Price of MEV:
Towards a Game Theoretical Approach to MEV

Bruno Mazorra
Nokia Bell-labs
Universitat Pompeu Fabra
brunomazorra@gmail.com

Michael Reynolds
University College London
mireynolds@pm.me

Vanesa Daza
Universitat Pompeu Fabra
vanesa.daza@upf.edu

August 30, 2022

Abstract

Maximal (also miner) extractable value, or MEV, usually refers to the value that *privileged* players can extract by strategically ordering, censoring, and placing transactions in a blockchain. Each blockchain network, which we refer to as a domain, has its own consensus, ordering, and block-creation mechanisms, which gives rise to different optimal strategies to extract MEV. The strategic behaviour of rational players, known as searchers, lead to MEV games that have different impacts and externalities in each domain. Several ordering mechanisms, which determine the inclusion and position of transactions in a block, have been considered to construct alternative games to organise MEV extraction, and minimize negative externalities; examples include sealed bid auctions, first input first output, and private priority gas auctions. However, to date, no sufficiently formal and abstract definition of MEV games have been made. In this paper, we take a step toward the formalization of MEV games and compare different ordering mechanisms and their externalities. In particular, we attempt to formalize games that arise from common knowledge MEV opportunities, such as arbitrage and sandwich attacks. In defining these games, we utilise a theoretical framework that provides groundwork for several important roles and concepts, such as the searcher, sequencer, domain, and bundle. We also introduce the price of MEV as the price of anarchy of MEV games, a measure that provides formal comparison between different ordering mechanisms.

Keywords:Blockchain, MEV, Game Theory, Price of Anarchy

1 Introduction

The notion of miner-extractable value (MEV) introduced in [5], and formally defined in [3], measures the value that miners can extract by strategically ordering, censoring, and including transactions in a block. In general, block proposers (typically validators or miners), have the power to dictate or influence the inclusion and ordering of pending transactions in a new block. Thus, rational proposers have access to extra rewards per block. However as shown in [5], this power is not limited to block proposers. Blockchain users and bots, usually termed *searchers*, can use strategies like spamming transactions or outbidding competitors to extract MEV themselves. For this and other reasons, the term miner-extractable value has evolved to mean *maximal extractable value*, formally defined in [16], which does not limit extraction to block proposers. We often refer to block proposers and searchers as players.

Depending on the network protocol, which we refer to as the domain, and type of MEV being extracted, rational searchers have incentives to take different actions to increase their expected revenue. For example, [5] showed that Ethereum, a network with higher propagation latency, incentivises searchers to widen their view of the mempool to outbid their competitors, while domains like Polygon PoS, where the propagation of transactions is probabilistic, incentivises searchers to spam transactions to extract MEV. This motivates a more general formulation of the MEV game presented in [5] such that models of rational searcher behaviour can be derived from a domain input.

1.1 Our contributions

The main goal of this paper is to formalise MEV games with a finite number of players under different

ordering mechanism and auction designs which are defined by various environment conditions and rules such as latency of players, PoW/PoS consensus, and mempool visibility. To this end, the following topics will be covered:

- Inspired by [3], we provide a formal definition of MEV and local MEV as an optimization problem constrained by player resources. In our definition, we emphasize relative constraints like capital, software resources, and view of the mempool.

- We formally define a more abstract and general MEV game that provides a foundation for analysing the incentives of searchers in different domains.

- We also introduce different negative externalities that arise from MEV games, and we formally define the Price of Anarchy in MEV games under different cost functions as a way to quantify their impact. We explore the specific case of block space misuse, which we refer to as the price of MEV, and compute this to characterize and classify different ordering mechanisms.

1.2 Previous Work

In Flashboys 2.0 [5] Daian et al. proposed a formalization of the priority gas auction (PGA) model and proved mathematically and empirically the existence of Grim-Trigger Nash equilibrium under certain conditions. However, the results are limited to two players and the PoW Ethereum domain, and do not consider reverted transaction costs proportional to the bid or that players can engage in Sybil attacks. In [9], the authors explore the equilibrium of sandwich MEV as a router game. Likewise, in [18], the authors compare empirically the miner's revenue and the searchers competition of arbitrage opportunities before and after the introduction of Flashbots MEV auctions on Ethereum. In [10], the first formalization of cross chain MEV was proposed. However, to the best of our knowledge, there is no formal and empirical study on the impacts of MEV on several distinct domains.

2 Theoretical Framework

In general, miner or maximum extractive value (MEV) is a term that refers to any excess profits that a block proposer or searcher can make based on transaction ordering and/or transaction inclusion. MEV was introduced in [5], formally defined in [3], and extended in cross-domain environments in [10].

Another similar definition was given in [16], except MEV was treated independently of a player. The MEV opportunities we consider in this paper are limited to the profits a player can obtain by modifying the blockchain state. In this section, we will formalize this kind of MEV opportunity using the concept of profitable 'bundles'. Then, we will define 'local' MEV as the maximally profitable bundle a specific player can construct. Similar to [10], we start by formally defining the domain and searcher:

Definition 2.1. A *domain* $\mathcal{D}$ is a self-contained system with a globally shared state $\texttt{st}$. This state is altered by various agents through actions (sending transactions, constructing blocks, slashing, etc.), that execute within a shared execution environment's semantics. Each domain has a predefined consensus protocol that includes a set of valid algorithms to order transactions, denoted by $\texttt{prt}(\mathcal{D})$.

A blockchain is a domain, however, there are other non-blockchain domains that also have MEV, like centralized exchanges.

Definition 2.2. A *searcher* (in general, we will call it a player) in a domain $\mathcal{D}$ is a participant that assumes that sequencers follow a specific set of rules and take strategic actions (send bundles with specific bids) to maximize their own utility. In general, we will assume that a player's utility depends linearly on their token balances.

In a domain $\mathcal{D}$ with state $\texttt{st}$, the update of the state $\texttt{st}$ after executing transactions $\texttt{tx}$ is given by $\texttt{st} \circ \texttt{tx}$. For an ordered set of transactions $B = \{\texttt{tx}_1, ..., \texttt{tx}_l\}$, we have the composition $\texttt{st} \circ B = \texttt{st} \circ \texttt{tx}_1 \circ .... \circ \texttt{tx}_l$.

Similar to [3], we use $\texttt{Addr}$ to denote the set of all possible accounts and $\mathbf{T}$ to denote the set of all tokens. We define $b : A \times \mathbf{T} \to \mathbb{Z}$ as the function that maps a pair of, an account and a token, to its current balance. More precisely, for $a \in \texttt{Addr}$ we let $b(a, \cdot)$ denote the balance of all tokens held in a and $b(a, T)$ denote the account balance of token T. Abusing notation, we will denote by $b(a)$ the value of $b(a, \cdot)$ priced by a numéraire E. That is, if there is a pricing vector $p = (p_{T \to E})_{T \in \mathbf{T}}$, then $b(a) = p \cdot b(a, \cdot) = \sum_{T \in \mathbf{T}} p_{T \to E} b(a, T)$.

Definition 2.3. An *ordering mechanism* is a set of rules that determines the order and inclusion of a set of transactions in a block. More formally, let $\mathcal{T}$ be the set of all transactions, an ordering mechanism is a map $\mathbf{or} : \mathcal{P}^{\leq}(\mathcal{T}) \to \mathcal{P}^{\leq}(\mathcal{T})$, where $\mathcal{P}^{\leq}(\mathcal{T})$ is the set of all ordered subsets of $\mathcal{T}$, such that $\mathbf{or}(T) \subseteq T$ for all $T \in \mathcal{P}^{\leq}(\mathcal{T})$.

Definition 2.4. A *sequencer* is an agent of a domain responsible for maintaining the liveness and consistency through a set of actions. We distinguish four types of sequencers: *dummy*, *dummy Byzantine*, *rational* and *partially rational*. A sequencer is *dummy* if he follows the validator consensus protocol $\mathtt{prt}(\mathcal{D})$. A sequencer is *dummy Byzantine* if they misbehave, but other nodes can detect his misconduct. A sequencer is *rational* if it follows a set of valid actions on the domain $\mathcal{D}$ to maximize its revenue (including deviating from an ordering mechanism). Therefore, if a player misbehaves to maximize their payoff but can not be identified, punished, or slashed, we say that is a rational player. A sequencer is *partially rational* if they commit to using a specific valid ordering mechanism to maximize its payoff.

On Ethereum, miners are usually partially rational. In general, miners run `mev-geth`, which receives a block transaction ordering of highest revenue from the Flashbots relayer. Rational sequencers could take the Flashbots block and reorg to maximize their own revenue. Nevertheless, miners do not deviate from the `mev-geth` intended protocol[1]. For this reason, players participating in the MEV extraction are not necessarily sequencers; they take a sequence of actions to bias nodes to maximize their own revenue. The set of actions that a sequencer can follow depends on the domain and can include arbitrary actions. Studying the impact that rational players and sequencers can have in a domain is helpful to bound the set of actions. For example, in selfish mining [17] the set of actions to construct or publish the private chains.

From now on, we will assume that sequencers are partially rational. An example of an ordering mechanism is the default `geth` client, which uses a greedy approximation algorithm to optimize the blocks' transaction fee revenue.

A sequencer receives a set of concurrent transactions $\mathtt{tx}_1, ..., \mathtt{tx}_n$ with gas price $m_1, ..., m_n$ and $g_1, ..., g_n$ units of gas. If the sequencer includes $\mathtt{tx}_i$, it obtains $m_i g_i$ in fees. Since the gas used per block is restricted in every domain by some constant L, the sequencer must choose a subset of transactions $\mathcal{T}$ such that $\sum_{i:\mathtt{tx}_i \in \mathcal{T}} g_i \leq L$. Then, a node that tries to maximize its revenues per block needs to solve the following Knapsack optimization problem, which we name Knapsack Extractable Value (KEV) problem:

$$
\begin{aligned}
\max \quad & \sum_{i=1}^{n} x_i m_i g_i \\
\text{s.t.} \quad & \sum_{i=1}^{n} x_i g_i \leq L, \\
& x_i \in \{0, 1\}.
\end{aligned}
$$

We note by $\mathrm{KEV}(\mathcal{T})$ the solutions' revenue of the optimization problem. Knapsack optimization problems [15] are NP-complete; that is, no known polynomial time algorithm finds an optimal solution. Thus, each sequencer usually chooses different algorithms to approximate the optimal solution. For example, Parity nodes order transactions by gas price m without considering the gas costs.

Another example is the Flashbots relayer, which uses a greedy approximation algorithm [1], ordering transactions by the ratio of miner payment and gas consumed (a natural extension of ordering by gas price taking into account direct payments and more than one transaction). However, the problem that the Flashbots relayer tries to solve is quite different, since it does not include reverted transactions or competing bundles. Moreover, we proved (see C) that there are examples where this algorithm does not produce a good approximation. In [2], the authors reformulate the block production as a linear programming problem without taking into account competing bundles. We will give more details after defining bundles.

Now that we have settled up some ground definitions, we will focus next on the formalization of the extraction of MEV opportunities. To simplify the games, we will not consider the complete MEV extraction per block, but separate the MEV opportunities into "independent/concurrent" ones.

Definition 2.5. Let $\mathcal{P}$ be the set of all players. A set of transactions $\mathcal{T} = \{\mathtt{tx}_1, ..., \mathtt{tx}_k\}$ and a state $\mathtt{st}$ induce an MEV opportunity in a domain $\mathcal{D}$ to a player $P \in \mathcal{P}$ if they can construct an ordered set of transactions B such that:

$$
\Delta b(P; \mathtt{st} \circ B, \mathtt{st}) := b(P, \mathtt{st} \circ B) - b(P, \mathtt{st}) > 0, \quad (1)
$$

where b is the balance of P with the corresponding order. We call B a *profitable bundle* or *bundle*. If B consists of a unique transaction, we say that B is an *MEV-transaction*. Each bundle can contain extra metadata, such as sequencer timestamps, bundle hash, sender ID, and gas price. For a given state $\mathtt{st}$, each bundle incurs execution costs called gas costs. From the bundle metadata and the domain state, the bundle execution incurs some payments, denoted by

---

[1]However, recent miners are deviating from the Flashbots protocol to extract more value. No formal or academic work has proved this yet. These are statements from some Flashbots team members.

pr(B), to a sequencer or set of sequencers responsible for executing the bundle.

Definition 2.6. A set of bundles $B_1, ..., B_n$ are *order-invariant valid* if for every permutation $\sigma \in S_n$ we have that the state transition

$$\texttt{st} \longrightarrow \texttt{st} \circ B_{\sigma(1)} \circ ... \circ B_{\sigma(n)} \qquad (2)$$

is a valid state transition and is invariant among all the permutations. A set of bundles $B_1, ..., B_n$ *compete* in a state s if for all i and j, $\texttt{st} \rightarrow \texttt{st} \circ B_i \circ B_j$ is not a valid state transition.

Definition 2.7. We say that a bundle B is a partial extraction of a bundle B' if B and B' compete, and $\Delta b(P, \texttt{st} \circ B) < \Delta b(P', \texttt{st} \circ B')$.

The Flashbots combinatorial auction (FBCA) allows players to bid for bundles. The Flashbots allocation rule tries to solve the block optimization problem with conflicting constraints. That is, the block can not contain competing bundles (bundles that contain same transactions or bundles that revert). So, the FBCA can be modelled as the knapsack problem with a conflict graph $G = (V, E)$ [11], where V is the set of bundles, and $uv \in E$ if and only if bundles u and v compete. Flashbots use a greedy approximation algorithm, leading to an auction mechanism similar to a first-price sealed auction, since players can only observe winning bundles. That is, the bundles are ordered by effective gas price (or average gas price, see more details in the appendix A) and afterwards prune the conflicting bundles. In case of symmetric gas efficiencies, the MEV opportunity is sealed to the higher bidder and pays what they bid. In general, this algorithm does not give the optimal solution (see appendix C). It also allows searchers to check for relayer deviation with just the executed block and the bundle sent by the searcher. In other words, theoretically, searchers can privately monitor the correct functioning of the relayer.

Now we are ready to define the local MEV for a player P or MEV_P for short. The definition we provide is similar to the one provided in [3]. However, in [3], players have constraints on the state transitions, but not on the set of bundles that they can construct.

Definition 2.8. Let $\mathcal{D}$ be a domain with state $\texttt{st}$, a player P with local mempool view $\mathcal{T}_P^M$ and a set of transactions $\mathcal{T}_P$ that the player P can construct. We denote by $\mathcal{C}_P = \mathcal{T}_P^M \cup \mathcal{T}_P$ to be the set of reachable transactions. We define the *local* MEV *of P with state $\texttt{st}$* ($\text{MEV}_P(\texttt{st})$) as the solution to the following

optimization problem

$$\max_{B} \quad \Delta b(P; \texttt{st} \circ B, \texttt{st})$$
$$\text{s.t.} \quad B \subseteq \mathcal{C}_P,$$
$$\quad \texttt{st} \rightarrow \texttt{st} \circ B \text{ is a valid state transition in } \mathcal{D}$$

Let $\text{argmev}_P(\texttt{st})$ be the set of bundles that are a solution to the optimization problem[2]. The constraints of reachable bundles is subject to a player's information, gas efficiency, budget, ability to propose blocks, etc.

Observe that if all players have access to the same MEV opportunities and have access to all bundles, then this definition is equivalent to the one provided in [3]. If the mempool view is the empty set, we will refer to the MEV as *on top of block* MEV, and we will denote it by TMEV_P.

Lemma 2.9. For a given player P and a state $\texttt{st}$, if $B \in \text{argmev}_P(\texttt{st})$, then $\text{MEV}_P(\texttt{st} \circ B) = 0$.

Proof. Assume otherwise. Let $\texttt{st}$ and $B \in \text{argmev}_P(\texttt{st})$ such that $\text{MEV}_P(\texttt{st}) > 0$. Then, there exist B' such that $\Delta b(P, \texttt{st} \circ B \circ B', \texttt{st} \circ B) > 0$. Taking the bundle $B'' = B \cup B'$, we have that $\Delta(P, \texttt{st} \circ B'', \texttt{st}) > \text{MEV}_P(\texttt{st})$, that is a contradiction. $\square$

Definition 2.10. For a set of reachable bundles $\mathcal{C}$, we define the *$\mathcal{C}$-permissionless* MEV ($\text{MEV}^{\mathcal{C}}$) as the minimum local MEV that can be extracted among players that have access to the bundles in $\mathcal{C}$. More formally,

$$\text{MEV}^{\mathcal{C}}(\texttt{st}) \coloneqq \min_{P \text{ s.t } \mathcal{C} \subseteq \mathcal{T}_P} \text{MEV}_P(\texttt{st}).$$

If for all players P with reachable bundles $\mathcal{C}$, $\text{MEV}_P(\texttt{st}) = 0$, we say that $\texttt{st}$ is a null MEV state. We denote the set of null MEV states as NS.

Remark: Note that this definition is an extension of the definition made in [16] when $\mathcal{C}_K$ is the set of transactions that burns K coins. In other words, $\text{MEV}^{\mathcal{C}_K}$ permissionless is MEV that can be extracted by players with at least K coins. In this paper, we will assume that a finite number of players are able to capture a specific MEV opportunity. In other words,

---

[2]Observe that this definition fundamentally depends on the token balance. In the presence of a unique token, MEV_P is trivially defined. Nevertheless, in the presence of multiple domains and tokens this definition is non-trivial. Moreover, in this definition, we are assuming that all players value equally the tokens, assuming the existence of some transferable utility. We leave a more general definition of local MEV for future work.

we will fix a set of players $\mathcal{P} = \{P_1, ..., P_n\}$. For each $P \in \mathcal{P}$, we define $\mathtt{st}_i$ as the state that realizes the extraction MEV_P. Then, we will assume that for each player $P \in \mathcal{P}$, $\mathrm{MEV}_P(\mathtt{st}_j) = 0$ for all $P \in \mathcal{P}$.

3 The MEV stage game

In this section, we will consider an abstraction of the PGA model proposed in [5]. We will model and formalize the MEV stage game. In other words, we will formalize the game played for extracting a given MEV opportunity in a specific block. Afterwards, we will define the utility of the players as a function of their balance and the notion of strategy. Finally, we introduce a solution concept of the MEV stage game, the Sybil resistant Nash equilibrium.

We model the MEV game as a sequential game among a set of n searchers $\mathcal{P} = \{P_1, ..., P_n\}$ who can send bundles to obtain an MEV opportunity. We will assume that all players compete for the same MEV opportunity and have sufficient capital to extract it. When a specific player wins the MEV opportunity, it reaches a null MEV state for all players. In the following, we will provide a list of points that will define the MEV stage game. This game will take into account the latency of the players, the duration of the blocks, the mechanism of transaction inclusion, and the costs of improving software, node location, etc.

1. **Continuous time**: Searchers act in continuous time rather than discrete rounds (as in typical extensive-form games). That is, at any moment in time, players can take an action that, in our case study, will be sending a bundle.

2. **Local MEV**: At time t, each player P_i finds an MEV opportunity of value $v_i(t) \sim V_i(t)$, (in general, such that $v_i(t) \geq v_i(s)$ for all $t \geq s$, with $\{V_i(t) : t \geq 0\}$ being a family of distributions. More specifically, for each time t, the player finds a bundle B such that $\Delta b(\mathtt{st} \circ B, \mathtt{st}) = v_i(t)$.

3. **Latency**: Searchers can see each other's actions, but not immediately, due to the latency in the peer-to-peer network. The latency is modeled by a directed weighted graph $G = (E, V)$. Each searcher controls a set of nodes $N_i \subseteq V$. So, if P_j sends a bundle from a subset of peers $L_j \subseteq N_j$ at time t_j, P_i observe the bundle at time $t = t_j + d(L_j, N_i)$, where d is the non-symmetric distance induced by the weight.

4. **Probabilistic auction duration**: The auction terminates at a randomly drawn time when a

new block is mined. We model the block interval as a positive random variable $\mathcal{B}$.

5. **Competitors information**: Players do not necessarily know the number of competitors and their features. However, each player estimates the number of competitors and their behavior.

6. **Access to a public correlating device**: Let (Ω, Pr) be a probability space. All searchers observe the first drawn $w \in [0, 1]$ of a uniform public random variable X (a beacon or the hash of the previous block). This can be used by players to coordinate their actions.

7. **Auction Mechanism**: Sequencers have a predefined algorithm that inputs a set of bundles and outputs an order of transactions for inclusion in a block. We will denote this algorithm by the *ordering mechanism*. This ordering mechanism and the characteristics of the MEV opportunity (Back-running, Front-running, sandwich,...) determines the revenue for each player. The ordering mechanism and the MEV opportunity determinate an auction $\mathbb{A} = (\mathbf{x}, \mathbf{pr})$, that is, a pair of maps that take as inputs the set of bundles and a random event and outputs a winner and the payment induced per each player. More formally,

$$\mathbf{x} : \mathtt{View}_s \times \Omega \to \{x \in \{0,1\}^n : x \cdot \mathbf{1}^T \leq 1\}$$
$$\mathbf{pr} : \mathtt{View}_s \times \Omega \to \mathbb{R}^n.$$

where $\mathtt{View}_s$ is the set of transactions seen by the sequencer when constructing the block.

8. **External costs**: Players can improve their mempool view, reduce their latency, and improve their software to increase their local MEV. The set of external actions that a player P_i can take rely on a set of actions A_i, and the costs of taking those actions are modeled by a function $c_i : \mathrm{A}_i \to \mathbb{R}$.

Definition 3.1. Given a set of players $\mathcal{P}$, a random positive variable $\mathcal{B}$, a gossip network graph G and an auction mechanism, $\mathbb{A} = (\mathbf{x}, \mathbf{pr})$, an MEV game is $\mathbb{G} = (\mathcal{P}, G, \mathcal{B}, \mathbb{A}, (c_i)_{i=1,..,n})$. We say the game is symmetric if all players share the same features. That is, $V_i = V_j$, $c_i = c_j$ for all i, j, $\mathbb{A}$ are symmetric functions and G is a homogeneous graph.

The MEV game has a structure of sequential continuous game, and therefore we can define utilities, strategies, and solution concepts such as Nash equilibrium.

Definition 3.2. A strategy S_i is a procedure for participating in the MEV stage game and may be probabilistic. S_i takes the following form, for a current time t and a local view $\mathtt{view}_i(t)$ of the player i: $(a, \mathtt{view}_i'(t)) \leftarrow S_i(t, \mathtt{view}_i(t))$. The output a is the action taken by P_i and is bounded by domain constraints. The output $\mathtt{view}_i'(t)$ is the updated state, that is, $\mathtt{view}_i'(t) = \mathtt{view}_i(t) \cup \{a\}$. A strategy of a player i is non-adaptive if it does not depend on the local view $\mathtt{view}_i(t)$. More formally, for every t and every pair $\mathtt{view}_i(t), \mathtt{view}_i'(t) \in \mathtt{View}_i(t)$, it holds $S_i(t, \mathtt{view}_i(t)) = S_i(t, \mathtt{view}_i'(t))$.

Definition 3.3. Let $S = (S_1, ..., S_n)$ be a strategy tuple, then the *expected payoff of the player P_i* is

$$u_i(S_i, S_{-i}) := \mathbb{E}[\Delta b_i \mid S],$$

where S_{-i} is an $n-1$ tuple without the ith coordinate. We denote by $\mathbb{S}_i$ the set of all strategies.

That is, $u_i(S_i, S_{-i})$ is the expected payoff of player i if they execute the strategy S_i, the other players execute the strategy S_{-i} in a domain $\mathcal{D}$. Notice that we assume players have a monotone risk-neutral utility over the balances.

Definition 3.4. Let $\mathbb{G}$ be an MEV stage game, we say that a tuple of strategies $(S_1, ..., S_n)$ is a *Nash equilibrium* (NE) if for each player P_i

$$u_i(S_i, S_{-i}) \geq u_i(\tilde{S}_i, S_{-i}), \text{ for all strategies } \tilde{S}_i.$$

In other words, if players are taking the strategies $(S_1, ..., S_n)$, none of them have incentives to deviate unilaterally. We denote $\mathrm{NE}(\mathbb{G})$ the set of all Nash equilibrium.

However, the notion of Nash equilibrium is not strong in permissionless environments. For example, the equilibrium of firms competing in the Cournot price model [14] is weak against Sybil attacks. Another example is the equilibrium constructed in the theorem of Flashboys 2.0 [5] that proves that exponential raise bidding strategies with grim-trigger area a Nash equilibrium in the game with two players. Nevertheless, one can prove that agents have incentives to use Sybil attacks to maximize their payoff. For this reason, we introduce a stronger notion of a solution concept for MEV games that are resistant to Sybil attacks:

Definition 3.5. Let $\mathbb{G}$ be a symmetric[3] MEV game with n players, and let $\phi : \mathbb{N} \to \bigoplus_{i=1}^{\infty} \mathbb{S}_i$ be a strategy mapping. A Sybil resistant Nash equilibrium is ϕ such that for all $n \in \mathbb{N}$, $\phi(n) \in \bigoplus_{i=1}^{n} \mathbb{S}_i$, and $\phi(n)$ is

---

a Nash equilibrium, where for each $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, n+1\} - \{i\}$,

$$u_i(\phi(n)_i, \phi(n)_{-i}) \geq u_i(\phi(n+1)_i, \phi(n+1)_{-i}) \\ + u_j(\phi(n+1)_j, \phi(n+1)_{-j}).$$

The notion of Sybil resistance is important in permissionless pseudo-anonymous environments such as blockchains. Players have the ability to generate additional addresses to take more profits from cooperative strategies. In future work, we will prove that Sybil resistant cooperative Nash equilibrium exist in the priority gas auction in the non-repeated games, and the existence of Sybil resistant equilibrium in games with private mempools.

4 Price of MEV

MEV games have an important impact on users, network congestion, computation overload, and blockchain liveness. On one hand, some MEV opportunities arise from value extracted from users. On the other hand, in general, MEV games induce an inefficient extraction of MEV opportunities, leading to network congestion (e.g. P2P network load), and chain congestion (e.g. block-space usage). In this paper, we will not take into account the negative externalities of MEV on individual users (for example, sandwich attacks and oracle manipulation), but rather the negative externalities that impact the consensus protocol, liveness, chain quality, stake distribution, etc. Suppose there is an MEV opportunity on a state $\mathtt{st}$, with a set of n players that compete to extract it, sending bundles $B_1, ..., B_k$. Then, a sequencer, using the ordering mechanism **or**, outputs a block (this order mechanism does not necessarily respect the internal order of the bundles). In this setting, we define the *block space cost* as the gas cost of executing the block built by the ordering mechanism using the bundles, more formally

$$C(B_1, ..., B_k; \mathtt{st}) = \mathrm{gasUsed}\left(\mathtt{st} \circ \mathbf{or}(\cup_{i=1}^{k} B_i)\right). \quad (3)$$

We naturally extend the function of cost over strategies by taking the outcome (or the expected outcomes in case of mixed strategies) of the strategies (the broadcasted bundles). Clearly, ex-post it is trivial to compute the gas cost. However, ex-ante, estimating the gas costs induced by the MEV game is much more complex due to the non-commutative nature of execution costs. Also, other cost functions can be very relevant in some domains. Different MEV games can induce other types of negative externalities, such as

---

[3]This definition can be extended to non-symmetric games, but we will leave it for future work.

wasted resources induced by computation costs[4] or centralization effects.

In general, self-interested behaviour by strategic players leads to an inefficient result, an outcome that could be improved upon given centralized control over everyone's action [12]. Nevertheless, imposing such control can be costly, infeasible or undesirable (due to trust assumptions). This motivates the search for conditions and mechanisms in which decentralized optimization by strategic agents is guaranteed to produce a near-optimal outcome. The price of anarchy (PoA) [13] is a measure that quantifies how far is the worst Nash equilibrium (in the sense of social cost) with respect to any optimal configuration that minimizes the social cost. More formally, given a cost function C and the set of Nash equilibrium NE, the price of anarchy is defined as:

$$\text{PoA} = \frac{\max_{S \in NE} C(S)}{\min_S C(S)}.$$

Different examples of the study of the price of anarchy can be seen in [4, 12, 13]. However, the price of anarchy in the MEV game is, in general, not well-defined. For example, assume that two players are competing for extracting the same arbitrage opportunity. Then, as we will see, the block space cost of extracting the arbitrage opportunity will be the sum of gas used by executing both searchers' transactions. However, the minimal cost of the game is zero, since not extracting the MEV opportunity is a feasible outcome. Therefore, the ratio is not defined, leaving an inconsistent definition of the price of anarchy in the MEV game. In the following, we propose a small adjustment to have a well-defined price of anarchy in the MEV game, which we denote as the Price of MEV. This measure tracks the social costs induced by the competition among individually rational agents for MEV extraction in a particular MEV game. More precisely, the Price of MEV is a family of measures parametrized by the social cost functions. This family of measurements can be useful to compare the negative externalities and trade-offs of different MEV games. Similar to the price of anarchy definition, the price of MEV of game $\mathbb{G}$ with social cost C is the ratio of the worst Nash equilibrium with respect to the extraction made by the most efficient player in an order-free consensus protocol blockchain.

Definition 4.1. Given a cost function C, the set of Sybil resistant Nash equilibrium $\text{SNE}(\mathbb{G})$, and the set of actions that induce a null MEV state NS, we define the price of MEV as:

$$\text{PoMEV}(\mathbb{G}, n) = \frac{\max_{S \in SNE(\mathbb{G})} C(S)}{\min_{S \in \text{NS}} C(S)},$$

where $\min_{a \in \text{NS}} C(a)$ is the minimum cost taken by extracting the MEV opportunity.

We argue that a more efficient mechanism to extract MEV opportunities (low Price of MEV) can have an important impact on blockchain stability, users utility, and consensus protocol[5]. In the case where the cost function is defined over pure strategies, we can extend the definition naturally over the mixed strategies, taking the expectancy of the outcomes.

5 Conclusions and Future Work

In this work, we proposed measures to formally study the Nash equilibrium and negative externalities of different ordering mechanisms. We think that this is import due to the future changes on Ethereum mainnet about Proposer-Builder separation and MEV-boost [7]. We leave for future work to study and compute the price of MEV of popular proposed ordering mechanisms. Also, in the future we will study the negative externalities of zero-sum MEV opportunities such as sandwich attacks, time bandit attacks, eclipse attacks, draining bot attacks[6], and censorship attacks. We conjecture that a domain with suboptimal block-space market design will lead to more block-space misuse (high price of MEV), raising the transactions fees of the underlying domain. The higher transactions fees will increase the direct payments of the miner (another form of MEV) and the market inefficiencies. This market inefficient will imply inefficient price discover, creating more internal and cross chain MEV opportunities. We call this effect the Circular forces of MEV 1.

References

[1] Yalçın Akçay, Haijun Li, and Susan H Xu. "Greedy algorithm for the general multidimensional knapsack problem". In: *Annals of Operations Research* 150.1 (2007), pp. 17–29.

[2] Guillermo Angeris, Alex Evans, and Tarun Chitra. "A Note on Bundle Profit Maximization". In: Stanford University, 2021.

---

[4]For example, the energetic costs induced by computing TH/s in blockchains that order transactions by nonce. See BSC-PR for more details.

[5]Note that we are assuming that the extractable value exists and will be extracted.
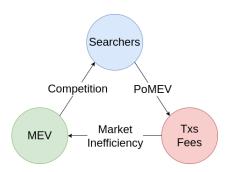
[6]https://github.com/Defi-Cartel/salmonella

Figure 1: Circular forces of MEV

[3] Kushal Babel et al. "Clockwork finance: Automated analysis of economic security in smart contracts". In: *arXiv preprint arXiv:2109.04347* (2021).

[4] George Christodoulou and Elias Koutsoupias. "The price of anarchy of finite congestion games". In: *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing.* 2005, pp. 67–73.

[5] Philip Daian et al. "Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability". In: *2020 IEEE Symposium on Security and Privacy (SP).* IEEE. 2020, pp. 910–927.

[6] Ethereum. "Sort transactions at the same gas price by received time". In: 2020.

[7] Flashbots. *MEV-Boost.* 2022. URL: https://github.com/flashbots/mev-boost.

[8] Mahimna Kelkar et al. "Order-fairness for byzantine consensus". In: *Annual International Cryptology Conference.* Springer. 2020, pp. 451–480.

[9] Kshitij Kulkarni, Theo Diamandis, Tarun Chitra, et al. "Towards a Theory of Maximal Extractable Value I: Constant Function Market Makers". In: *arXiv preprint arXiv:2207.11835* (2022).

[10] Alexandre Obadia et al. "Unity is Strength: A Formalization of Cross-Domain Maximal Extractable Value". In: *arXiv preprint arXiv:2112.01472* (2021).

[11] Ulrich Pferschy and Joachim Schauer. "The Knapsack Problem with Conflict Graphs." In: *J. Graph Algorithms Appl.* 13.2 (2009), pp. 233–249.

[12] Tim Roughgarden. "Intrinsic robustness of the price of anarchy". In: *Journal of the ACM (JACM)* 62.5 (2015), pp. 1–42.

[13] Tim Roughgarden. *Selfish routing and the price of anarchy.* MIT press, 2005.

[14] Roy J Ruffin. "Cournot oligopoly and competitive behaviour". In: *The Review of Economic Studies* 38.4 (1971), pp. 493–502.

[15] Harvey M Salkin and Cornelis A De Kluyver. "The knapsack problem: a survey". In: *Naval Research Logistics Quarterly* 22.1 (1975), pp. 127–144.

[16] Alejo Salles. "On the Formalization of MEV". In: *https://writings.flashbots.net/research/formalization-mev/* (2021).

[17] Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. "Optimal selfish mining strategies in bitcoin". In: *International Conference on Financial Cryptography and Data Security.* Springer. 2016, pp. 515–532.

[18] Ben Weintraub et al. "A Flash (bot) in the Pan: Measuring Maximal Extractable Value in Private Pools". In: *arXiv preprint arXiv:2206.04185* (2022).

A Examples of ordering mechanisms

For the sake of completeness, we will provide a list of examples of ordering mechanisms. They are induced in different domain and consensus protocol designs.

- **Priority gas ordering mechanism** : Sequencers try to solve the KEV by using the greedy approximation algorithm that consists of ordering the transactions by gas price. In this case, if a player is trying to capture an MEV opportunity, it must monitor the mempool and choose an optimal gas price m. If a player is trying to front-run a transaction tx, with gas price m, it is enough to outbid it with $m + \eta$. In this setting, if the gas cost of exploiting this MEV opportunity is g, the block-space price of MEV of the uni-agent game is 1.

- **Flashbots mechanism**: Searchers send bundles to the relayer through a private channel. The Flashbots relayer tries to build the block with the highest profits among all the blocks that can be constructed using the transactions in the public mempool and the Flashbots mempool of bundles. But the bundles have a number of allocation constraints that the Flashbots relayer must account for [16]. In order to build the block with the highest profit, Flashbots (to our knowledge) uses a greedy approximation algorithm. As described in the Flashbots documentation, a bundle B is ordered by effective

gas price / bundle score, which is defined as,

$$sc(B) := \frac{\Delta_{coinbase} + \sum_{tx \in B \smallsetminus \mathcal{TX}} g(tx)m(tx)}{\sum_{tx \in B} g(tx)}$$

where $\Delta_{coinbase}$ denotes the direct payment to the miner, $\mathcal{TX}$ is the set of mempool transactions, $g(tx)$ is the gas used by tx and $m(tx)$ is the gas price of tx.

- **Random ordering mechanism**: The transactions included in the next block and the order of transaction execution are probabilistic with a uniform distribution.

- **First input first output mechanism**: The transactions are ordered by the sequencer local's timestamps or by a pseudo-global timestamp such as the one mentioned in [8]. In this sense, players with better geolocation and propagation algorithms will win the MEV game. However, in decentralized systems, this will depend on the leader geolocation that will change randomly per round.

- **Dictatorship/Permissioned mechanism**: The sequencer has its own arbitrary ordering rule, prioritizing transactions of a fixed set of addresses. In this setting, players do not have a lot of freedom to interact or win the MEV opportunity. In other words, the sequencer will censor other players' transactions to prioritize its own extraction. Moreover, this potentially will induce inefficient market prices. However, the block-space price of anarchy is minimized since just one player is extracting it. This rule also models the situation where the miner captures the MEV opportunity, prioritizing its own profitable bundles.

- **Metadata mechanism**: Let $(\{0,1\}^n, \le)$ be a total ordered set. Transactions and bundles can add a parameter `nonce`, giving them an associated hash identification. Then the bundles and transactions are ordered by hashes. For example, if a transaction `tx` with `nonce` tries to extract a back-running arbitrage opportunity, then a player will try to produce a transaction that extracts the opportunity nonce `nonce' < nonce`.

B   Bot example

Before the pull request [6], transactions with the same gas price were randomly ordered, creating incentives

	PGA	FSS	R.O.	Perms.	MP
Ethereum (Geth)	✓	✗/~	✗	✓	✗
Polygon	✗	✗	✓	✓	✗
BSC	✓	✗	✗	✓	✗
Avalanche	✗	✓	✗	✗	✓
Arbitrum	✗	✓	✗	✓	✓
Shutter Network	✓	✗	✗	✓	✓
Solana	✗	✓	✗	✓	✗
Flashbots (alpha-v0.6)	✓	✗	✗/	✓	✓

Table 1: MEV games features different chains. Perms= Permissionless and MP = Mempool Privacy.

for searchers to spam transactions to capture an MEV opportunity. In the following, we will show how a particular MEV bot that captured back running opportunities, was responsible for consuming unnecessary block space to increase its expected revenue. A lot of examples can be seen using tools such as `mev-inspect-py`. Moreover, we could lower bound the estimated block space price of MEV by ≈ 7.
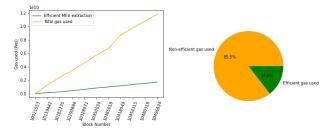

Figure 2: Total gas used and efficient gas used by bot MEV Bot 0x00...E9a

C   Flashbots counter-example

Let $\mathrm{FBR}(B_1, ..., B_n)$ be the revenue using the Flashbots greedy approximation algorithm. Let $\mathrm{OTP}(B_1, ..., B_n)$ be the maximal revenue of the Flashbots combinatorial problem.

Claim: The Flashbots combinatorial auction is not optimal. More specifically,

$$\inf\{\frac{\mathrm{FBR}(B_1, ..., B_k)}{\mathrm{OPT}(B_1, ..., B_k)} : \text{ for } B_1, ..., B_k \text{ bundles}\} \le \frac{1}{\lfloor \frac{L}{g_{min}} \rfloor - 1},$$

where g_{min} is the minimal gas consumed by competing bundles and L is the gas limit of a block.

Proof: Let $B_1, ..., B_k$ all the bundles such that, B_1 compete with B_i for all $i \ne 1$ and B_i, B_j are pairwise non-competing bundles. Moreover, assume that B_1

has gas costs L/k and effective gas bid $m + \varepsilon$ for $\varepsilon > 0$ and all the other bundles have gas bid m. Then, the Flashbots algorithm outputs $B = \{B_1\}$, leaving to a sequencers' revenue of $m + \varepsilon$. On the other hand, the optimal valid block is $B = \{B_2, ..., B_k\}$ with $m(k-1)$ revenue. The result follows using bundles with gas cost g_{min}.