination of guarantees shows that TFMs with searchers can evade impossibility results that apply to TFMs without searchers (such as Theorem 13). We further show in Theorem 21 that, even when transaction sizes are small, no DSIC and sybil-proof deterministic TFM can guarantee more than 50% of the maximum-possible welfare. (By “sybil-proof,” we mean that no user or searcher can ever profit from creating additional user or searcher identities and submitting fake transactions or bundles under those identities.)

1.5 Related Work

1.5.1 General TFM literature

The model in this paper is closest to the one used by Roughgarden [45] to analyze (with passive block producers) the economic properties of the EIP-1559 mechanism [12], the TFM used currently in the Ethereum blockchain. Precursors to that work (also with passive block producers) include studies of a “monopolistic price” transaction fee mechanism [32, 51] (also considered recently by Nisan [38]), and work of Basu et al. [10] that proposed a more sophisticated variant of that mechanism. There have also been several follow-up works to [45] that use similar models (again, with passive block producers). Chung and Shi [16] proved impossibility results showing that the incentive-compatible guarantees of the EIP-1559 mechanism are in some respects the best possible. There have also been attempts to circumvent this impossibility result by relaxing the notion of incentive compatibility [16, 24], using cryptography [47], considering a Bayesian setting [53], or mixtures of these ideas [49]. Other recent works [20, 33] study the dynamics of the base fee in the EIP-1559 mechanism.

1.5.2 MEV-aware mechanism design

There has been much interest among both researchers and practitioners in restructuring the block production supply chain to address MEV [50, 26]. On the academic side, the bulk of these approaches involve cryptographic techniques [29, 34, 52, 11] or changes at the consensus layer [28, 27, 13, 31]. Relatively recently, there have been some initial studies on the impact of economic mechanisms for mitigating MEV such as order-flow auctions [25] and mev-boost [2]; see [40, 43, 6]. In practice, to this point, economic approaches to addressing MEV have been more popular than cryptographic ones; examples include, among others, mev-share [36], UniswapX [3], and MEV Blocker [1]. The model in this paper aims to integrate some of the ideas behind these deployed applications into the existing mathematical frameworks for transaction fee mechanism design.

1.5.3 Credible mechanisms

Akbarpour and Li [4] introduce the notion of *credible* mechanisms, where any profitable deviations by the auctioneer can be detected by at least one user. While similar in spirit to the concept of BPIC introduced here (and the special case of MMIC introduced in [45]), there are several important differences. For example, the theory of credible mechanisms assumes fully private communication between bidders and the auctioneer and no communication among bidders, whereas TFM bids are commonly collected from a public mempool. Another difference is that a block producer in our model can manipulate only the allocation rule of a mechanism (as the payment rule is enforced by the blockchain protocol), while in the credible mechanisms framework the auctioneer can also manipulate the payment rule. In a different direction, there is also a line of follow-up work that takes advantage of cryptographic primitives to build credible auctions on the blockchain [22, 19, 15, 21].

1.5.4 Defining MEV

Daian et al. [17] introduced the notion of miner/maximal extractable value. They defined MEV as the value that miners or validators could obtain by manipulating the transactions in a block. Since this work, there have been many follow-up works attempting to formalize MEV and analyze its effects in both theory and practice. Attempts to give exact theoretical characterizations of MEV appear in [46, 39, 9, 5]. Broadly, these works define MEV by defining sets of valid transaction sequences and allowing the block producer to maximize their value over these sequences. These definitions are very general, but in exchange have to this point proved analytically intractable. Several empirical papers study the impact and magnitude of MEV using heuristics applied to on-chain data [41, 42, 48]. Another line of work [30, 26, 8] studies MEV in specific contexts, such as for arbitrage in AMMs, in which it is possible to characterize how much MEV can be realized from certain transactions. In particular, Kulkarni et al. [30] give formal statements on how, under different AMM designs, MEV affects the social welfare of the overall system.

1.5.5 Impossibility results in mechanism design

The impossibility results in Section 3 may appear superficially related to other such results in mechanism design. For example, the classic Myerson-Satterwhaite Theorem [37] states that there is no efficient, individually rational, Bayesian incentive compatible, and budget-balanced mechanism for bilateral trade. Fundamentally, this result is driven by the tension between welfare and budget-balance in the presence of incentive-compatibility constraints on the participants. Our main impossibility result (Theorem 11), meanwhile, is driven by the combination of incentive-compatibility constraints for users (analogous to the usual participants of a mechanism design problem) and also such a constraint for a self-interested party that is tasked with carrying out the allocation rule of the mechanism (the block producer). As such, our setup more closely resembles that of credible mechanisms than more traditional mechanism design settings. In particular, Theorem 11 holds even in the absence of any welfare-maximization or exact budget-balance requirements (a non-zero burning rule in the sense of Section 2.3 is tantamount to relaxing budget-balance).

2 Model

This section defines transaction fee mechanisms, the relevant players and their objectives, and the relevant incentive-compatibility notions. Sections 2.1–2.4 describe the basic model (with vertically integrated, active block producers) that is considered in Section 3, while Section 2.5 augments this model with searchers, which play a central role in Sections 4 and 5.

2.1 The Players and Their Objectives

2.1.1 Users

Users submit *transactions* to the blockchain protocol. The execution of a transaction updates the state of the protocol (e.g., users' account balances). The rules of the protocol specify whether a given transaction is *valid* (e.g., whether it is accompanied by the required cryptographic signatures). From now on, we assume that all transactions under consideration are valid. Every transaction t has a publicly known *size* s_t (e.g., the gas limit of an Ethereum transaction).

We assume that each user submits a single transaction t and has a nonnegative *valuation* v_t, denominated in a base currency like USD or ETH, for its execution in the next block. This valuation is *private*, in the sense that it is initially unknown to all other parties. We assume that the utility function of each user – the function that the user acts to maximize – is quasi-linear, meaning that its utility is either 0 (if its transaction is not included in the next block) or $v_t - p$ (if its transaction is included and it must pay a fee of p). We denote the set of transactions submitted to the TFM by T.

2.1.2 Blocks

A *block* is a finite set of transactions. A *feasible block* is a block that respects any additional constraints imposed by the protocol. For example, if the protocol specifies a maximum block size, then feasible blocks might be defined as those that comprise only valid transactions and also respect the block size limit.

2.1.3 Block producers (BPs)

We consider blockchain protocols for which the contents of each block are selected by a single entity, which we call the *block producer (BP)*. We focus on the decision-making of the BP that has been chosen at a particular moment in time (perhaps using a proof-of-work or proof-of-stake-based lottery) to produce the next block. We assume that whatever block the BP chooses is in fact published, with all the included transactions finalized and executed.

A BP chooses a block B from some abstract non-empty set $\mathcal{B}$ of feasible blocks, called its *blockset*. For example, the set $\mathcal{B}$ might consist of all the feasible blocks that comprise only transactions that the BP knows about (perhaps from a public mempool, or perhaps from private communications) along with transactions that the BP is in a position to create itself (e.g., a backrunning transaction). As with users, we model the preferences of a BP with a quasi-linear utility function, meaning the difference between its private value for a block (again, denominated in a base currency like USD or ETH) minus the (possibly negative) payment that it must make. Unlike with users, to avoid modeling any details of why a BP might value a block (e.g., due to the extraction of value from the application layer), we allow a BP to have essentially arbitrary preferences over blocks. More formally, we assume that a BP has a private valuation that is an arbitrary (real-valued) function v_{BP} over blocks, and the BP acts to maximize its *block producer surplus (BPS)*:

$$\underbrace{v_{BP}(B) + \text{net fees earned}}_{\text{block producer surplus (BPS)}}.$$

2.1.4 Holders

The final category of participants, which are non-strategic in our model but relevant for our definition of welfare in Section 2.2, are the holders of the blockchain protocol's native currency. As we'll see in Section 2.3, TFMs are in a position to mint or burn this currency, which corresponds to inflation or deflation, respectively. We treat TFM mints and burns as transfers from and to, respectively, the existing holders of this currency. Formally, we define the collective utility function of currency holders to be the net amount of currency burned by a TFM.

2.2 Welfare

According to the principle of welfare-maximization, a scarce resource like blockspace should be allocated to maximize the total utility of all the "relevant participants," which in our case includes the users, the BP, and the currency holders. Because all parties have quasi-linear utility functions and all TFM transfers will be between members of this group (from users to the BP, from the BP to holders, etc.), the welfare of a block is simply the sum of the user and BP valuations for it:

$$\underbrace{W(B) := v_{BP}(B) + \sum_{t \in B} v_t}_{\text{welfare of } B}. \tag{1}$$

Holders are assumed to be passive and thus have no valuations to contribute to the sum.[3]

2.3 Transaction Fee Mechanisms

The outcome of a transaction fee mechanism is a block to publish and a set of transfers (user payments, burns, etc.) that will be made upon the block's publication. In line with the preceding literature on TFMs and the currently deployed TFM designs, we assume that each user that creates a transaction t submits along with it a nonnegative *bid* b_t (i.e., willingness to pay), and that a TFM bases its transfers on the set of available transactions and the corresponding bids. (The BP submits nothing to the TFM.) A TFM is defined primarily by its *payment* and *burning* rules, which specify the fees paid by users and the burned funds implicitly received by holders (with the BP pocketing the difference).

2.3.1 Payment and burning rules

The payment rule specifies the payments made by users in exchange for transaction inclusion.

▶ **Definition 1** (Payment Rule). *A* payment rule *is a function* $\mathbf{p}$ *that specifies a nonnegative payment* $p_t(B, \mathbf{b})$ *for each transaction* $t \in B$ *in a block* B*, given the bids* $\mathbf{b}$ *of all known transactions.*

[3] We stress that the welfare of a block (1) measures the "size of the pie" and says nothing about how this welfare might be split between users, the BP, and holders (i.e., about the size of each slice). Distributional considerations are important, of course, but they are outside the scope of this paper.

The value of $p_t(B, \mathbf{b})$ indicates the payment from the creator of an included transaction $t \in B$ to the BP that published that block. (Or, if the rule is randomized, the expected payment.[4]) We consider only *individually rational* payment rules, meaning that $p_t(B, \mathbf{b}) \leq b_t$ for every included transaction $t \in B$. We can interpret $p_t(B, \mathbf{b})$ as 0 whenever $t \notin B$. Finally, we assume that every creator of an included transaction has the funds available to pay its full bid, if necessary (otherwise, the block B should be considered infeasible).

The burning rule specifies how much money must be burned by a BP along with the publication of a given block.[5]

▶ **Definition 2** (Burning Rule). *A* burning rule *is a function q that specifies a nonnegative burn $q(B, \mathbf{b})$ for a block B, given the bids $\mathbf{b}$ of all known transactions.*

The value of $q_t(B, \mathbf{b})$ indicates the amount of money burned (i.e., paid to currency holders) by the BP upon publication of the block B. (Or, if the rule is randomized, the expected amount.)[6] We assume that, after receiving users' payments for the block, the BP has sufficient funds to pay the burn required of the block that it publishes (otherwise, the block B should be considered infeasible).

We stress that the payment and burning rules of a TFM are hard-wired into a blockchain protocol as part of its code. This is why their arguments – the transactions chosen for execution and their bids, and perhaps (as in [16]) the bids of some additional, not-to-be-executed transactions – must be publicly recorded as part of the blockchain's history. (E.g., late arrivals should be able to reconstruct users' balances, including any payments dictated by a TFM, from this history.) A BP cannot manipulate the payment and burning rules of a TFM, except inasmuch as it can choose which block $B \in \mathcal{B}$ to publish.

2.3.2 Allocation rules

In our model, a BP has unilateral control over the block that it chooses to publish. Thus, a TFM's allocation rule – which specifies the block that should be published, given all of the relevant information – can only be viewed as a recommendation to a BP. Because the (suggested) allocation rule would be carried out by the BP and not by the TFM directly, it can sensibly depend on arguments not known to the TFM (but known to the BP), specifically the BP's valuation v_{BP} and blockset $\mathcal{B}$.

▶ **Definition 3** (Allocation Rule). *An* allocation rule *is a function $\mathbf{x}$ that specifies a block $\mathbf{x}(\mathbf{b}, v_{BP}, \mathcal{B}) \in \mathcal{B}$, given the bids $\mathbf{b}$ of all known transactions, the BP valuation v_{BP}, and the BP blockset $\mathcal{B}$.*

An allocation rule $\mathbf{x}$ induces per-transaction allocation rules with, for a transaction t, $x_t(\mathbf{b}, v_{BP}, \mathcal{B}) = 1$ if $t \in \mathbf{x}(\mathbf{b}, v_{BP}, \mathcal{B})$ and 0 otherwise.

▶ **Definition 4** (Transaction Fee Mechanism (TFM)). *A* transaction fee mechanism (TFM) *is a triple $(\mathbf{x}, \mathbf{p}, q)$ in which $\mathbf{x}$ is a (suggested) allocation rule, $\mathbf{p}$ is a payment rule, and q is a burning rule.*

[4] We assume that users and BPs are risk-neutral when interacting with a randomized TFM.

[5] This differs superficially from the formalism in [45], in which a burning rule specifies per-transaction (rather than per-block) transfers from users (rather than the BP) to currency holders. The payment rule here can be interpreted as the sum of the payment and burning rules in [45], and the per-block burning rule here can be interpreted as the sum of the burns of a block's transactions in [45].

[6] An alternative to money-burning that has similar game-theoretic and welfare properties is to transfer $q(B, \mathbf{b})$ to entities other than the BP, such as a foundation or the producers of future blocks.

A TFM is defined relative to a specific block publishing opportunity. A blockchain protocol is free to use different TFMs for different blocks (e.g., with different base fees), perhaps informed by the blockchain's past history.

2.3.3 Utility functions and BPS revisited

With Definitions 1–4 in place, we can express more precisely the strategy spaces and utility functions introduced in Section 2.1. We begin with an expression for the utility of a user (as a function of its bid) for a TFM's outcome, under the assumption that the BP always chooses the block suggested by the TFM's allocation rule.

▶ **Definition 5** (User Utility Function). *For a TFM* $(\mathbf{x}, \mathbf{p}, q)$*, BP valuation* v_{BP}*, BP blockset* $\mathcal{B}$*, and bids* $\mathbf{b}_{-t}$ *of other transactions, the utility of the originator of a transaction* t *with valuation* v_t *and bid* b_t *is*

$$u_t(b_t) := v_t \cdot x_t((b_t, \mathbf{b}_{-t}), v_{BP}, \mathcal{B}) - p_t(B, (b_t, \mathbf{b}_{-t})), \tag{2}$$

where $B := x_t((b_t, \mathbf{b}_{-t}), v_{BP}, \mathcal{B})$*.*

In (2), we highlight the dependence of the utility function on the argument that is directly under a user's control, the bid b_t submitted with its transaction.

The BP's utility function, the block producer surplus, is then:

▶ **Definition 6** (Block Producer Surplus (BPS)). *For a TFM* $(\mathbf{x}, \mathbf{p}, q)$*, BP valuation* v_{BP}*, BP blockset* $\mathcal{B}$*, and transaction bids* $\mathbf{b}$*, the block producer surplus of a BP that chooses the block* $B \in \mathcal{B}$ *is*

$$u_{BP}(B) := v_{BP}(B) + \sum_{t \in B} p_t(B, \mathbf{b}) - q(B, \mathbf{b}). \tag{3}$$

In (3), we highlight the dependence of the BP's utility function on the argument that is under its direct control, its choice of a block. The BP's utility depends on the payment and burning rules of the TFM, but not on its allocation rule (which the BP is free to ignore, if desired).

Finally, the collective utility function of (passive) currency holders for a block B with transaction bids $\mathbf{b}$ is $q(B, \mathbf{b})$, the amount of currency burned by the BP. (As promised, for a block B, no matter what the bids and the TFM, the sum of the utilities of users, the BP, and holders is exactly the welfare defined in (1).)

2.4 Incentive-Compatible TFMs

In this paper, we focus on two incentive-compatibility notions for TFMs – which, as we'll see, are already largely incompatible – one for users and one for block producers. We begin with the latter.

2.4.1 BPIC TFMs

We assume that a BP will choose a block to maximize its utility function, the BPS (Definition 6). The defining equation (3) shows that, once the payment and burning rules of a TFM are fixed, a BP's valuation and blockset dictate the unique (up to tie-breaking) BPS-maximizing block for each bid vector. We call an allocation rule *consonant* if, given the payment and burning rules, it instructs a BP to always choose such a block (breaking ties in an arbitrary but consistent fashion). Because a BP can see all bids after they are submitted, they can also insert their own "fake" transactions along with "shill" bids for them (e.g., to manipulate the payment and/or burning rules of the TFM), we require that a BP is never incentivized to include such shill bids.

▶ **Definition 7** (Consonant Allocation Rule). *An allocation rule* $\mathbf{x}$ *is* consonant *with the payment and burning rules* $\mathbf{p}$ *and* q *if:*

(a) *for every BP valuation* v_{BP} *and blockset* $\mathcal{B}$*, and for every choice of transaction bids* $\mathbf{b}$*,*

$$\underbrace{\mathbf{x}(\mathbf{b}, v_{BP}, \mathcal{B})}_{\text{recommended block}} \in \underbrace{\operatorname*{argmax}_{B \in \mathcal{B}} \left\{ v_{BP}(B) + \sum_{t \in B} p_t(B, \mathbf{b}) - q(B, \mathbf{b}) \right\}}_{\text{BPS-maximizing block}};$$

(b) *for some fixed total ordering on the blocks of* $\mathcal{B}$*, the rule breaks ties between BPS-maximizing blocks according to this ordering.*

▶ **Definition 8** (Shill-Proof). *Payment and burning rules* $\mathbf{p}$ *and* q *are* shill-proof *if for every BP valuation* v_{BP} *and blockset* $\mathcal{B}$*, and for every choice of transaction bids* $\mathbf{b}$*, there is no set* F *of fake transactions with shill bids* $\mathbf{b}'$ *such that:*

$$\underbrace{\max_{B \in \mathcal{B}} \left\{ v_{BP}(B) + \sum_{t \in B \setminus F} p_t(B, (\mathbf{b}, \mathbf{b}')) - q(B, (\mathbf{b}, \mathbf{b}')) \right\}}_{\text{optimal BPS with shill bids}}$$
$$> \underbrace{\max_{B \in \mathcal{B}} \left\{ v_{BP}(B) + \sum_{t \in B} p_t(B, \mathbf{b}) - q(B, \mathbf{b}) \right\}}_{\text{optimal BPS without shill bids}}. \quad (4)$$

BPIC TFMs are then precisely those that always instruct a BP to choose a BPS-maximizing block (breaking ties consistently) while also being shill-proof.

▶ **Definition 9** (Incentive-Compatibility for Block Producers (BPIC)). *A TFM* $(\mathbf{x}, \mathbf{p}, q)$ *is* incentive-compatible for block producers (BPIC) *if:*

(a) $\mathbf{x}$ *is consonant with* $\mathbf{p}$ *and* q*;*

(b) $\mathbf{p}$ *and* q *are shill-proof.*

2.4.1.1 DSIC TFMs

Dominant-strategy incentive-compatibility (DSIC) is one way to formalize the idea of a "good user experience (UX)" for TFMs. The condition asserts that every user has an "obviously optimal" bid, meaning a bid that, provided the BP follows the TFM's allocation rule, is guaranteed to maximize the user's utility (no matter what other users might be bidding). In the next definition, by a *bidding strategy*, we mean a function σ that maps a valuation to a recommended bid for a user with that valuation.

▶ **Definition 10** (Dominant-Strategy Incentive-Compatibility (DSIC)). *A TFM* $(\mathbf{x}, \mathbf{p}, q)$ *is* dominant-strategy incentive-compatible (DSIC) *if there is a bidding strategy* σ *such that, for every BP valuation* v_{BP} *and blockset* $\mathcal{B}$*, every user* i *with transaction* t*, every valuation* v_t *for* i*, and every choice of other users' bids* $\mathbf{b}_{-t}$*,*

$$\underbrace{\sigma(v_t)}_{\text{recommended bid}} \in \underbrace{\operatorname*{argmax}_{b_t} \{u_t(b_t)\}}_{\text{utility-maximizing bid}}, \quad (5)$$

where u_t *is defined as in* (2)*.*

That is, bidding according to the recommendation of the bidding strategy σ is guaranteed to maximize a user's utility.[7] This is a strong property: a bidding strategy can depend only on what a user knows (i.e., its private valuation), while the right-hand side of (5) implicitly depends (through (2)) also on the bids of the other users and the BP's preferences.

Note that the classic EIP-1559 mechanism [12] is no longer BPIC for an active BP, even when the base fee is not excessively low (all the transactions with bid at least the base fee fit into the block). The concern is that an active BP is incentivized to include transactions that bid below the base fee (effectively subsidizing them) if the BP has sufficiently high value for those transactions. The main result of Section 3 (Theorem 11) shows that the difficulty of achieving DSIC and BPIC simultaneously is not particular to the EIP-1559 mechanism: When BPs are active, *no* TFM that charges non-zero user fees can be both DSIC and BPIC. In contrast, for a passive BP the DSIC and BPIC properties can be achieved simultaneously via the *tipless mechanism* [45].

In this work we do not focus on *offchain agreement proofness*, a third incentive-compatibility notion commonly studied in the context of transaction fee mechanisms. We note that our impossibility results (Theorems 11 and 13) apply already to mechanisms that are merely DSIC and BPIC (and not necessarily OCA-proof).

2.5 Adding Competitive Searchers

Next we describe the changes to the basic model that are needed in Sections 4 and 5, in which we suppose that block proposers outsource the problem of value extraction to searchers.

2.5.1 Searchers and bundles

Searchers submit bundles to the blockchain protocol, where a *bundle* consists of a single user-submitted transaction t and any additional transactions needed to extract value from the transaction. We interchange between referring to bundles by either w or t^i, with t^i explicitly referencing a bundle that includes transaction t. We assume that there is a canonical way to extend a transaction with size s_t into a bundle, and denote by s'_t the size of the latter (with $s'_t \geq s_t$). For example, if t represents an AMM trade, the corresponding canonical bundle might include a subsequent backrunning trade. Just as users submit bids with their transactions, searchers submit bids with their bundles. A TFM now takes as input both transactions (with their user bids) and bundles (with their searcher bids), and its allocation, payment, and burning rules can depend on the bids of all users and all searchers. We assume that a TFM can distinguish between transactions and bundles, and can therefore treat them differently (e.g., the payment rule can differ for users and for searchers). Like users, searchers have private nonnegative valuations for bundle inclusion and quasi-linear utility functions. The DSIC condition is defined for searchers exactly as it is for users (Definition 10).

[7] The term "DSIC" is often used to refer specifically to mechanisms that satisfy the condition in Definition 10 with the truthful bidding strategy, $\sigma(v_t) = v_t$. Any mechanism that is DSIC in the sense of Definition 10 can be transformed into one in which truthful bidding is a dominant strategy, simply by enclosing the mechanism in an outer wrapper that accepts truthful bids, applies the assumed bidding strategy σ to each, and passes on the results to the given DSIC mechanism. (This trick is known as the "Revelation Principle"; see e.g. [44].)

2.5.2 Blocks

Blocks can now include both transactions and bundles. Multiple searchers may submit bundles corresponding to the same transaction, but in a feasible block, a given transaction can be included (directly or as part of a bundle) at most once. The inclusion of a bundle that contains a transaction t necessarily implies the inclusion of t itself – in this sense, the space of feasible allocations is no longer downward-closed. Equivalently, a block now specifies a set of user-submitted transactions and, for each such transaction t, the searcher (if any) responsible for the included bundle that contains t. Users continue to have a private value v_t for inclusion (whether as part of a bundle or not).

2.5.3 Revised incentive-compatibility goals

Thus far, the addition of searchers strictly generalizes the model in Sections 1–4, and so our impossibility results (Theorems 11 and 13) for the basic model apply immediately to it as well.

But the whole point of accommodating a competitive ecosystem of searchers is for proposers (the entities that participate directly in the blockchain protocol) to outsource the specialized task of assembling high-value blocks to searchers. That is, searchers are meant to allow proposers to on the one hand act passively (by simply using the most valuable bundles submitted by searchers) and on the other hand earn almost all of the extractable value (with searchers competing the value of their bundles away to the proposer through the bidding process).[8] Mathematically, with searchers, the idea is that what had been the private valuation v_{BP} of the (vertically integrated) BP in Section 2.1 is now distributed specifically across the searchers. This interpretation is particularly clear in the additive case – meaning the vertically integrated BP valuation $v_{BP}(B)$ would have been $\sum_{t \in B} \mu_t$, with μ_t the value extractable from a transaction t and no interactions between different transactions – with every searcher that submits a bundle involving transaction t having a value of μ_t for that bundle.[9]

With this interpretation in mind, in the model with searchers, there will be three incentive-compatibility goals: (i) DSIC for users; (ii) DSIC for searchers; and (iii) BPIC for the proposer, assuming that the proposer is passive (i.e., with the all-zero valuation for blocks and with utility equal to the net revenue at the consensus layer, including any payments to it from searchers). In effect, this revised model shatters what had been a vertically integrated BP into a single proposer and a number of searchers, and what had been BPIC (with an active BP) now translates to DSIC for (active) searchers and BPIC for a passive proposer.[10]

[8] See [6] for a rigorous analysis of this idea.

[9] For example, transactions could represent trades on different AMMs, or once-per-block MEV opportunities such as top-of-block CEX-DEX arbitrage or liquidation opportunities (the latter two types modeled via a dummy transaction that has a user bid of zero but non-zero value for searchers).

[10] The combination of (i)–(iii) can technically be achieved by using the tipless mechanism and always ignoring any searchers that might be present. Our interest in Section 4 will be the incentive-compatibility properties of a more interesting TFM that incorporates searchers in a way that resembles current practice; the goal in Section 5 is to design novel TFMs that, in addition to satisfying (i) – (iii) and unlike the searcher-excluding tipless mechanism, guarantee a constant fraction of the maximum-possible welfare.

2.5.4 Welfare

With searchers, we redefine the welfare (1) of a block B to reflect the private valuations of searchers and the fact that the proposer is assumed to have an all-zero valuation:

$$W(B) := \sum_{t \in B_T} v_t + \sum_{w \in B_S} v_w, \tag{6}$$

where B_T and B_S denote the transactions and bundles, respectively, in the block B.

3 An Impossibility Result for DSIC and BPIC Mechanisms

3.1 Can DSIC and BPIC Be Achieved Simultaneously?

The DSIC property (Definition 10) encodes the idea of a transaction fee mechanism with "good UX," meaning that bidding is straightforward for users. Given the unilateral power of BPs in typical blockchain protocols, the BPIC property (Definition 9) would seem necessary, absent any additional assumptions, to have any faith that a TFM will be carried out by BPs as intended. One can imagine a long wish list of properties that we'd like a TFM to satisfy; can we at least achieve these two?

The tipless mechanism [45] is an example of a TFM that is DSIC and BPIC in the special case of passive BPs. This TFM is also "non-trivial," in the sense that users generally pay for the privilege of transaction inclusion. With active BPs, meanwhile, the DSIC and BPIC properties can technically be achieved simultaneously by the following "trivial" TFM: the payment rule $\mathbf{p}$ and burning rule q are identically zero, and the allocation rule $\mathbf{x}$ instructs the BP to choose the feasible block that maximizes its private value (breaking ties in a bid-independent way). This TFM is BPIC by construction, and it is DSIC because a user has no control over whether it is included in the chosen block (it's either in the BP's favorite block or it's not) or its payment (which is always 0).

Thus, the refined version of the key question is:

Does there exist a non-trivial TFM that is DSIC and BPIC with active BPs?

3.2 Only Trivial Mechanisms Can Be DSIC and BPIC

The main result of this section is a negative answer to the preceding question. By the *range* of a bidding strategy σ, we mean the set of bid vectors realized by nonnegative valuations: $\{\sigma(\mathbf{v}) : \mathbf{v} \geq 0\}$, where $\sigma(\mathbf{v})$ denotes the componentwise application of σ.

▶ **Theorem 11** (Impossibility of DSIC, BPIC, Non-Triviality)**.** *If the TFM* $(\mathbf{x}, \mathbf{p}, q)$ *is DSIC with bidding strategy* σ *and BPIC with active block producers, then the payment rule* $\mathbf{p}$ *is identically zero on the range of* σ*.*

The proof of Theorem 11 is quite general and holds even if BPs are restricted to have nonnegative additive valuations and all known transactions have the same size and can be included simultaneously into a single feasible block. We sketch the proof here with details in the full version. Towards a contradiction, let $(\mathbf{x}, \mathbf{p}, q)$ define a BPIC and DSIC TFM with a non-zero payment rule. Thus assume there is a transaction t^* and a set of bids $\mathbf{b} = (b_{t^*}, \mathbf{b}_{-t^*})$ where $p_{t^*}(B, \mathbf{b}) > 0$ where B is the BP's BPS maximizing block for some v_{BP}. Now define an alternative bid vector $\mathbf{b}' = (0, \mathbf{b}_{-t^*})$ that is identical to $\mathbf{b}$ except for t^* dropping their bid to 0. Since $0 < p_{t^*}(B, \mathbf{b})$, for the mechanism to be DSIC, we must have $\mathbf{x}_t(v_{BP}, \mathbf{b}', \mathcal{B}) = 0$ *regardless* of v_{BP}. However, we show that we can define a BP valuation $\hat{v}_{BP}$ where $\hat{v}_{BP}(\{t^*\})$ is sufficiently high such that for the mechanism to be BPIC, $\mathbf{x}_{t^*}(\hat{v}_{BP}, \mathbf{b}', \mathcal{B}) = 1$ even with

$\mathbf{b}'_{t^*} = 0$. This in turn leads to a contradiction. The technical part of the proof lies in choosing $\hat{v}_{BP}$ properly to show there is *no* choice of payment and burning rule such that there is a consonant allocation rule where the BP doesn't include t^* under $\mathbf{b}'$.

3.2.1 Discussion

The role of an impossibility result like Theorem 11 is to illuminate the most promising paths forward. From it, we learn that our options are (i) constrained; and (ii) already being actively explored by the blockchain research community. Specifically, with active BPs, to design a non-trivial TFM, we must choose from among three options:

1. Give up on "good UX," at least as it is expressed by the DSIC property.
2. Give up on the BPIC property, presumably compensating with restrictions on block producer behavior (perhaps enforced using, e.g., trusted hardware [23] or cryptographic techniques [14]).
3. Expand the TFM design space, for example by incorporating order flow auctions (e.g., [36]) or block producer competition (e.g., [18]) to expose information about a BP's private valuation to a TFM. We explore this idea further in Sections 4 and 5.

▶ Remark 12 (Variations of Theorem 11). Variations on the proof of Theorem 11 show that the same conclusion holds for:

(a) BPs that have a non-zero private value for only one block (a very special case of single-minded valuations). This version of the argument does not require the consistent tie-breaking assumption in Definition 7(b).

(b) Burning rules that need not be nonnegative (i.e., rules that can print money), provided that, for every bid vector $\mathbf{b}$, there is a finite lower bound on the minimum-possible burn $\min_{B \in \mathcal{B}} q(B, \mathbf{b})$. (This would be the case if, for example, the blockset $\mathcal{B}$ is finite.)

(c) Bid spaces and payment rules that need not be nonnegative (i.e., with negative bids and user rebates allowed, subject to individual rationality), provided there is a finite minimum bid $b_{min} \in (-\infty, 0]$ and that $p_t(B, \mathbf{b}) = b_{min}$ whenever $t \in B$ with $b_t = b_{min}$. In this case, the argument shows that the payment rule $\mathbf{p}$ must be identically equal to b_{min} on the range of σ.

3.3 The Welfare Achieved by DSIC and BPIC Mechanisms

Theorem 11 shows that TFMs that are DSIC and BPIC must be "trivial," in the sense that users are never charged for the privilege of transaction inclusion. The next result formalizes the intuitive consequence that such TFMs may, if both users and the BP follow their recommended actions, produce blocks with welfare arbitrarily worse than the maximum possible. (Recall that the welfare $W(B)$ of a block B is defined in expression (1) in Section 2.2.) That is, no approximately welfare-maximizing TFM can be both DSIC and BPIC with active BPs. This result is not entirely trivial because the conclusion of Theorem 11 imposes no restrictions on the burning rule of a TFM.

▶ **Theorem 13** (Impossibility of DSIC, BPIC, and Non-Trivial Welfare Guarantees). *Let $(\mathbf{x}, \mathbf{p}, q)$ denote a TFM that is BPIC and DSIC with bidding strategy σ. For every approximation factor $\rho > 0$, there exists a BP valuation v_{BP}, BP blockset $\mathcal{B}$, block $B^* \in \mathcal{B}$, and transactions with corresponding user valuations $\mathbf{v}$ such that*

$$W(B) \leq \rho \cdot W(B^*),$$

where $B = \mathbf{x}(\sigma(\mathbf{v}), v_{BP}, \mathcal{B})$.

In the absence of a burning rule, Theorem 13 follows directly from Theorem 11, since any mechanism with $\mathbf{p} = 0$ effectively ignores user bids when choosing a block. However, it's not immediately obvious that there is no burning rule that can entice the BP to pick a welfare maximizing block even while ignoring users' payments. We show that DSIC rules such mechanisms out since a user being able to affect the burning rule and hence allocation rule while having their payment fixed to 0 would give them an incentive to misreport their value.

▶ Remark 14 (Generalizations of Theorem 13). The proof of Theorem 13 shows that the result holds already with BPs that have additive or single-minded valuations. (As discussed in Remark 12, Theorem 11 holds in both these cases, and the BP valuation v_{BP} used in the proof of Theorem 13 is both additive and single-minded). A slight variation of the proof shows that the result holds more generally for DSIC and BPIC TFMs that use a not-always-nonnegative burning rule, under the same condition as in Remark 12(b).

4 Transaction Fee Mechanisms with Searchers

4.1 Incorporating Searchers

The impossibility results in Section 3 are consistent with practice, in the sense that modern attempts to mitigate the negative consequence of MEV through economic mechanisms generally lie outside the basic design space of TFMs introduced in Sections 1–4. The dominant such mechanisms distribute the task of block production across multiple parties; in this section and the next, we adopt the model described in Section 2.5, which captures some of this complexity through the addition of searchers that can submit bundles (of a user-submitted transaction together with the searcher's value-extracting transactions) to a TFM. Recall from Section 2.5 that, in this model, what had been the private valuation v_{BP} of a vertically integrated BP is effectively distributed across a set of searchers, with the block proposer, having outsourced the task of value extraction, then acting passively to maximize its revenue (including the payments from searchers for included bundles). The winning bid of a searcher can be interpreted as an "MEV oracle" that provides a TFM with an estimate of the value that can be extracted from the bundled transaction. In this sense, the TFM design space with searchers is richer than the basic model with users only, and there is hope that a TFM can take advantage of such estimates to define payments for user-submitted transactions in a DSIC-respecting way (e.g., with searchers' bids leading in some cases to user refunds). Indeed, we'll see that this expanded design space allows for positive results that would be impossible in the basic model.

In this section, we propose an abstraction of how searchers have traditionally been incorporated into the block production process, inspired specifically by mev-geth (see Section 2.5), and study the incentive-compatibility properties of the resulting mechanism. Section 5 explores the TFM design space with searchers more generally, with a focus on welfare guarantees.

4.2 The s-Tipless Mechanism

We next introduce the *Searcher Augmented Tipless Mechanism (s-tipless mechanism)*. Like the EIP-1559 and tipless mechanisms, it has a fixed base fee r that is charged per unit size. Intuitively, for each user-submitted transaction t, the mechanism runs a first-price auction among the interested searchers; such an auction is often referred to as an "order-flow auction." (Thus, the mechanism does not attempt to be DSIC for searchers.) If the winning bid b_w

in this auction is high enough to pay the base fee charges (i.e., $b_w \geq r \cdot s'_t$, where s'_t is the size of a bundle that contains t), then w's bundle is included in the block and w pays its bid (while the user that submitted t pays nothing). If the winning searcher bid is less than $r \cdot s'_t$ then, if the user that submitted t bids at least the relevant base fee charges (i.e., $b_t \geq r \cdot s_t$), the transaction t is included in the block and the submitting user pays $r \cdot s_t$. In either case, all base fee revenues ($r \cdot s_t$ or $r \cdot s'_t$) are burned. (The block proposer may still collect revenue from the first-price auction among searchers if the winning bid exceeds $r \cdot s'_t$.) In effect, searchers can cover base fee charges for a user if their transaction is sufficiently valuable for them.

▶ **Definition 15** (Searcher-Augmented Tipless Mechanism (s-tipless mechanism)). *Fix a base fee $r \geq 0$:*

(a) **Allocation rule:** *A transaction should be included if either it clears its base fee, or it has a bundle that clears the bundle's base fee. If multiple bundles for a transaction clear the base fee, the bundle with the highest bid should be included. For each $t \in T$, let S_t denote the submitted bundles that contain t, w a generic such bundle, and $t^* = \text{argmax}_{w \in S_t}\{b_w\}$. Define*

$$S^* = \{t^* : t \in T, b_{t^*} \geq r \cdot s'_t\} \text{ and } T^* = \{t \in T : b_t \geq r \cdot s_t \vee S_t \cap S^* \neq \emptyset\},$$

and the allocation rule by

$$\mathbf{x}(\mathbf{b}, \mathcal{B}) = T^* \cup S^*.$$

(b) **Payment rule:**
For all transactions t in a block B:

$$p_t(B, \mathbf{b}) = \begin{cases} 0 & \text{if } S_t \cap B \neq \emptyset \\ r \cdot s_t & \text{otherwise.} \end{cases}$$

For all bundles w in a block B:

$$p_w(B, \mathbf{b}) = b_w.$$

(c) **Burning rule:** *For a block B with transactions B_T and bundles B_S,*[11]

$$q(B, \mathbf{b}) = \sum_{t \in B_T} r \cdot s_t + \sum_{w \in B_S} r \cdot (s'_t - s_t).$$

In Definition 15 and Theorem 16 below, we assume for simplicity that the base fee r is large enough that there is sufficient room in the block for all of the transactions that the mechanism would like to include (i.e., all transactions for which either the user or some searcher is willing to cover the relevant base fee charges). In practice, a la the EIP-1559 mechanism, the base fee r would generally be adjusted by local search so that this property typically holds. Definition 15 and Theorem 16 can be extended to the general case (with contention between sufficiently high-bidding transactions and bundles) by redefining the allocation rule to maximize the total revenue (i.e., $\sum_{t^i \in B_S}(b_{t^i} - r \cdot s'_t)$), breaking ties in a consistent fashion.

[11] We subtract s_t for every bundle $w \in B_s$ as to not double count s_t both as part of a bundle and as a standalone transaction

▶ **Theorem 16.** *The s-tipless mechanism is DSIC for users and BPIC.*

We give a sketch of the proof here with the details in the full version. To see that the mechanism is DSIC for users, note that if a transaction has a bundle included for it, then it always pays 0 regardless of what it bids, trivially giving the user a dominant bidding strategy. Otherwise, the user faces a fixed price for inclusion and hence has a dominant strategy to only bid above that price if their value is above it. To see that the mechanism is BPIC, note that the only revenue the BP gets is from searcher bids above the basefee. Standalone transactions have no net effect on the BP's BPS. Thus any allocation rule that picks the highest bid searchers above the base fee and picks transactions clearing the base fee is consonant. Furthermore, since the amount searchers pay is only a function of their own bids, the BP has no way to increase their BPS via inserting shill bids.

5 Welfare Guarantees

This section continues to investigate transaction fee mechanism design in the presence of searchers, as in the model in Section 2.5. While the previous section proposed abstractions for some of the economic mechanisms that are currently used in practice, this section zooms out and explores the expanded design space more generally.

5.1 What Do We Want from a TFM?

Starting from a blank page, we naturally want to design a mechanism that scores well with respect to all the criteria we have considered thus far:

(P1) DSIC for users;
(P2) DSIC for (active) searchers;
(P3) BPIC (with a passive block proposer);
(P4) good welfare guarantees.

Without searchers, Theorem 13 shows that the combination of (P1), (P3), and (P4) is unachiavable. We also noted in passing (footnote 10) that the tipless mechanism, modified to always ignore searchers, satisfies (P1)–(P3). (Such a mechanism can obviously lead to a highly welfare-suboptimal outcome when the valuations of searchers are significantly bigger than those of the users.)

Given the welfare-maximization goal (P4), one obvious starting point is the Vickrey-Clarke-Groves (VCG) mechanism, which in this context would accept bids from all users and searchers, output a feasible block that maximizes the social welfare (6) (taking users' and searchers' bids at face value), and charge each user or searcher its externality (i.e., what the maximum social welfare would have been had that user or searcher been absent). As always, the VCG mechanism is DSIC (in this case, for both users and searchers) and maximizes the social welfare at its dominant-strategy equilibrium. It does not, however, satisfy property (P3). For example, even with only one user-submitted transaction and a number of corresponding searchers (i.e., a second-price auction), the block proposer is generally incentivized to masquerade as a searcher and insert a shill bid (just below the highest searcher bid) to increase its revenue.[12]

[12] A similar problem would arise if the s-tipless mechanism in Section 4 were defined with second-price rather than first-price searcher auctions.

One easy way to turn the VCG mechanism – or really, any TFM with a passive block proposer – into a BPIC mechanism is to always burn all the payments made by users and searchers. The block proposer would then be indifferent over blocks and willing to carry out an arbitrary allocation rule. An extension of this idea that attempts to trade welfare for a non-zero amount of BP revenue would be to use bidder-specific reserve prices (like $r \cdot s_t$ and $r \cdot s'_t$ in the s-tipless mechanism) that don't get burned.[13]

Summarizing, the VCG mechanism with all payments burned satisfies all of (P1)–(P4), and in particular shows that the addition of searchers allows TFMs to circumvent the impossibility result in Theorem 13. Should we declare victory?

5.2 Sybil-Proof Mechanisms

In a permissionless blockchain protocol like Bitcoin or Ethereum, it is easy to generate multiple identities in an undetectable way. For example, a user can easily participate as a "fake searcher" in a TFM if it so chooses. This challenge of "sybils," especially in tandem with the non-downward-closed nature of the set of feasible blocks (with inclusion of a bundle implying inclusion of the corresponding transaction), renders the VCG mechanism extremely easy to manipulate (despite being DSIC for users and searchers separately).

For example, consider a sample instance with a block size of k where all the transactions and bundles are unit sized and there is one searcher per transaction, i.e. $\forall t \in T$, $s_t = s'_t = 1$ and $S_t = \{t^*\}$. In this case, the VCG mechanism will include the transactions and bundles corresponding to the k highest values of $b_t + b_{t^*}$. Let the $(k+1)$th-highest of these values be r. The included user and searcher for transaction t would then pay $\max\{r - b_{t^*}, 0\}$ and $\max\{r - b_t, 0\}$ respectively. In the case that both $b_t \geq r$ and $b_{t^*} \geq r$ it follows that neither the user nor searcher has to pay anything at all. Hence there is a clear incentive for a user to deviate by making their bid arbitrarily high and including an arbitrarily high searcher bid for their transaction to get included without paying anything. Even a user with only ϵ value for inclusion has an incentive to do this. It follows that users engaging in such manipulations can cause the mechanism to produce outcomes with arbitrarily bad welfare. Furthermore, in permissionless blockchains, such manipulations are easy to carry out. This motivates seeking out TFMs that are, among other properties, "sybil-proof" in some sense.

Our definition of sybil-proofness (for users and searchers) mirrors our definition of BPIC, in that it asserts that the party in question cannot increase their utility through the submission of fake transactions and shill bids.

▶ **Definition 17** (Sybil-Proofness). *A mechanism is* sybil-proof *if for every agent t and every vector of bids $\mathbf{b}'$, there exists some bid b_t such that $u_t(b_t) \geq \bar{u}_t(\mathbf{b}')$ where $\bar{u}_t(\cdot)$ is the agent's net utility across the multiple transactions and/or bundles they submitted.*

Intuitively, this definition asserts that a user or searcher should never earn more utility from submitting multiple bids than they could have through a single bid for their transaction or bundle.

We now augment our previous desiderata with:

(P5) sybil-proof.

Next we provide a TFM that satisfies the full set (P1)–(P5) of desired properties.

[13] A mechanism with any non-zero reserve prices cannot offer any worst-case approximate welfare guarantees: for all the mechanism knows, only one participant has a non-zero valuation, which is just below the mechanism's non-zero reserve price for that participant. We leave a Bayesian analysis (e.g., with the choice of reserve prices informed by historical bidding data) of the revenue-welfare trade-offs of such mechanisms to future work.

5.3 The Searcher-Augmented Knapsack Auction

We will consider a mechanism that chooses which transactions and bundles to include based on their bid-to-size ratios. For ease of exposition, we assume that these ratios are distinct. (This assumption can be removed through standard lexicographic tie-breaking.) The mechanism finds a threshold ratio such that all transactions and bundles that have bid-to-size ratios above this threshold can fit into the block. This ratio is then used as a per-size price charged to included transactions and bundles. Similarly to the s-tipless mechanism, an included bundle pays all the costs for its corresponding transaction. For included bundles, in the case that the second highest bundle bid for a transaction is greater than the threshold payment, the winning searcher pays the second-highest bid instead. Finally, the burning rule is set to be the sum of users' and searchers' payments so that the block proposer always receives zero BPS.

▶ **Definition 18** (Searcher-Augmented Knapsack Auction (SAKA)).

(a) ***Allocation rule:*** *Recall that t^* denotes the bundle with the highest bid for transaction t. For a given μ, let*

$$S^\mu = \{t^* : t \in T, b_{t^*}/s'_t \geq \mu\} \text{ and } T^\mu = \{t \in T : b_t/s_t \geq \mu \vee S_t \cap S^\mu \neq \emptyset\}.$$

Then let $B^\mu = T^\mu \cup S^\mu$ be the block consisting of all transactions and bundles that have a bid-to-size ratio of at least μ.[14]
Define $\mu^* := \inf\{\mu : \sum_{t \in B^\mu_T} s_t + \sum_{t^i \in B^\mu_S}(s'_t - s_t) \leq k\}$, where B^μ_T and B^μ_S denote the transactions and bundles, respectively, in the block B^μ. Then,

$$x(\mathbf{b}, \mathcal{B}) = B^{\mu^*}.$$

(b) ***Payment rule:*** *Define $b_{t'} := \max_{t^i \in S^t, t^i \neq t^*}\{b_{t^i}\}$ as the second-highest bundle bid for transaction t. (If there is no such bid, interpret $b_{t'}$ as 0.) For $t \in B_T$:*

$$p_t(B, \mathbf{b}) = \begin{cases} 0 & \text{if } S_t \cap B \neq \emptyset \\ \mu^* \cdot s_t & \text{otherwise.} \end{cases}$$

For $t^i \in B_S$:

$$p_{t^i}(B, \mathbf{b}) = \max\{\mu^* \cdot s'_t, b_{t'}\}.$$

(c) ***Burning rule:***

$$q(B, \mathbf{b}) = \sum_{t \in B_T} p_t(B, \mathbf{b}) + \sum_{w \in B_S} p_w(B, \mathbf{b}).$$

5.4 Analysis

We consider the incentive-compatibility properties of the SAKA mechanism in Theorem 19 and its welfare guarantee in Theorem 20. We conclude with Theorem 21, which shows that the welfare guarantee in Theorem 20 is near-optimal among TFMs that satisfy properties (P1)–(P5).

[14] Subject to the usual constraint that each transaction is included (by itself or as part of a bundle) at most once.

▶ **Theorem 19.** *The Searcher-Augmented Knapsack Auction (SAKA) mechanism is DSIC for both users and searchers, BPIC, and sybil-proof.*

We give the main ideas of the proof here with details in the full version. SAKA being BPIC follows immediately from the burning rule. To see that the mechanism is DSIC, we can focus on the case where a transaction doesn't have a bundle included for it (otherwise the transaction always pays 0). The allocation rule is monotone since once a user's bid clears $\mu^* \cdot s_t$ they will always be included. Furthermore, $\mu^* \cdot s_t$ is the minimal amount t can bid to be included since otherwise bidding below $\mu^* \cdot s_t$ and being included would contradict the definition of μ^*. The case for searcher DSIC follows identically with the addition of needing to pay at least the second highest searcher bid to still be included. Sybil-proofness follows from the fact that μ^* is weakly increasing in the number of bids. So users and searchers have no way to decrease their payment by bidding on fake transactions.[15]

We parameterize the mechanism's welfare guarantee by the maximum fraction γ of a block's capacity that is consumed by a single transaction or bundle. (In many blockchain protocols, γ is typically 2% or less.)

▶ **Theorem 20.** *Assuming truthful bids by users and searchers, the outcome of the SAKA mechanism has social welfare at least* $(1-\gamma)/2$ *times the maximum possible welfare.*

Note that SAKA implements a greedy knapsack algorithm, except it scores bundles using a scoring rule of $\frac{v_{t^*}}{s'_t}$ instead of $\frac{v_t+v_{t^*}}{s'_t}$ as it would optimally. However, we either have $\frac{v_{t^*}}{s'_t} \geq \frac{v_t+v_{t^*}}{2s'_t}$ or $\frac{v_{t^*}}{s'_t} < \frac{v_t+v_{t^*}}{2s'_t} \implies \frac{v_t}{s_t} > \frac{v_t+v_{t^*}}{2s'_t}$. It follows that the density SAKA assigns to a bundle is either at least half of what it should be or that the bundle's corresponding transaction carries half the bundle's true density by itself. Since the mechanism includes transactions and bundles with the highest densities, it follows that a bundle being left out because its density was misjudged would be replaced with transactions and/or bundles with at least half its density. Since the greedy algorithm will fill up at least $1-\gamma$ of the block limit, this implies an approximation factor of $\frac{1-\gamma}{2}$. The details can be found in the full version.

Our final result shows that, modulo the factor of $1-\gamma$ – which, as discussed above, is typically close to 1 in our context – the welfare approximation guarantee in Theorem 20 is optimal among deterministic mechanisms that are both DSIC (for users and searchers) and sybil-proof in the sense of Definition 17. The key insight is that it's difficult to split the payment between user and searcher when a bundle is included due to the bundle requiring its corresponding transaction's inclusion. In particular we show DSIC + sybil-proofness implies user/searcher pairs can only be included based on the max of their values rather than the sum of their values (as would be optimal). We leave the details to the full version.

▶ **Theorem 21.** *No deterministic mechanism that is DSIC for users and searchers and sybil-proof can achieve better than a 1/2-approximation to the optimal social welfare, even when transaction sizes are a negligible fraction of the block size.*

References

1 Mev blocker. `https://mevblocker.io/`, 2024.

2 What is mev-boost. `https://docs.flashbots.net/flashbots-mev-boost/introduction`, 2024.

[15] As a bonus, the SAKA auction can be implemented as a deferred acceptance mechanism [35] and is therefore also robust to certain forms of collusion between users and searchers (formally, the mechanism is weakly groupstrategypoof).

3 Hayden Adams, Emily Williams, Will Pote, Zhiyuan Yang, Noah Zinsmeister, Xin Wan, Allen Lin, Riley Campbell, Dan Robinson, Mark Toda, Matteo Leibowitz, Eric Zhong, and Alex Karys. Uniswapx protocol, July 2023. Available at Uniswap.org.

4 Mohammad Akbarpour and Shengwu Li. Credible auctions: A trilemma. *Econometrica*, 88(2):425–467, 2020.

5 Kushal Babel, Philip Daian, Mahimna Kelkar, and Ari Juels. Clockwork finance: Automated analysis of economic security in smart contracts. In *IEEE Symposium on Security and Privacy*, pages 2499–2516, 2023.

6 Maryam Bahrani, Pranav Garimidi, and Tim Roughgarden. Centralization in block building and proposer-builder separation. In Jeremy Clark and Elaine Shi, editors, *Financial Cryptography and Data Security - 28th International Conference, FC 2024, Curaçao, March 4-8, 2024*, 2024.

7 Maryam Bahrani, Pranav Garimidi, and Tim Roughgarden. Transaction fee mechanism design in a post-mev world. *IACR Cryptol. ePrint Arch.*, page 331, 2024. URL: https://eprint.iacr.org/2024/331.

8 Massimo Bartoletti, James Hsin-yu Chiang, and Alberto Lluch-Lafuente. Maximizing extractable value from automated market makers. In Ittay Eyal and Juan A. Garay, editors, *Financial Cryptography and Data Security - 26th International Conference, FC 2022, Grenada, May 2-6, 2022, Revised Selected Papers*, volume 13411 of *Lecture Notes in Computer Science*, pages 3–19. Springer, 2022.

9 Massimo Bartoletti and Roberto Zunino. A theoretical basis for blockchain extractable value. *CoRR*, abs/2302.02154, 2023. arXiv:2302.02154.

10 Soumya Basu, David Easley, Maureen O'Hara, and Emin Gün Sirer. Towards a functional fee market for cryptocurrencies. *arXiv preprint arXiv:1901.06830*, 2019.

11 Iddo Bentov, Yan Ji, Fan Zhang, Lorenz Breidenbach, Philip Daian, and Ari Juels. Tesseract: Real-time cryptocurrency exchange using trusted hardware. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, pages 1521–1538. ACM, 2019.

12 Vitalik Buterin, Eric Conner, Rick Dudley, Matthew Slipper, Ian Norden, and Abdelhamid Bakhta. EIP-1559: Fee market change for ETH 1.0 chain. URL: https://github.com/ethereum/EIPs/blob/master/EIPS/eip-1559.md, 2019.

13 Christian Cachin, Jovana Micic, Nathalie Steinhauer, and Luca Zanolini. Quick order fairness. In Ittay Eyal and Juan A. Garay, editors, *Financial Cryptography and Data Security - 26th International Conference, FC 2022, Grenada, May 2-6, 2022, Revised Selected Papers*, volume 13411 of *Lecture Notes in Computer Science*, pages 316–333. Springer, 2022.

14 Jon Charbonneau. Encrypted mempools. URL: https://joncharbonneau.substack.com/p/encrypted-mempools, March 2023.

15 Tarun Chitra, Matheus V. X. Ferreira, and Kshitij Kulkarni. Credible, optimal auctions via blockchains. *arXiv preprint arXiv:2301.12532*, 2023.

16 Hao Chung and Elaine Shi. Foundations of transaction fee mechanism design. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3856–3899. SIAM, 2023.

17 Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In *2020 IEEE Symposium on Security and Privacy, SP 2020, San Francisco, CA, USA, May 18-21, 2020*, pages 910–927. IEEE, 2020.

18 Domothy. Burning MEV through block proposer auctions. URL: https://ethresear.ch/t/burning-mev-through-block-proposer-auctions/14029, October 2022.

19 Meryem Essaidi, Matheus V. X. Ferreira, and S Matthew Weinberg. Credible, strategyproof, optimal, and bounded expected-round single-item auctions for all distributions. In *Proceedings of the 13th Innovations in Theoretical Computer Science Conference (ITCS)*, 2022.

20 Matheus V. X. Ferreira, Daniel J. Moroz, David C. Parkes, and Mitchell Stern. Dynamic posted-price mechanisms for the blockchain transaction-fee market. In *Proceedings of the 3rd ACM Conference on Advances in Financial Technologies*, pages 86–99, 2021.

21 Matheus V. X. Ferreira and David C. Parkes. Credible decentralized exchange design via verifiable sequencing rules. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 723–736, 2023.

22 Matheus V. X. Ferreira and S Matthew Weinberg. Credible, truthful, and two-round (optimal) auctions via cryptographic commitments. In *Proceedings of the 21st ACM Conference on Economics and Computation*, pages 683–712, 2020.

23 Flashbots. The Future of MEV is SUAVE. URL: https://writings.flashbots.net/the-future-of-mev-is-suave/, November 2022.

24 Yotam Gafni and Aviv Yaish. Greedy transaction fee mechanisms for (non-) myopic miners. *arXiv preprint arXiv:2210.07793*, 2022.

25 Stephane Gosselin and Ankit Chiplunkar. The orderflow auction design space. https://frontier.tech/the-orderflow-auction-design-space, 2023.

26 Lioba Heimbach and Roger Wattenhofer. Eliminating sandwich attacks with the help of game theory. In Yuji Suga, Kouichi Sakurai, Xuhua Ding, and Kazue Sako, editors, *ASIA CCS '22: ACM Asia Conference on Computer and Communications Security, Nagasaki, Japan, 30 May 2022 - 3 June 2022*, pages 153–167. ACM, 2022.

27 Mahimna Kelkar, Soubhik Deb, Sishan Long, Ari Juels, and Sreeram Kannan. Themis: Fast, strong order-fairness in byzantine consensus. In Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda, editors, *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, CCS 2023, Copenhagen, Denmark, November 26-30, 2023*, pages 475–489. ACM, 2023.

28 Mahimna Kelkar, Fan Zhang, Steven Goldfeder, and Ari Juels. Order-fairness for byzantine consensus. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part III*, volume 12172 of *Lecture Notes in Computer Science*, pages 451–480. Springer, 2020.

29 Rami Khalil, Arthur Gervais, and Guillaume Felley. TEX - A securely scalable trustless exchange. *IACR Cryptol. ePrint Arch.*, page 265, 2019. URL: https://eprint.iacr.org/2019/265.

30 Kshitij Kulkarni, Theo Diamandis, and Tarun Chitra. Towards a theory of maximal extractable value I: constant function market makers. *CoRR*, abs/2207.11835, 2022. arXiv:2207.11835.

31 Klaus Kursawe. Wendy, the good little fairness widget: Achieving order fairness for blockchains. In *AFT '20: 2nd ACM Conference on Advances in Financial Technologies, New York, NY, USA, October 21-23, 2020*, pages 25–36. ACM, 2020.

32 Ron Lavi, Or Sattath, and Aviv Zohar. Redesigning bitcoin's fee market. *ACM Transactions on Economics and Computation*, 10(1):1–31, 2022.

33 S. Leonardos, B. Monnot, D. Reijsbergen, S. Skoulakis, and G. Piliouras. Dynamical analysis of the EIP-1559 Ethereum fee market. In *Proceedings of the 3rd ACM Advances in Financial Technologies*, 2021.

34 Dahlia Malkhi and Pawel Szalachowski. Maximal extractable value (MEV) protection on a DAG. In Yackolley Amoussou-Guenou, Aggelos Kiayias, and Marianne Verdier, editors, *4th International Conference on Blockchain Economics, Security and Protocols, Tokenomics 2022, December 12-13, 2022, Paris, France*, volume 110 of *OASIcs*, pages 6:1–6:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

35 Paul Milgrom and Ilya Segal. Deferred-acceptance auctions and radio spectrum reallocation. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 185–186, 2014.

36 Robert Miller. MEV-Share: programmably private orderflow to share MEV with users. URL: https://collective.flashbots.net/t/

mev-share-programmably-private-orderflow-to-share-mev-with-users/1264, February 2023.

37 Roger B Myerson and Mark A Satterthwaite. Efficient mechanisms for bilateral trading. *Journal of economic theory*, 29(2):265–281, 1983.

38 Noam Nisan. Serial monopoly on blockchains. URL: https://www.cs.huji.ac.il/~noam/publications/ser-mon.pdf, April 2023.

39 Alexandre Obadia, Alejo Salles, Lakshman Sankar, Tarun Chitra, Vaibhav Chellani, and Philip Daian. Unity is strength: A formalization of cross-domain maximal extractable value. *CoRR*, abs/2112.01472, 2021. arXiv:2112.01472.

40 Mallesh Pai and Max Resnick. Structural advantages for integrated builders in mev-boost. *arXiv preprint arXiv:2311.09083*, 2023.

41 Kaihua Qin, Liyi Zhou, and Arthur Gervais. Quantifying blockchain extractable value: How dark is the forest? In *43rd IEEE Symposium on Security and Privacy, SP 2022, San Francisco, CA, USA, May 22-26, 2022*, pages 198–214. IEEE, 2022.

42 Kaihua Qin, Liyi Zhou, Benjamin Livshits, and Arthur Gervais. Attacking the DeFi ecosystem with flash loans for fun and profit. In Nikita Borisov and Claudia Díaz, editors, *Financial Cryptography and Data Security - 25th International Conference, FC 2021, Virtual Event, March 1-5, 2021, Revised Selected Papers, Part I*, volume 12674 of *Lecture Notes in Computer Science*, pages 3–32. Springer, 2021.

43 Max Resnick. Contingent fees in order flow auctions. *arXiv preprint arXiv:2304.04981*, 2023.

44 Tim Roughgarden. *Twenty Lectures on Algorithmic Game Theory*. Cambridge University Press, 2016.

45 Tim Roughgarden. Transaction fee mechanism design. *ACM SIGecom Exchanges*, 19(1):52–55, 2021. Full version at https://arxiv.org/abs/2106.01340.

46 Alejo Salles. On the formalization of MEV. URL: https://collective.flashbots.net/t/on-the-formalization-of-mev/879, decemember 2021.

47 Elaine Shi, Hao Chung, and Ke Wu. What can cryptography do for decentralized mechanism design. *arXiv preprint arXiv:2209.14462*, 2022.

48 Christof Ferreira Torres, Ramiro Camino, and Radu State. Frontrunner jones and the raiders of the dark forest: An empirical study of frontrunning on the Ethereum blockchain. In Michael Bailey and Rachel Greenstadt, editors, *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, pages 1343–1359. USENIX Association, 2021.

49 Ke Wu, Elaine Shi, and Hao Chung. Maximizing miner revenue in transaction fee mechanism design. In Venkatesan Guruswami, editor, *15th Innovations in Theoretical Computer Science Conference, ITCS 2024, January 30 to February 2, 2024, Berkeley, CA, USA*, volume 287 of *LIPIcs*, pages 98:1–98:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024.

50 Sen Yang, Fan Zhang, Ken Huang, Xi Chen, Youwei Yang, and Feng Zhu. Sok: MEV countermeasures: Theory and practice. *CoRR*, abs/2212.05111, 2022.

51 Andrew C.-C. Yao. An incentive analysis of some Bitcoin fee designs. In *Proceedings of the 47th International Colloquium on Automata, Languages, and Programming (ICALP)*, 2020.

52 Haoqian Zhang, Louis-Henri Merino, Vero Estrada-Galiñanes, and Bryan Ford. Flash freezing flash boys: Countering blockchain front-running. In *42nd IEEE International Conference on Distributed Computing Systems, ICDCS Workshops, Bologna, Italy, July 10, 2022*, pages 90–95. IEEE, 2022.

53 Zishuo Zhao, Xi Chen, and Yuan Zhou. Bayesian-Nash-incentive-compatible mechanism for blockchain transaction fee allocation. *arXiv preprint arXiv:2209.13099*, 2022.