MEV games(Work in progress)

Bruno Mazorra
brunomazorra@gmail.com
UPF and Nokia Bell-labs

Michael Reynolds
mireynolds@pm.me
Nethermind

September 11, 2022

Abstract

Miner or Maximal extractable value known as MEV usually refers to the extra value that *privileged* players can extract through strategically order, censor and place transactions in a domain. While every domain has its own consensus, ordering and block-creation mechanisms, each one of them arise different optimal strategies to extract the value. This strategical behavior played by rational players (searchers) arise have different impact and externalities in each domain. In this paper, we will formalize the games that arises from common knowledge MEV opportunities. First, we introduce the MEV game, the game played by searchers to extract the MEV. Moreover, to study the negative externalities of the MEV games, we define the price of MEV (the price of anarchy of MEV games) with different cost functions given special emphasis to the block-space misusage. We compute the Nash equilibrium for these games and their respective block space price of MEV. We use real data of different domains such as Ethereum, Polygon and Binance smart chain to test the theoretical results obtained and compute the estimated gas misused in the MEV games. Furthermore, we introduce the Fairy tale/tail strategies, the category of negative sum MEV games induced by bots to abuse other bots bad design. These strategies can lead to different consequences. The most common one is draining other bots funds. However, other more involved strategies can lead to use the bots funds to DDoS the network without costs for the attacker. Finally, we introduce a more abstract definition of MEV without assuming symmetric monotone utilities over players balance. We claim that this definition provides a more insightful way of understanding the existence and the arising of MEV opportunities in decentralized domains. Providing a better framework for MEV mitigation solutions.

Contents

1 Introduction

The notion of miner-extractable value (MEV) introduced in [6], and formally defined in [3], measures the extra value that miners can extract through strategically order, censor and place transactions. In general, validators/miners have the power to dictate or influence to include and order the pending transactions in each block. Thus, rational validators have access to extra rewards per block. However, the power of partially reordering and including transactions is not limited to miners. As shown in [6], different users and bots, usually denominated *searchers*, can spam transactions or outbid competitors to increase their chances to extract value. For this reason, the term miner-extractable value has progressively changed to *maximal-extractable value*, formally defined on [20], not limiting the value extraction to miners/validators.

In different consensus protocol settings and MEV opportunities, rational searchers have incentives to take different actions to increase their expected revenue. In "high" latency chains like Ethereum, it is observed empirically [6] that searchers have interest to increase their mempool view to outbid their competitors. While in low latency chains like Avalanche, searchers have incentives to reduce their latency. However, until the date, no game theory model have been proposed to explain the incentive vectors of searchers in different chains.

In Flashboys 2.0. [6] P. Daian et al. proposed a formalization of the price gas auction (PGA) model and proved mathematically and empirically the existence of Grim-Trigger Nash equilibrium under certain constraints. However, the authors prove the results for two players and do not take into that the cost of reverted transactions is proportional to the bid, that players can generate Sybil attacks and more general ordering mechanisms.

The first goal of this paper is to review different auction and MEV games played in different environment conditions (latency of players, fair sequencing service, PoW/PoS, private mempool, first-price auction,...). To do so, we will assume that the auctions are played by a finite set of players one time, that is, we will assume that the game is not played indefinitely. In this setting, we will find and characterize the different Nash equilibrium, allowing us to compute the expected revenue of miners and searchers for different MEV-games.

Further, we initially focus on a proper game theory formalization of latency, gas optimization, extractable value optimization, and action space of the games. We called this formalization the MEV games. With this formalization, we find and characterize the Nash equilibrium of MEV auction games, providing data and mathematical proofs to show the results obtained. We prove that in each game, the players will have different strategies such as spamming, cartelizing, censoring, and/or improving geographic localization to maximize their payoff.

Following, we provide another set of MEV games, that we called the zero-sum MEV games. Games played by searchers to punish other searchers, inefficient bots to extract value. We provide different zero-sum strategies. Finally, we introduce attacks to builders to make them construct more inefficient bundles.

Organization of the paper: In Section 2 we formalize the notion of bundle, extractable value and maximal extractable value. Moreover, we informally introduce the notion of MEV game, strategy, the solution concept of Nash equilibrium, the negative externalities and the price of MEV, the price of anarchy applied to MEV games with different cost functions. In Section 3, we formalize the MEV game and make a taxonomy of different chains by different MEV games characteristics. In Section ref, we introduce the MEV game induced by the priority gas allocation rule, and we study the Nash equilibrium and price of anarchy in different domains. In Section ref we study the game induced by the Flashbots allocation rule, its Nash equilibrium and its price of anarchy.

Our contributions. We make the following contributions:

- Similar to [3] but with fundamental differences, we provide a formal definition MEV and local MEV as an optimization problem limited to the player resources. In our definition, we emphasize the relative constraints, such as capital, software resources and mempool view for each player.
- We formally define a sufficiently abstract and general MEV game that allows studying the incentives of searchers in different domains. Moreover, we study specific MEV games with different ordering mechanisms such as: front-running priority gas auction (PGA), back-running PGA, private mempool PGA,

random ordering, latency games and Flashbots auction for common knowledge MEV opportunities. For these games, we compute and characterize the Nash equilibrium.

- We introduce different negative externalities that can arise on the MEV games. To study the negative externalities, we formally define the Price of Anarchy in the MEV games for different costs function. For the block space misuse, we call it price of MEV. We compute the price of MEV that arise from different MEV games, and we use it to characterize and classify different ordering mechanisms.
- We formally introduce the MEV-boost as an MEV game by extending the MEV abstract game. The formalization provides a framework, to compare and predict different outcomes when builders compete for block construction.
- We provide data of Ethereum and polygon through different stages, where the rules of MEV extraction were completely different.
- We introduce the *fairy tail* strategies. A set of strategies, like salmonella, that tricks other bad design MEV bots to capture fake MEV opportunities to manipulate them in a specific behavior. These strategies can be used for a wild range of purposes. Like DDoS in the network, or stealing their funds.
- Finally, we introduce a vector attack to the Flashbots builder in v0.6. on Proof of Work.

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 [6], formally defined in [3], and extended in cross-domain environments in [13]. Another similar definition was given in [20], 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 [13], 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 $\mathtt{st}$. 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 $\mathtt{prt}(\mathcal{D})$.

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 $\mathtt{st}$, the update of the state $\mathtt{st}$ after executing transactions $\mathtt{tx}$ is given by $\mathtt{st} \circ \mathtt{tx}$. For an ordered set of transactions $B = \{\mathtt{tx}_1, ..., \mathtt{tx}_l\}$, we have the composition $\mathtt{st} \circ B = \mathtt{st} \circ \mathtt{tx}_1 \circ .... \circ \mathtt{tx}_l$.

Similar to [3], we use $\mathtt{Addr}$ 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 $a \in \mathtt{Addr}$ 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 $T \subseteq \mathcal{T}$.

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 $\texttt{prt}(\mathcal{D})$. 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 [21] 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 $\texttt{tx}_1, ..., \texttt{tx}_n$ with gas price $m_1, ..., m_n$ and $g_1, ..., g_n$ units of gas. If the sequencer includes $\texttt{tx}_i$, 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 $\sum_{i:\texttt{tx}_i \in \mathcal{T}} g_i \leq L$. 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 $\text{KEV}(\mathcal{T})$ the solutions' revenue of the optimization problem. Knapsack optimization problems [19] 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 4.5) 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 $\mathcal{T} = \{\texttt{tx}_1, ..., \texttt{tx}_k\}$ and a state $\texttt{st}$ 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:

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

[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.

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 $\mathbf{pr}(B)$, 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

$$\mathtt{st} \longrightarrow \mathtt{st} \circ B_{\sigma(1)} \circ ... \circ B_{\sigma(n)} \tag{2}$$

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, $\mathtt{st} \to \mathtt{st} \circ B_i \circ B_j$ 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 $\Delta b(P, \mathtt{st} \circ B) < \Delta b(P', \mathtt{st} \circ B')$.

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 $G = (V, E)$ [14], 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 effective gas price, see more details in the appendix 4.5) 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 4.5). 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 $\mathtt{st}$, 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* $\mathtt{st}$ ($\text{MEV}_P(\mathtt{st})$) as the solution to the following optimization problem

$$\begin{aligned} \max_{B} \quad & \Delta b(P; \mathtt{st} \circ B, \mathtt{st}) \\ \text{s.t.} \quad & B \subseteq \mathcal{C}_P, \\ & \mathtt{st} \to \mathtt{st} \circ B \text{ is a valid state transition in } \mathcal{D} \end{aligned}$$

Let $\text{argmev}_P(\mathtt{st})$ 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 $\mathtt{st}$, if $B \in \text{argmev}_P(\mathtt{st})$, then $\text{MEV}_P(\mathtt{st} \circ B) = 0$.

[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.

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

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,

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

If for all players P with reachable bundles $\mathcal{C}$, $\text{MEV}_P(\mathtt{st}) = 0$, we say that $\mathtt{st}$ 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 [20] 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, 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 $P \in \mathcal{P}$, $\text{MEV}_P(\mathtt{st}_j) = 0$ for all $P \in \mathcal{P}$.

2.1 Game theory preliminars

Informal definition: A MEV stage game is a continuous time game with a set of players $\{1, ..., n\}$, that can take actions in a set A_i, with imperfect information and probabilistic time duration. That is, for each time $t \in [0, \infty)$, players take an action $a_i(t) \in A_i$. At each time t, players have information set $\mathtt{view}_i(t)$. The game ends in probabilistic time modeled by a random variable X. Each player has its own utility function u_i that takes as input the actions taken by all players until the end of the game and outputs a real number.

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 2.11. 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 2.12. 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 2.13. 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 $\text{NE}(\mathbb{G})$ 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 [17] is weak against Sybil attacks. Another example is the equilibrium constructed in the theorem of Flashboys 2.0 [6] 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 2.14. 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 $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, n+1\} - \{i\}$,

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

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.

2.2 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

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

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 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 [15]. 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) [16] 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 [5, 15, 16]. 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

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

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

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 2.15. 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.

3 Formalization of MEV games

In this section, we will consider an abstraction of the PGA model proposed in [6]. 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 $\Delta b(\texttt{st} \circ B, \texttt{st}) = v_i(t)$.
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.

[5]Note that we are assuming that the extractable value exists and will be extracted.

6. **Access to a public correlating device**: Let $(\Omega, \Pr)$ 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,

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

   where $\texttt{View}_s$ 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.

Remark: The differences between the model presented in [6] and this model is given by points ii),iii),2) and 6). The properties 3) and 5) are important changes that impact the Nash equilibrium of [6] if the allocation rule is assumed to be the PGA, but, as we will see in this section, the equilibrium maintain a similar Grim-trigger strategy. However, if we modify the point 5), and we do not assume that players know n, the equilibrium mentioned in [6] is no longer a Nash equilibrium since breaks under Sybil strategies, leading to a fundamentally different game. Another example is the following: Assume that 2 players are competing for an arbitrage opportunity of 100\$. If both players do not know that there exist another searcher extracting the same opportunity, then they maximize their ex ante payoff bidding $b_{min} = 10\$$. And ex post, they will obtain a payoff of $\frac{1}{2}90\$ = 45\$$ on average. However, if the searchers have information of the existence of a competitor, the ex ante and ex post payoff in the Flashbots auction is 0\$ by proposition 6.8, where we prove that there is a unique Nash equilibrium with null profits. Modelling the information and how do players react to the information given have a crucial impact on the equilibrium of the game. In general, in this paper, we will assume perfect information and will leave for future work the Bayesian game. On the other hand, changes in the allocation rule can change completely the dominating strategies played by searcher on the uni-agent and multi-agents MEV game.

Ordering mechanism examples: In the following, we will provide a list of examples. Some of the following ordering mechanism are induced in different domains nodes and consensus protocols designs.

- **Priority gas ordering mechanism** : Sequencers try to solve the KPEV 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. In the case a player is trying to front-run a transaction $\texttt{tx}$, with gas price m, is enough to outbid it with bid $m + \eta$. In this setting, if the gas cost of exploiting this MEV opportunity is g, then 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. Flashbots relayer try to build the block with the highest profits among all the block 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 which the Flasbhots relayer must be accounted for [20]. In order to build the block with the highest profit, Flashbots (to our knowledge) uses a greedy approximation

algorithm, ordering the bundles by average gas price or as the Flashbots team described in their docs, the score of the bundle. The score of a bundle B is defined as

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

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 appended to the next block and the order of transaction execution are chosen uniformly random. For example, assume that a transaction `tx` induces an MEV opportunity of value `ev` if a specific transaction B is placed afterwards (back-running). Assume that the gas costs of the transaction B is γ if the transaction does not capture the opportunity, and g is the gas cost if the transaction captures the MEV opportunity. The min gas fee for inclusion is m. Let $\mathbb{E}(\Delta b \mid \#\text{tx} = x)$ be the expected revenue of the player if it sends x transactions. Then, the number of transactions x that the player must send in order to maximize its profit is:

$$\begin{aligned}\text{argmax}_{x\in\mathbb{N}}\mathbb{E}(\Delta b \mid \#\text{tx} = x) &= \text{argmax}_{x\in\mathbb{N}}\left(1 - \frac{1}{x+1}\right)(\texttt{ev} - gm - (x-1)\gamma m) - \frac{1}{x+1}x\gamma m \\ &\approx \sqrt{\frac{\texttt{ev}}{\gamma m} - \frac{g}{\gamma}} - 1 - 1 = \sqrt{\frac{\text{net profit}}{\text{tx fail cost}}} - 1 - 1\end{aligned}$$

  Therefore, the block usage price-of-anarchy in the uni-agent game is $\Omega\left(\frac{\gamma}{g}\sqrt{\frac{\texttt{ev}}{\gamma m} - \frac{g}{\gamma}} - 1\right)$.

- **Fair sequencing service mechanism**: The transactions are ordered by the sequencer local's timestamps or by a pseudo-global timestamp such as the one mentioned in [9]. In this sense, players with better location and propagation algorithms will win the MEV game. However, in decentralized systems this will depend on the leader location that will change randomly per round.

- **Dictatorship/Permissioned mechanism**: The sequencer have 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 to 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 $(\{0,1\}^n, \leq)$ a total ordered set. Transactions and bundles can add a parameter `nonce`. Then each bundle or transaction has an associated hash identification. Then the bundles and transactions are ordered by hashes. For example, if a transaction `tx` with nonce `nonce` back-running arbitrage opportunity, then a player will try to produce a transaction that extracts the opportunity with nonce `nonce' < nonce`.

4 MEV games

4.1 Front-running PGA

In this section, all searchers are assumed to have unique parameters Δ_i, γ_i and g_i unless otherwise stated. However, we will assume that all searchers find the MEV opportunity at the beginning of the auction. That is, $v_i(t) = \texttt{ev}$ for all $t \geq 0$. We let $B_i = (t_i, m_i; i)$ denote a EV profitable bundle of a searcher i. Here t_i is the time at which the bid is placed, m_i is the gas price, and i is the identity of the bidder. By the FPGA model, a player j, sees a bundle $B = (t, m; i)$ at time $t_i + \Delta_j$. We denote by $\texttt{view}_i(t)$ the set of all bundles seen by the searcher i at time t (including their own bundle) and the state of the auction (active or finished). We denote by $\texttt{View}_i(t)$ the set of all possible views at time t for a searcher i. If $\Delta = \mathcal{B} + \varepsilon$, then searchers do not have any information of each other's bundles, therefore $\texttt{View}_i(t) \cap \texttt{View}_j(t) = \varnothing$ for all t.

Definition 4.1. A strategy S_i is a procedure for participating in the FPGA, and may be probabilistic. S_i takes the following form, for a current time t and a local view $\texttt{view}_i(t)$ of the player i:

$$(a, \texttt{view}'_i(t)) \leftarrow_{\$} S_i(t, \texttt{view}_i(t)).$$

The output a is the action taken by $\mathcal{P}_i$. Either $a = (t, b; i)$ with $m > m_{min}$ or else $a = (t, 0; i)$ indicating that the searcher is not placing any bid. The output $\texttt{view'}_i(t)$ is the updated state, that is, $\texttt{view'}_i(t) = \texttt{view}_i(t) \cup \{a\}$. A strategy of a searcher i is non-adaptive if it does not depend on the local view $\texttt{view}_i(t)$. More formally, for every pair t and every pair $\texttt{view}_i(t), \texttt{view}'_i(t) \in \texttt{View}_i(t)$, it holds $S_i(t, \texttt{view}_i(t)) = S_i(t, \texttt{view}'_i(t))$.

A basic example of non-adaptive pure strategy is the blind raising strategy described in [6]:

Blind raising S_{BR}: is a non-adaptive strategy, where a searcher raises their own bids on a predetermined schedule, independently of actions by other searchers. A player $\mathcal{P}_i$ follows the blind raising strategy if at $t = 0$, they emit a bid with gas price m_0 and for $t = k\delta$ with $k \in \mathbb{N}$ emits $m_0 \times (1 + \tau)^k$.

On the other hand, an example of adaptive pure strategy is the counterbidding strategy, also mentioned in [6].

Reactive pure counterbidding S_{RC}**:** Is a strategy in which a searcher observes an opponent's bidding and reacts by placing higher opposing bids. More formally, a searcher $\mathcal{P}_i$ uses the reactive pure counterbidding strategy if:

- at time $t = 0$, $\mathcal{P}_i$ bids m_{min}.
- let m_1 be the most recent gas bid of $\mathcal{P}_i$. The searcher $\mathcal{P}_i$ waits until it sees a bid $m_0 > m_1$ cast by another player, where m_0 is the largest of all such bids, and immediately counterbids $\min\{\max\{m_1 \times (1 + \tau), m_0 + \eta\}, (\texttt{ev} + \gamma_i m_1) \times g_i^{-1}\}$, where η is the minimum gas price tick allowed by the protocol.

Blind raising can achieve a competitive advantage over reactive pure counterbidding, however, both strategies are strictly dominated. Moreover, we will see in the following propositions that, if searchers are rational, a searcher playing the blind raising or reactive pure counterbidding can at most obtain null revenue.

The following proposition shows that a searcher $\mathcal{P}_i$ using S_{BR} or S_{RC} against other searchers, can at most obtain zero profit.

Proposition 4.2. Let $\mathcal{P}_1, \mathcal{P}_2$ be the set of searchers playing the FPGA game with $g_2 \leq g_1$. Then, if $S_1 = S_{BR}$ or $S_1 = S_{RC}$ and η is a sufficiently small minimum gas price tick, there exists a blind raising strategy S_2 such that $u_2(S_2, S_1) \geq 0 \geq u_1(S_1, S_2)$.

Definition 4.3. We define the *perfect* grim-trigger gas bid for a FPGA as follows: at time t_{gm} it removes the incentive for all searchers $\mathcal{P}_i$ to increase their last gas bid. That is, the smallest possible gas bid m_{gm} such that,

$$\texttt{ev} - g_i m_{gm} < -\gamma_i m_i, \tag{4}$$

for all searchers $\mathcal{P}_i \in P$, where m_i is $\mathcal{P}_i$'s highest gas bid at $t = t_{gm}$.

Definition 4.4. We define the *imperfect* grim-trigger gas bid for a searcher $\mathcal{P}_i$ at time t as the smallest possible gas bid m_{gmi} such that $\texttt{ev} - g_j m_{\text{gmi}} < -\gamma_j m_j$ for all searchers $\mathcal{P}_j \in P$, where m_j is $\mathcal{P}_j$'s highest gas bid in $\texttt{view}_i(t)$.

Observation 4.5. If $\mathcal{P}_i$ has $\Delta_i = 0$, then their *imperfect* grim-trigger is equivalent to the *perfect* grim-trigger.

Now, we will construct simple strategies that induce a NE. We do not claim that the following cooperative strategies happen in reality, but rather to establish a basic understanding of the FPGA NE.

Cooperative exponential raising $S_E(m, \varepsilon)$: Is a strategy where searchers coordinate in time their bid increments introduced in [6]. The strategy works as follows. For each time $t = k\Delta$:

- the player i bids $W[k] = (m_0 + i\varepsilon) \times (1 + \tau)^{\lfloor k/n \rfloor}$ if $k \equiv i \pmod{n}$.

- If $\mathcal{P}_i$ observes a bid b at time t such that $t < k\Delta$ and $m > W[k]$, then $\mathcal{P}_i$ releases their *imperfect* grim-trigger immediately.

In this strategy, searchers coordinate in time to outbid each other. In order to do so, searchers should coordinate on chain to determine the order (potentially using the public correlating device). If a searcher deviates from the protocol, then one or more players release an *imperfect* grim-trigger leading to losses for all searchers. In [6] the authors prove that this strategy is a Nash equilibrium for a wide range of parameters. Moreover, the authors claim that the searchers do not necessarily need to coordinate out-of-band, but the strategy can develop organically on-chain. We claim that this is not necessarily true. First, the cooperative strategies explained are weak against Sybil attacks. Players could generate an arbitrary amount of addresses and push the equilibrium to its own benefit, leading to non-positive payoffs to other searchers. Also, this cooperation is suboptimal, i.e. players can take other strategies that lead to NE and have greater revenue.

Optimal cooperation $S_C(m_0)$: is a strategy where a searcher tries to coordinate the minimum bid with other searchers. The strategy tries to split the `ev` opportunity randomly. The player bids m_0, if he sees other bid $m_1 > m_0$, then all searchers release their *imperfect* grim-trigger. More formally:

- at $t = 0$, the players bid m_0.
- If at time t a player sees a bid $m_1 > m_0$, then $\mathcal{P}_i$ releases their *imperfect* grim-trigger immediately.

Observation 4.6. If players play the optimal cooperation, they are incentivized to generate more addresses and bid m_{min} with all of them. In the following proposition, we will suppose that players just control one address.

Proposition 4.7. Let $\mathcal{D}$ be a domain based on a proof of work consensus protocol with $\mathcal{B} \sim \texttt{Exp}(\lambda)$. Then, the $S_C(m)$ strategy induces a symmetric Nash equilibrium in the FPGA model, if and only if,

$$\frac{1}{n} \geq 1 - e^{-\lambda \Delta_{\min}} - O(\eta) \quad \text{and} \quad \frac{\texttt{ev}}{g_i + (n-1)\gamma_i} \geq m, \quad \text{for all } i,$$

where $\Delta_{\min} = \min\{\Delta_i\}$, and $\eta \geq 0$ is the minimum gas bid tick of the protocol.

If searchers can generate an arbitrary number of addresses, then, they are incentivized to use more than one address to maximize its payoff. Moreover, this equilibrium is highly unstable and needs off-chain communication to agree on the bid m. A small increment $m + \varepsilon$ lead to loses to all searchers.

The possibility of having a probabilistic ending time and seeing the bids of the competitors after some time Δ, allow searchers to "rationally censor" other bidders, not allowing them to extract the `ev` opportunity. We will present a strategy that allow a player to extract all the `ev` opportunity without sharing it with other players and not leading to any rational counter strategy.

Censor strategy $S_{CE}(m, \mathcal{A})$: is similar to optimal cooperation strategy. At the beginning, $t = 0$ the player bids m per gas unit. If he sees another player $P \notin \mathcal{A}$ bids $m_1 \geq m$ or a player $P \in \mathcal{A}$ such that $m_1 > m$, then the searcher executes the grim-trigger.

Null strategy $\varnothing^\star(m, s; \Delta)$: We define the strategy $\varnothing^\star(m)$ as, the searcher waits Δ to take an action. If the first gas bid observed is greater or equal to m, then the player does not bid or participate in the FPGA, otherwise follows the strategy s.

In the following proposition, we will show that the censor strategy induces a NE.

Proposition 4.8. Let $\mathcal{D}$ be a PoW domain with $\mathcal{B} \sim \texttt{Exp}(\lambda)$. The tuple of strategies given by the censor strategy and the null strategy $S = (S_{CE}(m), \varnothing^\star(m, S_C, \Delta), ..., \varnothing^\star(m, S_C, \Delta))$ is a Nash equilibrium for all $m \geq m_{\min}$ and $\texttt{ev}/g_1 \geq m$ if:

$$m_{\min} = \frac{\texttt{ev}(1 - e^{-\lambda \Delta_1})}{g_{1,\min}(1 - e^{-\lambda \Delta_1}) + \gamma_{1,\min} e^{-\lambda \Delta_1}}.$$

Moreover, the expected revenue `rv`, for the miner/validator, is $g_1 m$. Therefore, if $g_1 \leq g_{1,min}$,

$$\lim_{(\lambda \Delta_1, \gamma_{\min}) \to (+\infty, 0)} \texttt{rv}(S; \lambda, \Delta_1, \gamma_{\min}) = \texttt{ev} \frac{g_1}{g_{1,\min}}.$$

This implies that a player that has more than 50% of power, can enforce the total extraction of the MEV. More specifically, in FPGA, if the game is played infinite times (is a repeated game) among players with equity $\pi_1, ..., \pi_n$, with $\pi_1 > \frac{\sum_{i=2}^{n} \pi_i}{2}$, then the censorship strategies are Nash equilibrium and is an optimal Nash equilibrium for player 1.

In the model explained in the section 4.1, the following theorem states that, a wide ranges of outcomes are feasible in the FPGA auction. In other words, the Nash equilibrium condition is not very restrictive.

Theorem 4.9 (FPGA Folk)**.** Let $\mathcal{D}$ be a PoW domain with $\mathcal{B} \sim \texttt{Exp}(\lambda)$. Let,

$$F = \text{co}(\{(x_1, ..., x_n) : x_i \in [0, \texttt{ev} - g_i m], \exists! x_i \neq 0\}) = \{(x_1, ..., x_n) : x_i \in [0, \texttt{ev} - g_i m], \sum_{i=1}^{n} \frac{x_i}{\texttt{ev} - g_i m} \leq 1\},$$

where $m = \texttt{ev}(1 - e^{-\lambda \Delta_{\min}})/(g_{\min}(1 - e^{-\lambda \Delta_{\min}}) + \gamma_{\min} e^{-\lambda \Delta_{\min}})$ and $\gamma_{\min} = \min\{\gamma_i\}$. Then, given a suitable correlating device D, for any payoff vector $\boldsymbol{v} \in F$ there exists a correlated Nash equilibrium $(S_1, ..., S_n)$ such that $\boldsymbol{u} = (u_1(S_1, S_{-1}), ..., u_n(S_n, S_{-n})) = \boldsymbol{v}$.

With these results, we deduce that, in PoW domains, there are Nash equilibrium where the sequencer does not maximize their surplus and players can increase their revenue by optimizing their biding strategies

Observation 4.10. The cooperative, the exponential raise and the censor strategy induce, for a wide range of parameters, a Nash equilibrium. However, once a player deviates, all the other players do not have incentive to follow the strategy, since it leads to major costs. This motivates the following definition.

Subgame perfect Nash equilibrium (SPNE): We say that a tuple $(S_1, ..., S_n)$ of strategies is a subgame perfect Nash equilibrium if for time t and state $\texttt{st} \in \texttt{View}_i(t)$ of the game have that for every player i,

$$u_i(S_i, S_{-i} \mid t, \texttt{st}) \geq u_i(\tilde{S}_i, S_{-i} \mid t, \texttt{st}), \text{ for every strategy } \tilde{S}_i. \tag{5}$$

In other words, a subgame perfect Nash equilibrium lead players to be individually rational at any moment of the game. Observe that a SPNE is a NE and if $\texttt{view}(t) = \varnothing$ for all t, we have that a NE is a SPNE.

Remark: The NE given in [6] and all previous NE are not SPNE. This follows from the fact that executing the perfect grim-trigger have more costs than leaving the auction. However, we can map the previous NE to SPNE in the following way. Let $S_1, ..., S_n$ be the correlated NE constructed in Theorem 3.10 and $\overline{S_1}, ..., \overline{S_n}$ be a SPNE, then we define the SPNE-transform as:

$$S_i^t = \begin{cases} S_i, \texttt{view}_i(t) \cap \texttt{Deviation}(S_i) = \varnothing, \\ \overline{S_i}, \text{ otherwise} \end{cases}$$

Proposition 4.11. Let F be the set defined in theorem 4.17 and $\boldsymbol{v} \in F$. Let $S_1, ..., S_n$ be the NE constructed in 4.17 and $S_1, ..., S_n$ be a SPNE s.t. $u_i(S_1, ..., S_n) < v_i$, then $S_1^t, ..., S_n^t$ is a SPNE with $u_i(S_1^t, ..., S_n^t) = \boldsymbol{v_i}$.

In other words, 4.11 establish a correspondence between NE and SPNE.

In the last results, we have assumed $\mathcal{D}$ is a PoW-based with $\mathcal{B} \sim \texttt{Exp}(\lambda)$ and the NE hold if $\Delta \in \mathbb{R}_{\geq 0}$. In the following, we will study the Nash equilibrium when $\Delta = \mathcal{B} + \varepsilon$. Moreover, we will deduce the NE in PoS/PoA based blockchains.

Theorem 4.12 (Private Mempool)**.** Assume that all players share the same gas cost and gas efficiency parameters. The PGA model with a private mempool (independent of PoW or PoS/PoA) ($\Delta = +\infty, \gamma > 0$) with incomplete information (players know $\texttt{ev}$ and n) possess two type of equilibrium. Either all players use the same continuous mixed strategy with support $[b_{min}, \texttt{ev}]$, or at least two players randomize over $[b_{min}, \texttt{ev}]$ with each other player i randomizing over $\{0\} \cup (b_i, \texttt{ev}]$ with $b_i > b_{min}$, and having a mass point at 0 equal to $F_i(b_i)$. When two or more players have a positive density over a common interval, they play the same continuous mixed strategy over that interval. Moreover, the expected revenue for the sequencer is $\texttt{ev}$ and the price of anarchy is $\Omega(n)$.

Corollary 4.13. Let $\mathcal{D}$ be a PoA/PoS domain such that supp $\mathcal{B} \subseteq (\mu - \varepsilon, \mu + \varepsilon)$ with $0 \leq \varepsilon < \Delta$ and $\mu \in \mathbb{R}_{\geq 0}$. Then the FPGA has the same NE stated in 4.12 and the expected revenue for the sequencer is $\texttt{ev}$.

4.1.1 The Flashboys 2.0 FGPA model post-merge

The FPGA model formulated in [6] has not been analysed for parameters consistent with Ethereum post-merge; so we include this here as it will be of interest to many readers. Their model for PoW Ethereum is actually reasonable for Ethereum post-merge except for the choice of an exponential distribution for the block time. This is primarily due to the simplicity of the model, assumptions of naive block producers, that we are only considering the one-block game (avoiding the issue of fork choice and attestations), and there being little change to the execution layer post-merge.

Modelling the Ethereum block time post-merge

Using the merge honest validator specification [18], we can establish clear rules that govern the block time.

1. Validators are expected to produce a block at the beginning of their assigned slot, time t_s say.
2. The slot committee will only broadcast their attestations for the block if it is valid **and** received a maximum of 4 seconds after the beginning of the slot. That is, a validator must produce and broadcast a valid block by $t_s + 4$ at the latest to ensure sufficient attestation for inclusion.
3. Due to the implementation of attestation inclusion delay [4], a validator will have access to all the attestations it needs, along with a valid block from the previous slot, by time $t_s - 8$.

So, we can expect the block time to occur on the interval $[t_s - 8, t_s + 4]$, as block times outside this interval amount either to dishonest validator behaviour or a block that won't be included. Validators will be aiming for t_s, so we can anticipate that the most probable block time is t_s. Given the block time is a result of many unknown variables it should conform to a natural distribution. Therefore, a reasonable model of the Ethereum block time post-merge is a PERT beta distribution on $[t_s - 8, t_s + 4]$, with most likely value $b = t_s$, minima $a = t_s - 8$, and maxima $c = t_s + 4$; this could be simplified to a triangular distribution if needed.

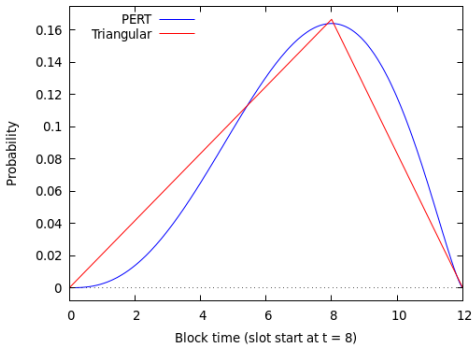

Figure 1: Block time PDFs for Ethereum post-merge

The main result from [6] was,

> ***Theorem 1*** *For parameters consistent with Ethereum PGAs, there exists a grim-trigger Nash equilibria for a 2 player PGA where both players follow a D,W cooperative strategy.*

Due to the change in probability distribution of the block time, this theorem will not hold post-merge. For example, if two searchers use the $S_c(t_s - 8, b_0)$ cooperative strategy, we have the following proposition.

Proposition 4.14. For parameters consistent with Ethereum FPGAs post-merge, a grim-trigger Nash equilibria does not exist for a 2 player FPGA where both searchers follow the $S_c(t_s - 8, b_0)$ strategy.

Furthermore, if there exists a non-trivial profitable deviation time when both players use the $S_c(t_s - 8, b_0)$ strategy, it is given by the following proposition.

A graph with parameters consistent with an Ethereum FPGA post-merge is given along with the proofs in the appendix.

Proposition 4.15. For a 2 player FPGA on Ethereum post-merge, where both searchers use the $S_c(t_s-8, b_0)$ strategy, it is profitable for either searcher to deviate with bid $b_1 > b_0$ at time t_{deviate} such that,

$$f(t_{\text{deviate}}, \Delta) - l(b_1)(1 - f(t_{\text{deviate}}, \Delta)) > \frac{1}{2}(1 - l(b_0)) \quad .$$

4.1.2 Sybil resistance Nash equilibrium

As we have mentioned previously, nearly all Nash equilibrium are constructed assuming that players know the number of players extracting the MEV opportunity. The following theorem states that there exists Sybil-resistent Nash equilibrium with positive players surplus.

Theorem 4.16. Let $\mathcal{D}$ be a domain. Assume that there are n symmetric players extracting an MEV opportunity with gross profit $\mathtt{ev}$, gas cost g and efficiency γ. So, for all m such that

$$\frac{\mathtt{ev} + (2^n\gamma - g)m}{2^n(\mathtt{ev} + (\gamma - g)m)} \geq \mathbb{P}\left[\mathcal{B} \in [t, t+\Delta]\right] \text{ for all } t \geq 0, \tag{6}$$

then there is a sybil-resistent Nash equilibrium with expected profit $\frac{\mathtt{ev}-gm}{2^n}$.

FPGA conclusions: In this section, we proved that PoW based chains with low latency and block duration ratio lead to a wild range of Nash equilibrium. This equilibrium lead to unpredictable sequencers' revenue and number of on chain transactions. We observed that optimal cooperative strategies, in general, lead to high block size use and that censoring strategies lead to MEV centralization. Moreover, we obtain that in blockchains with private mempools, the number of transactions that land on-chain vary from 2 to n and the price of anarchy is $\Omega(n)$. Also, the expected sequencers' revenue is $\mathtt{ev}$. A straightforward conclusion is that PoS/PoA chains with high latency and block duration ratio lead to Nash equilibrium with same on chain transactions and sequencers' revenue.

4.2 Random ordering

The back-running price gas auction game or random ordering game is modeled by the abstract MEV game with a random allocation rule and fixed gas price.

In this game, players observe an MEV opportunity that is produced by a change of state s produced by a transaction or set of transactions $\mathcal{TX}$ that will be executed in the next block (with probability p_v, we will assume $p_v = 1$). These transactions induce an MEV opportunity for each player of gross value $\mathtt{ev}$ if their bundle is executed after the event execution.

Since all transactions sent by searchers will have the same bid, the order of the transactions is not defined. Before the geth-client change [7] the Ethereum blockchain ordered the transactions randomly, however after the change the transactions with the same gas price were ordered by received time. Different domains have their own protocol to order transactions with the same bid.

Uniform Random ordering: Sequencers order transactions randomly, and all orders are equally probable. If there is a total of m transactions competing for the $\mathtt{ev}$ opportunity and $k \leq m$ are sent by the player P_i, then the expected payoff is:

$$\mathbb{E}[\Delta b_i] = \frac{k}{m+1}(\mathtt{ev} - g_i m_{\min} - (k-1)\gamma_i m_{\min}) - (1 - \frac{k}{m+1})k\gamma_i m_{\min} \tag{7}$$

In this setting, a strategy is defined similarly to FPGA. A strategy is defined analogously, but the actions a are of the form $(t, \mathtt{k}; i)$ where t is the time $\mathtt{k}$ is the number of transactions sent and i is the player.

Theorem 4.17. [R.O. Folk] Let $\mathcal{D}$ be a PoW domain with $\mathcal{B} \sim \mathtt{Exp}(\lambda)$. Let,

$$F = \{(x_1, ..., x_n) : x_i \in [0, \tilde{\mathtt{ev}}_i], \sum_{i=1}^{n} \frac{x_i}{\tilde{\mathtt{ev}}_i} \leq 1\},$$

where $m = m(\gamma_{min}, g_{min}, \mathtt{ev}, \Delta_{\min})$ and $\gamma_{\min} = \min\{\gamma_i\}$. Then, given a suitable correlating device D, for any payoff vector $\boldsymbol{v} \in F$ there exists a correlated NE $(S_1, ..., S_n)$ such that $\boldsymbol{u} = (u_1(S_1, S_{-1}), ..., u_n(S_n, S_{-n})) = \boldsymbol{v}$.

Theorem 4.18. Let $\mathcal{D}$ be a domain with private mempool with random ordering allocation rule. Assume that all players share the same gas cost and gas efficiency parameters. Also, assume that the gas price is fixed for all players or the gas price of the transaction `tx` that induces an MEV opportunity is leaked. If $n >> 1$, then $NE \subseteq V_z := \{(s_1, ..., s_n) : s_i \in \{z, z+1\}\}$ for some $z \in \mathbb{R}_{\geq 0}$. Moreover, the price of mev is $O(\frac{\texttt{ev}-gm}{\gamma m})$.

4.3 Fair Sequencing service

FSS ordering (Post pull request [7]): Sequencers order transactions with same gas price by received time.

$$\Delta b_i = \begin{cases} \texttt{ev} - \beta_i b_{\min}, \text{ if its the first transaction received by the sequencer,} \\ -\gamma_i b_{\min}, \text{ otherwise} \end{cases} \tag{8}$$

Therefore, the searcher with the smaller latency will win each game, obtaining $\texttt{ev} - \beta_i b_{\min}$. Being more realistic, we can include the costs of having less latency (bribing searchers, becoming searchers, improve geographic localization). These costs are not necessarily paid to sequencers. Now, let $l = (l_1, ..., l_n)$ denote all the searchers' latency costs. If we assume that $\Pr[i \text{ wins} \mid l] = \frac{l_i}{\sum_{j=1}^n l_j}$, then the expected utility is

$$\mathbb{E}[\Delta b_i] = \begin{cases} \frac{l_i}{\sum_{j=1}^n l_j}(\texttt{ev} - \beta_i b_{\min}) - l_i - \frac{l_{-i}}{\sum_{j=1}^n l_j}\gamma_i b_{\min}, \text{ if } i \text{ send tx}(\texttt{ev}), \\ 0, \text{ otherwise.} \end{cases} \tag{9}$$

The costs l are not necessarily economic costs but rather can be economic risks. For example, in PoS in order to minimize their latency, searchers can become validators. While this does not necessarily include any economic cost, it definitely includes some risks or costs of opportunity. Therefore, in the PR-BPGA, a searcher can choose between sending or not a transaction, and investing $l \in \mathbb{R}_{\geq 0}$ in improving the latency. More formally, the set of actions is $A = \{\texttt{send tx}, \perp\} \times \mathbb{R}_{\geq 0}$. Then, the set of strategies is the set of all probability distributions of A.

Theorem 4.19. In FSS BPGA (post PR [7]), if $\texttt{ev} \geq b_{min} + (n-1)\gamma b_{min}$, then there is a unique Nash equilibrium. Moreover, this equilibrium is pure, symmetric and have the form

$$l = \frac{n-1}{n^2}(\texttt{ev} + (\gamma - 1)m_{min})\mathbf{1}^T.$$

The price of anarchy is $\Omega(n)$. Moreover, this equilibrium is stable in the replicator strategy for $n \leq 5$. This equilibrium is independent of Δ and $\mathcal{B}$.

4.4 Meta-data ordering

4.5 Flashbots Auction

Flashbots Auction (the v0.6 before the Ethereum transition to proof of stake)[6] is a combinatorial auction that all allows players to bid for an ordered set of transactions, known as bundles. The allocation and payment rule used by the Flashbots auction has a similar mechanism to a first-price sealed auction. That is, the bundles are ordered by average gas price 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. That is, if a set of bundles $B_1, ..., B_n$ compete and have the same gas costs, then the bundle appended in the chain will be the one with higher effective gas bid (see below).

In other words, Flashbots relayer try to build the block with the highest profits among all the block 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 which the Flashbots relayer must be accounted for [20]. In order to build the block with the highest profit, Flashbots (to our knowledge[7]) uses a greedy

[6] After the merge, the Flashbots team will make a change to the Flashbots auction, not necessarily preserving the current structure of the auction. In the future, we will investigate this changes.

[7] https://docs.flashbots.net/

approximation algorithm, ordering the bundles by average gas price or as the Flashbots team described in their docs, the score of the bundle. The score of a bundle B is defined as

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

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. By definition, the score of the bundle depends on the current state $\mathtt{st}$. For an state $\mathtt{st}$ and a set of bundles $\mathcal{T}$, we denote by $\mathcal{T}^+(\mathtt{st})$ as the set of bundles that do not revert. For a bundle B, we denote by $\mathcal{R}(B; \mathtt{st})$ the bundles that conflict with B. Currently, MEV auctions are not open sourced or public available. However, we will make the following abstraction. Let $B_1, ..., B_n$ be all the bundles received by the Flashbots relayer, then the Flashbots combinatorial, can be simplified in the following way:

Algorithm 1 Abstraction of Flashbots auction algorithm v0.6

```
st ← Current state
block = []
while gasCost(st ∘ block) ≤ gasLimit
and T ≠ ∅ do
    B ← argmax_{B∈T⁺(st^(i))} {s(st ∘ B)}
    T ← T ∖ R(B, st)
    st ← st ∘ B
    block.append(B)
end while
```

Assuming that the number of operations per bundle is bounded by some constant M, we have that the computational complexity of this algorithm is $O(n^2)$. Since in general bundles do not change their scoring, another potential abstraction could be the following:

Algorithm 2 More efficient abstraction of Flashbots auction algorithm v0.6

```
T⁺(st) ← SortByScore (T⁺(st))
while gasCost(st ∘ B) ≤ gasLimit and T^(i) ≠ ∅ do
    B ← T⁺(st)[0]
    st ← st ∘ B
end while
```

Moreover, in some cases, ordering the bundles by effective gas price does not give an optimal revenue. 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.

Proposition 4.20. The Flashbots combinatorial auction is not optimal. More specifically,

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

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 $i \neq 1$ 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}.

Searchers reputation: In order to maintain reliable Flashbots relay performance, Flashbots introduced searcher reputation to provide consistent access to the relay for searchers with a good performance track

record during periods of heavy load. Reputation mechanism is the current solution used by the Flashbots builder to make the relay robust against sophisticated Layer 7 attacks. The current reputation system is designed to classify searchers into a high reputation and low reputation queue. The high reputation queue is designed to filter out searchers who use an excessive amount of computation resources on the relay. Otherwise, both queues are identical. According to Flashbots, the reputation of a searcher S is defined as

$$r(S) = \frac{\sum_{\texttt{tx}\in\mathcal{T}^e(S)} \Delta_{coinbase}(\texttt{tx}) + g(\texttt{tx})m(\texttt{tx})}{\sum_{\texttt{tx}\in\mathcal{T}(S)} g(\texttt{tx})} \tag{10}$$

where $\mathcal{T}(S)$ is the set of bundles send by S and $\mathcal{T}^e(S)$ is the subset of bundles sends by S and executed on chain. Nevertheless, probably Flashbots team has added a decay factor to the reputation to add more fairness to new users and current inefficient users in the high reputation queue. That is, for some time windows $[t_i, t_{i+1}]$, the reputation $r(S,i)$ is computed. Let t_k be the current time. Then, the total reputation is computed by

$$R(S) := \sum_{i=0}^{k-1} f(t_k - t_{i+1}) r(S,i) \tag{11}$$

where $f : \mathbb{R}^+ \to [0,1]$ is a monotone decreasing function.

4.5.1 Common knowledge MEV opportunities

In this section, we introduce the parameter p, that models the probability of finding the transaction that allows the searcher to extract the MEV opportunity. For each auction, the probability of a searcher to find a bundle B that extracts MEV is modeled by a Bernoulli(p). The intuition behind this model goes as follows. In some cases, MEV extraction can be thought as a NP-problem, so the algorithm of a searcher that finds it before a time t can be modeled as a bounded random variable X. Assuming that the auctions terminates in a given time t_{end} known by the players, the probability of bidding for MEV for the current auction is $p = \mathbb{P}(X \leq t_{end} - \Delta)$, where Δ is the latency between searchers and sequencers. Similarly, if there are no bidders for the first auction, the probability of finding the MEV in the second auction is $p = \mathbb{P}(X \leq 2t_{end} - \Delta \mid X \geq t_{end})$, and so on. In other cases, players have different software programs that find competing bundles with different MEV opportunities, for examples in finding Arbitrages and Liquidation. However, in this section we will assume that all searchers have the same resources, that is, we will assume that all players can extract all `ev` tokens from the MEV opportunity or can extract no value. Also, searchers can not see each other's bids before the blocks are appended to the chain. Moreover, the players know the number of players n and the probability p of finding the MEV. In this section, we find all the equilibrium in the non-repeated one shoot Flashbots auction with no partial extraction.

Proposition 4.21. Define the strategy of a searcher bidding m be $s(m)$. Assume wlog that the n have $g_1 = ... = g_k < g_{k+1} ... \leq g_n$ gas efficiencies for some $n \geq k \geq 0$. Let g' be the second smallest gas efficiency. and $\mathcal{P}'$ be the set of players that gas bidding $\texttt{ev}/g'$ is not a dominated strategy. Then one-shoot Flashbots auction with $p = 1$ and MEV opportunity with value `ev` being public information have the following set of mixed Nash equilibrium:

$$\{(s_1, s_2, s_3, ..., s_n) : s_1, ..., s_n \in S_1 \times ... \times S_n \text{ and } \exists i, j \text{ s.t. } s_i = s_j = s(m/g')\}$$

where S_i is the set of all non strictly dominated strategies of player i. If players follow the best response strategy, they will converge to the equilibrium

$$(s_1(m/g'), ..., s_k(m/g'), s_{k+1}(\min\{m/g', m/g_{k+1}\}), ..., s_n(\min\{m/g', m/g_n\})).$$

The set of NE is invariant under risk-lover, risk-neutral and risk-averse agents.

Intuitively, this states that any Nash equilibrium in this setting must have at least two searchers bidding `ev`. Otherwise, if only one searcher is bidding `ev` they can decrease their bid by ϵ for a profit. If no searchers are bidding `ev` then any searcher that is currently losing can increase their bid to ϵ more than the highest bid for an increase in their value.

Theorem 4.22 (Flashbots Auction)**.** The Flashbots auction with incomplete information (players know n and p, but not the content of other players transactions) possess a one type of equilibrium. All players use the same continuous mixed strategy with support $[0, \overline{s}]$ with $\overline{s} = \texttt{ev}\left(1-(1-p)^{n-1}\right)$. For $p = 1$, the Nash equilibrium are the ones given in proposition 6.8. For $p < 1$, there is a unique Nash equilibrium and is symmetric. The equilibrium is given by the cumulative distribution function $F : [0, \texttt{ev}] \to [0, 1]$ defined by

$$F(x) = \begin{cases} \sqrt[n-1]{\frac{\texttt{ev}-\overline{s}}{p^{n-1}(\texttt{ev}-x)}} + 1 - \frac{1}{p}, & \text{if } 0 \leq x \leq \overline{s}, \\ 1, & \text{if } x > \overline{s} \end{cases}$$

with $\overline{s} = \texttt{ev}\left(1-(1-p)^{n-1}\right)$.

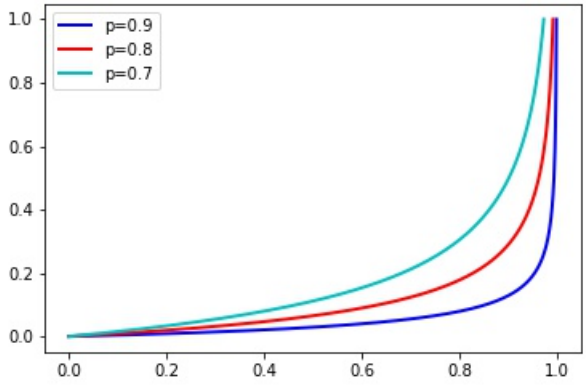

Figure 2: Distribution function of symmetric Nash equilibrium with $p = 0.9, s = 0.95$ and $\overline{s} = 0.8, 0.9$.

Using all lemmas above, we deduce the following theorem we deduce the theorem 4.22.

Corollary 4.23 (Revenue convergence)**.** The expected revenue for the sequencer converges to $\texttt{ev}$, if $p \to 1$ or $n \to +\infty$.

4.5.2 Flashbots asymmetric softwares and gas efficiency

In the above section, we assume that all players find or do not find the bundle B that extracts the $\texttt{ev}$ opportunity during the auction. In this section, we briefly overview the potential consequences if the bidders have different. We assume that each player $\texttt{ev}$ opportunity (observe competing $\texttt{ev}$ opportunities and have different gas efficiencies) is modeled by random variables $X_1, ..., X_n$ that are independently distributed according to the functions $F_1, ..., F_n$ on $[\texttt{min}_{\texttt{fee}}, \texttt{mev}]$. Moreover, we assume that these functions are strictly increasing and differentiable.

Notation 4.24. A bidding function β is a function of a player p such that if the observation of F is x, then bids $\beta(x)$. That is, is a pure strategy defined by its private value.

Theorem 4.25. In the Flashbots auction with $F_1, ..., F_n$ $\texttt{ev}$ software and gas efficiency distributions, there is a unique Nash equilibrium with $\beta_1, ..., \beta_n$ differential increasing bidding functions. If $\mathbb{E}(\|X_i - \texttt{mev}\|) \to 0$ for at least two players, then $\texttt{rv} \to \texttt{mev}$.

Proof. Auction theory, [10] Appendix G. □

In conclusion, if players are myopic and the $\texttt{mev}$ extraction is sufficiently competitive, then miners will tend to bid all the extractable value opportunity, leading all the benefits to the sequencers.

4.6 Comparative of theoretical results

The following table will give a brief comparative of FPGA and Flashbots auction [8]:

	FPGA PU	FPGA PR	R.O. PU	R.O. PR	L.G.	F.A.
|NE|	$\mathfrak{N}_1$	1	$\mathfrak{N}_1$	1	1	1
Expected searcher revenue $\leq$	mev	0	?	?		0
Expected miner revenue	0	mev	?	?	?	mev
Censorship Strategy resilient	✗	✓	✗	✓	✓	✓
Price of MEV (PoA)	$\Omega(n)$	$\Omega(n)$	$\Omega(ish)$	$\Omega\left(\frac{\mathtt{ev}-gm}{\gamma m}\right)$	$\Omega(n)$	$\Omega(1)$

Table 1

5 MEV-Boost

MEV auctions (Flashbots auction) introduce a first price sealed auctions among searchers to bid for MEV opportunities. MEV-boost, similiar to MEV auctions, can be defined as a protocol that introduce first price sealed auctions among builders (however current design of MEV-boost does not make the bidding process private[9]). Builders are middle man players between searchers/users and validators, responsible for building the most efficient, social welfare or profitable blocks and submit them to the validators. For each round, builders send the header of a block and a bid. Afterwords, the validator of that slot, chooses the header block with more payoff and commits to execute it. Afterwards, builder sends the block and the validator executes and broadcast it to their peers[10]. The only constraints that builders have is to build valid blocks with the consensus protocol semantics. Since the builders are trusted identities, to build trust among users and searchers, builders commit to ordering mechanisms. In this section, we will assume that builders do not deflect from the committed protocol and the ordering mechanism.

5.1 Formalization of MEV game with MEV-Boost

The MEV game where searchers can choose the builders that they trust (or increase their expected payoff) to send their bundles or transactions. The MEV-Boost game has MEV game structure $\mathbb{MB}$, with some minor changes. Builders have a fixed set of peers $\mathcal{O}_1, ..., \mathcal{O}_k$ in the p2p network $G = (E, V)$. Each builder $\mathcal{O}_i$ commits to its own auction mechanism $(\mathbf{x}_i, \mathbf{pr}_i)$. Each round, a builder constructs a block (with the bundles and transactions received) with value v_i. Therefore, the game between builders can be thought as an asymmetric MEV game, where the random variable $\mathcal{B}$ is determinate by the PoS consensus (and the proposal[11]). The gossip network G (in this context, is the weighted graph that determines the latency to see another builder bid) is induced by the location and the relayers algorithm.

Observation 5.1. In general, builders that execute reverted transactions will be more inefficient than builders that do not accept reverted transactions. The main reason for it is that searchers will have to price in the risk of being reverted. By the equivalence revenue theorem this will lead to the same expected revenue. However, since the block-space is limited this would lead to other transaction not being included, leading to less revenue overall. Moreover, the users will have preferred builders that execute their transactions, leading in general to blocks with more value.

[8] A standard auction is a type of auction where the player with the highest bid wins the auction

[9] https://github.com/flashbots/mev-boost/issues/112

[10] More architecture details can be seen in https://github.com/flashbots/mev-boost

[11] In general, a rational proposal will try to delay as much as possible its construction, since in general, MEV increases over time.

In the following, we will state the importance of having private bids in order to have a proper decentralized builder ecosystem.

Theorem 5.2. Let P_1 and P_2 be two builders. The value of the block is drawn by a distribution V. Assume that the auction terminates in a fixed time t_f and that P_i sees the bid of P_j in $\Delta_i > 0$ time for $i = 1, j = 2$ and $j = 1, i = 2$. Assume that $\Delta_1 < \Delta_2$. Then, in equilibrium, the expected profits of P_1 exceed the profits of the P_2.

6 Zero sum games

6.1 Hamelin strategy

More examples can arise in different MEV games. In the following, we will explain a clear example of how to execute a Hamelin strategy in the Polygon network that could potentially break liveness in the Polygon network.

The strategy creates visible arbitrage opportunities in the mempool to incentivise MEV bots to spam arbitrage transactions. Manipulation of account nonces and the properties of bor client transaction propagation is used to prevent MEV bots from actually extracting the arbitrage opportunity, rather the trade reverses at no cost to the strategy user except for gas fees. This allows an adversary to create a large number of on-chain spam transactions for minimal cost. In other words, **the strategy bribes MEV bots to fill Polygon PoS blocks with spam transactions, without actually paying the bribe**. The following is an example of such a strategy that can easily be deployed on Polygon PoS using a public RPC with minimal capital requirements:

1. Choose two popular tokens, for example Matic and DAI, with an Uniswap V2 pool with enough liquidity.
2. Deploy a token $\mathbb{T}$ and deploy two Uniswap V2 pools $\texttt{pool}_1$ and $\texttt{pool}_2$ respectively with Matic and Dai.
3. Fix set of addresses $\mathcal{A}$.
4. For each address, construct a buy transaction $\texttt{tx}_1$ in $\texttt{pool}_1$ that leaves a triangular arbitrage opportunity of value $\delta >> 0$.
5. Increment the address nonce and construct a sell transaction $\texttt{tx}_2$ with the quantity of tokens $\mathbb{T}$ that the transaction $\texttt{tx}_1$ will generate, and place a slightly higher gas bid.
6. For each address, broadcast $\texttt{tx}_2$. Wait a few seconds and broadcast $\texttt{tx}_1$.
7. Go back to step 4).

We ran short tests of the example strategy on Polygon PoS mainnet to obtain data to establish a proof of concept; it was impossible to use a private testnet as we were unable to replicate the obfuscated behaviour of MEV bots on mainnet. We generated transactions exclusively **off-chain** with code that is **private** to us; we did **not** attempt the example strategy with such volume and order size as to cause a DDoS event or to cause user transactions to be censored; and we did **not** test the example strategy with such volume, order size, and for a sustained period as to impact liveness of the chain.

Figure 1 displays the Polygon spam analysis tool results during testing of the example strategy using 5 addresses, an order size of 15 matic, for five iterations separated by 10 seconds; total estimated cost of execution was 0.5 Matic, where each 'burst' cost 0.1 Matic (gas price approximately 75 Gwei). This demonstrates that it was possible to increase the number of spam transactions in a block from 50 to around 150 (with spikes up to 400) for five blocks for a cost of 0.1 Matic with gas prices at 75 Gwei. We therefore anticipate from these results that by increasing the order size, number of addresses, and volume of trades, it would be possible for an attacker to fill every block with spam transactions initially at a price of 0.02 Matic per block given an initial base fee of 75 Gwei, with later block costs rising with the base fee according to the Polygon EIP-1559 implementation.

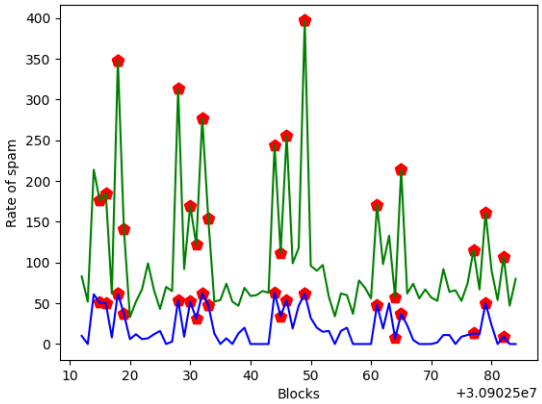

Figure 3: Green line = Number of transactions, Blue line = % Arbitrage transactions (lower bound), Red dots = Fake arbitrage opportunities

Put more simply, we anticipate that it is possible to fill an entire block with spam at 0.75% the cost of using the entire gas limit of a block.

Origin of the issue: It is well known that the transaction propagation mechanism of the bor client incentivizes spamming to extract MEV opportunities. A fair assumption is that the mechanism results in transactions landing in a random block according to a uniform distribution; with transactions ordered according to gas price within the block they end up in. Furthermore, it is reasonable to assume that transactions with the same gas price will be ordered according to arrival time at the elected validator provided they are valid. The above strategy works by ensuring the sell transaction $\texttt{tx}_2$ is propagated fully through the mempool, but is temporarily invalid since the nonce is too high. Then, after broadcasting the buy transaction $\texttt{tx}_1$, $\texttt{tx}_2$ only becomes valid once a validator receives $\texttt{tx}_1$, and since $\texttt{tx}_2$ has a higher gas price than $\texttt{tx}_2$, it is place directly after $\texttt{tx}_1$ in most cases. Now, MEV bots see $\texttt{tx}_1$ as a back-running MEV opportunity, and spam the network with transactions at the same gas price as $\texttt{tx}_1$ in an attempt to be positioned directly after it in the block to extract the MEV. $\texttt{tx}_2$ has a slightly higher gas price than $\texttt{tx}_1$, so $\texttt{tx}_2$ also appears as another back-running MEV opportunity, which causes MEV bots to spam the network to extract the MEV opportunity there as well. However, $\texttt{tx}_2$ doesn't represent a back-running MEV opportunity at all, since $\texttt{tx}_1$ will be placed directly before $\texttt{tx}_2$. On the other hand, bots do not have incentives to add more layers to their bots to prevent this kind of attack. Because, that would add less competitiveness against other bots, leaving extra profits if no one is executing this attack. For these reason, we do think that this is an error or bug of searchers' bots, but rather a fail on the ordering mechanism of Polygon chain.

Theoretical approximation: We will model the arbitrage game as a back running game that have structure of a random ordering MEV game where players do not see each other's bid before the next block. We will assume that the game is an MEV game $\mathbb{G}$, where all players observe the same value v. Players want to back run an MEV opportunity with gas price m. Also, players do not observe each other bids and the block production variable $\mathcal{B}$ is constant. Moreover, we will assume that all players same the same gas costs g for executing the MEV transaction and for the failed transactions γ. If there is a total of m transactions trying to extract the MEV opportunity and $k \leq m$ are sent by the player P_i, then:

$$\mathbb{E}[\Delta b_i] = \frac{k}{m+1}(\texttt{ev} - gm - (k-1)\gamma m_{\min}) - (1 - \frac{k}{m+1})k\gamma m$$

Theorem 6.1. Let $\mathcal{D}$ be a domain with private mempool with random ordering allocation rule. Assume that all players share the same gas cost g for extracting the opportunity and same gas costs for reverted transactions γ. Also, assume that the gas price is fixed for all players. Then,

the expected gas used to capture the fake MEV opportunities are given in the figure 4.

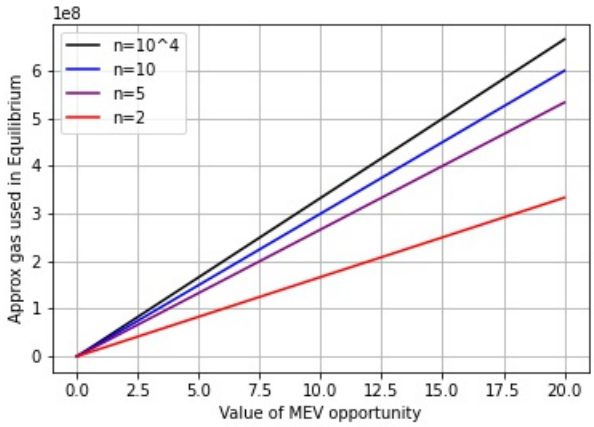

Figure 4: Gas misused in equilibrium in MEV back-running game in Polygon.

We recall that the block gas limit in polygon is $30,000,000$, therefore, with two and three competitive players would be enough to create a fake MEV opportunity of 20 and 10 respectively to fill the block with spam arbitrage transactions. Clearly, players that want to send a transaction to polygon will raise their bid enough. With a proper monitor of the mempool, the attacker could create fake MEV opportunities with a slighter gas bid. So, this transaction will be omitted in the following block. This method can be repeated for some time to stop break liveness for enough blocks. This strategy could be implemented to manipulate the price oracles or to artificially create cross chain MEV opportunities [13].

6.2 Hansel and Gretel strategy

As mentioned in the preliminary, Uniswap transactions with enough slippage can get sandwiched by searchers to extract net profits. The *Hansel and Gretel strategy*, similar to salmonella, tricks searchers to buy a token created by the executioner of the strategy, H for short. To this strategy to work, we have to assume that there are searchers in mainnet that try to capture the sandwich opportunities through PGA.

Let C be the total capital of H. The strategy works as follows:

- Deploy a token $\mathbf{T}$ and deploy an Uniswap V2 pools `pool` with $R_E \leq C$ and $R_{\mathbf{T}}$.
- Choose $\Delta + R_E \leq C$ and create a swap transaction `tx` with slippage η.
- Wait for a searcher S to send a buy transaction $\texttt{tx}_b$ with optimal sandwich Δ_{sand} in the mempool.
- Create a transaction that removes all the liquidity of the pool $\texttt{tx}_r$ and a bundle $B = \{\texttt{tx}_b, \texttt{tx}, \texttt{tx}_r\}$ and send it to Flashbots RPC with bid $\text{MEV}_S + \varepsilon$.

However, in general, searchers will not sandwich MEV opportunities with not enough net profits. Therefore, we must assume that searchers will try to sandwich the transaction trough PGA, if PNL $\geq$ L for some $L \in \mathbb{R}_{\geq 0}$. Therefore, MEV_H is the solution of the following optimization problem:

$$
\begin{aligned}
\max \quad & \Delta_{sand}(\Delta, R, \eta) \\
\text{s.t.} \quad & \Delta + R = C, \\
& \Delta - G^{-1}((1-\eta)G(\Delta)) \geq L, \\
& 0 \leq \Delta, 0 \leq R, 0 \leq \eta \leq \eta_{max},
\end{aligned}
$$

Lemma 6.2. Let P be a pool with reserves R and R' of token A and B respectively. Let tx be a trade with amount Δ and slippage η. Then, the solution to **??** is given by:

$$\Delta_{sand} = \frac{-(\gamma\Delta + 2R) + \sqrt{(\gamma\Delta + 2R)^2 + 4(R^2 + R\gamma\Delta)\frac{\eta}{1-\eta}}}{2} \tag{12}$$

Proof. Similar to [20]. □

Observe that the function is increasing on η, therefore, to solve MEV_H we can assume that $\eta = \eta_{max}$.

Lemma 6.3. The function $\Delta_{sand}(\Delta, C - \Delta, \eta_{max})$ is monotone decreasing in $\Delta \in [0, C]$.

Proof. It is enough to prove it for $C = 1$ doing a change of variables. Then it is easy to check that $\frac{\partial \Delta_{sand}(\Delta, 1-\Delta, \eta_{max})}{\partial \Delta} < 0$. □

Corollary 6.4. The solution of MEV_H is given by $(\Delta_{min}, C - \Delta_{min}, \eta_{max})$, where Δ_{min} is the unique solution of

$$\Delta - G^{-1}((1-\eta)G(\Delta)) = L.$$

Proof. Since $\Delta_{sand}(\Delta, C-\Delta, \eta_{max})$ is decreasing in Δ, $\Delta - G^{-1}((1-\eta)G(\Delta)) \geq L$ and $\Delta - G^{-1}((1-\eta)G(\Delta))$ is increasing in Δ, we have that the optimal solution of MEV_H is given by Δ_{min}. □

Proposition 6.5. Let S be searcher and H be the one executing the Hansel strategy. Assume that $\texttt{st}$ is a null MEV state, $\mathcal{T}_S^M = \{\texttt{tx}\}$ and $\mathcal{T}_H^M = \{\texttt{tx}, \texttt{tx}_b\}$. Then,

$$\text{PNL}(\Delta, \eta) = \text{MEV}_S(\texttt{st}) < \text{MEV}_H(\texttt{st} \circ \texttt{tx}_b) = \Delta_{sand} \tag{13}$$

where $\texttt{st}$ is the current state and $\texttt{tx}_b$ is the optimal buy transaction of S.

In general, Uniswap V2 sandwich attacks are executed by searchers using the Flashbots auction, paying more than 99%. To outbid conflicting bundles using the Flashbots auction, we have to bid $\text{MEV}_S(\texttt{st})$. By proposition 6.5, this leads to positive profits, making this strategy dominant under PGA bots.

Theorem 6.6. If exist a PGA bot, the Hansel and Gretel strategy is profitable bidding L. Moreover,

$$\text{MEV}_H = C\left(\sqrt{\frac{1}{1-\eta_{max}}} - 1\right) + O(\gamma) \tag{14}$$

6.3 Red strategy: A strictly dominant strategy in PoW Flashbots auction v0.6

Flashbots relayer can simulate bundles in order to construct the most profitable bundles. However, there is a potential (and exploitable) difference between the simulation and the execution payoff. Since Flashbots can not predict the *special variables* of the block beforehand, it can not ensure the exactly payoff of each bundle, and so, the payoff of the block. Using this, one can probabilistically boost its bundle score $sc(B)$. More concretely, the score function can change through the execution of the block. Theoretically, this strategy would work in the new Flashbots auction. However, the changes in the Flashbots auction are deeper (and more robust) than the ones specified in the documents (see more details in empirical analysis section). So, we emphasize that this strategy would not have an impact in the new Flashbots auction. However, after MEV-Boost, this strategy could be used in other suboptimal builders.

6.4 Motivational example

Let B_1 and B_2 be two conflicting bundles that extract an MEV opportunity of value 4 ETH. The $\Delta_{coinbase}$ of the first bundle is 2 ETH. However, the $\Delta_{coinbase}$ of the second bundle is probabilistic. If `block.timestamp` $\equiv 1 \pmod 2$ it pays 0.1 ETH. Otherwise, it pays $2 + \varepsilon$ ETH. If the Flashbots relayer makes a unique total block simulation, then he will choose B_1 with probability 1/2, and B_2 with probability 1/2. The expected payoff of the first player is 1 since he has 1/2 probability of winning and the total profit is 2. On the other hand, the second player has an expected profit of ≈ 2.

More generally, *fake bidding* is the strategy used by adversarial searchers to fake bids in order to outbid competitors in the FB block simulation. This, would in expectancy, increase their revenue. This strategy can be used after boosting their address reputation to increase its impact. Moreover, fake bidding strategy can be generalized with multiple accounts, to ensure winning the bundle and minimizing the payment with high probability.

6.5 Theoretical analysis

Assuming that, the Flashbots relayer does a unique simulation per block/bundle (this can be generalized with more simulations). The adversary have n accounts with high reputation queue. Let $p \in [0, 1]$. Using the special variables, we can construct a random variable such that the bid b is b_{max} with probability p and b_{min} with probability $(1 - p)$. Assume that b_{max} is the exact value of the MEV opportunity. Assume that a set of searchers $\mathcal{P} = \{1, ..., n\}$ are trying to capture the same common knowledge MEV opportunity of gross value v through Flashbots. Let B be the bundle of the player using this strategy and $B_1, ..., B_n$ the bundles of $1, ..., n$ respectively.

Let b_{max} be bid minimum bid such that $s(B) > \max_i s(B_i)$. In this setting, we have that the expected payoff of playing this strategy is

$$\mathbb{E}(u \mid p, n, b_{min}, b_{max}) = (1 - (1 - p)^n)(p(v - b_{max}) + (1 - p)(v - b_{min}))$$

From this equation, it follows immediately the following result.

Lemma 6.7. Assume that all players and in high reputation queue, then:

$$\lim_{p \to 0} \lim_{n \to +\infty} \mathbb{E}(u \mid p, n, b_{min}, b_{max}) = (v - b_{min}) \tag{15}$$

In reality, we see that players that try to capture common knowledge expend similar gas cost for extracting the opportunity. This, makes the estimation of b_{max} feasible. Moreover, in reality, we see that the common knowledge MEV opportunities are nearly completely paid to the miner. For example, we see that Arbitrages using Uniswap V2 protocol or sandwich attacks pay around 99% of the gross profit to the miner. A theoretical explanation of this is given in the following proposition. We prove that, if all players bid through gas price without using fake bidding strategy, and have identical gas efficiencies, then the all Nash equilibrium lead to maximum bid.

Proposition 6.8. Define the strategy of a searcher gas price bidding m be $s(m)$. Assume that gas costs induced by winning the MEV opportunity are $g_1 = ... = g_k = g$. Then, let $\mathbb{G}$ be the subgame of the Flashbots auction game with MEV opportunity with value v being public information where players just use gas price. Then, the set of Nash equilibrium is:

$$\{(s_1, ..., s_n) \in S_1 \times ... \times S_n : \exists i, j \text{ s.t. } s_i = s_j = s(v/g)\}$$

where S_i is the set of all strategies of player i. If players follow the best response strategy, they will converge to the equilibrium

$$(s_1(v/g), ..., s_n(v/g)) .$$

The set of Nash equilibrium is invariant under risk-lover, risk-neutral and risk-averse agents. In particular, in equilibrium, the all the value v is paid to the validator.

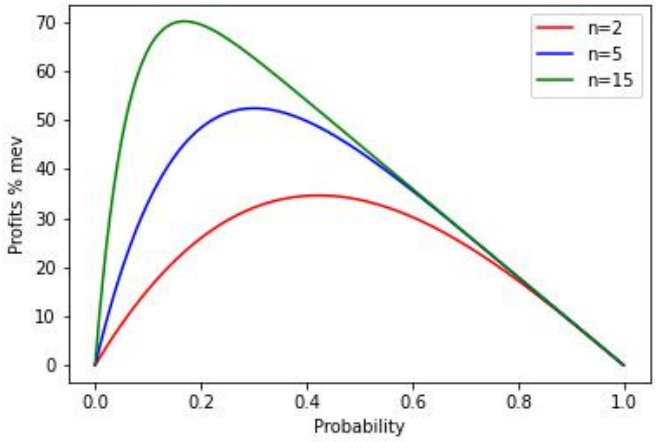

Figure 5: Profits of the Strategy

However, if a player can use the red strategy, we obtain a completely different results.

Theorem 6.9. Assume that PoW Flashbots builder follows the algorithm 1. Then, the red strategy strictly dominates honest bidding. Moreover, the Flashbots builder is not Sybil resistant and does not maximize validators' profits.

Proof. Assume wlog that players $i = 2, ..., n$ follow an honest strategy. Therefore, their payment $\mathbf{pr}_i \leq v$, since bidding more than v is a strictly dominated strategy, since lead to negative revenue. Let $v_{max} = \max_{j\neq 1}\{v_j\}$. Let's define a fake bidding strategy with $b_{min} = 0$ and $b_{max} = v_{max} + \varepsilon$ with probability p. This strategy leads to $p^2(v - v_{max} - \varepsilon) + p(1-p)v$ profits. For $v_{max} > v/2$ the strategy strictly dominates any other honest bidding strategy. Otherwise, set the fake bidding strategy with $b_{max} = b_{min}$. □

6.6 Game theoretical analysis of the strategy

In previous sections, we have seen that the strategy dominates honest bidding. Moreover, with enough accounts in the high priority queue, the profits of the searchers converge to all the MEV. In this section, we will analyze the game where all players are aware of the Fake bidding strategy.

PoW Flashbots auction game: We will model this game as a MEV game [12]. We have a set of players $\mathcal{P}$, we assume that players do not see each other's bids and the auction mechanism is the Flashbots mechanism. In the Flashbots subgame without fake bids, players must choose a bid b. However, in the Flashbots game players can choose a positive random variable X, called bidding variables, that determinate its priority in the block construction and its payment to the miner. That is, first there is an instance of the bidding variables to chooses which bundle is executed. Then, there is a second instance that chooses the payment of the winner of the simulation with her respective bidding variable. More precisely, if a set of players $\mathcal{P}$ with same gas efficiency bid for an MEV of net value v with bidding variables $X_1, ...X_n$. Then, the utility of the first players is

$$u_i(X_i, X_{-i}) = \Pr[X_i > \max_{j\neq i}\{X_j\}]\,(v - \mathbb{E}(X_i))\,. \tag{16}$$

Theorem 6.10. Let $\mathbb{G}$ be the Flashbots game and $\mathcal{P}$ a set of n players extracting an MEV opportunity of net value v. Assume that all players share the same gas costs for extracting the MEV opportunity. Then, there exist a Nash equilibrium S such that $u_i(S) > 0$ for all i. In other words, Flashbots auction does not maximize revenue.

Proof. Assume that $S = (X_1, ..., X_n)$ is a Nash equilibrium. Moreover, let's assume that $u_1(S) = 0$. Then, we have that $u_1(b, X_{-i}) = \Pr[b > \max_{j \neq i}\{X_j\}](v - b)$ for all $b \in \mathbb{R}_{\geq 0}$. Since, S is a Nash equilibrium, we have that, for all $b \geq v$, we have that $u_1(b, X_{-i}) = 0$. Therefore, we have that $\Pr[v > \max_{j \neq 1}\{X_j\}] = 0$. On the other hand, since S is a Nash equilibrium, we have that $\Pr[v < \max_{j \neq 1}\{X_j\}] = 0$, otherwise, there exist a player i with negative utility. So, we have that $\Pr[v < \max_{j \neq 1}\{X_j\}] = 1$. Now, consider the strategy given by the binomial variable $X = (v + \varepsilon)\mathrm{B}(2, p)$ with probability p. Clearly, $u_1(X, X_{-i}) > 0$ leading to a contradiction since S is a Nash equilibrium. □

6.7 Empirical anaylsis

To prove the assumption that bundles are simulated once, we made the following. We deployed a smart contract in Goerli network with two functions, `RandomBid` and `NotRandomBid`. To execute the functions, both require that a variable `bool` is True, otherwise the transaction reverts. Once executed, the functions change `bool` to False. Therefore, both transactions conflict. The function `NotRandomBid` pay the 0.0015 of Goerli ETH. While the `RandomBid` function bids 0.003 or 0.[12] We repeated this experiment 128, landing a total of 55 `RandomBid` and 73 `NonRandomBid`[13]. The 55 `RandomBid`, 32 were bidding the minimum. Therefore, we obtain that the 25% the random bid strategy outbid the normal one but with less ex-post effective gas price.

7 Data analysis on Price of MEV

To evaluate agents' behavior in different MEV games, we adopt a data-drive methodology. We describe our data collection in this section, as well as our methods for extracting insights from these data. In the service of open software and open science, we make our datasets and collection code openly available [11]. To compare different MEV games, our data collection focused on different domains in different block ranges.

7.1 Measuring Price of MEV in different domains

8 Conclusions

9 Future work

Our study focused on games with one block common knowledge MEV opportunities. Moreover, we focused the study of price of MEV in the block space cost. First, we believe that a deeper study can be made over the MEV games with a Bayesian game point of view, where players drawn the local MEV from random and independent variables. Second, we focused our study in the block space price of MEV leaving other social cost functions out of the scope of this paper. In the future, we will compute the price of MEV of more involved social cost functions. Third, we studied these games as stage games, assuming that the game is not played more than once. In future work, we will study these different MEV games in a repeated game, where players can cooperate, collude and punish each other to maximize their payoff.

- Repeated games - General Sybil resitent Nash equilibrium - Aequitas protoocol. - Non-common knowledge mev opportunities. - Centralization costs. - Multi block mev games, oracle manipulation and reorgs.

10 Acknowledgments

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.

[12] To be sure that both transaction were not rejected by the Flashbots relayer, we made more operations to at least expend 42000 gweis.

[13] More details in GrimmBrothers GitHub.

[2] Guillermo Angeris, Alex Evans, and Tarun Chitra. "A Note on Bundle Profit Maximization". In: Stanford University, 2021.
[3] Kushal Babel et al. "Clockwork finance: Automated analysis of economic security in smart contracts". In: *arXiv preprint arXiv:2109.04347* (2021).
[4] Vitalik Buterin et al. "Combining GHOST and Casper". In: (Mar. 2020). DOI: 10.48550/arxiv.2003.03052. URL: https://arxiv.org/abs/2003.03052.
[5] 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.
[6] 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.
[7] Ethereum. *Sort transactions at the same gas price by received time*. https://github.com/ethereum/go-ethereum/pull/21358.
[8] Takuya Iimura and Takahiro Watanabe. "Pure strategy equilibria in finite symmetric concave games and an application to symmetric discrete Cournot games". In: *Equilibrium Theory for Cournot Oligopolies and Related Games*. Springer, 2016, pp. 89–100.
[9] Mahimna Kelkar et al. "Order-fairness for byzantine consensus". In: *Annual International Cryptology Conference*. Springer. 2020, pp. 451–480.
[10] Vijay Krishna. *Auction theory*. Academic press, 2009.
[11] Bruno Mazorra and Michael Reynolds. *MEV games*. 2022. URL: https://github.com/BrunoMazorra/Flashbots-MEV-game.
[12] Bruno Mazorra, Michael Reynolds, and Vanesa Daza. "Price of MEV: Towards a Game Theoretical Approach to MEV". In: *arXiv preprint arXiv:2208.13464* (2022).
[13] Alexandre Obadia et al. "Unity is Strength: A Formalization of Cross-Domain Maximal Extractable Value". In: *arXiv preprint arXiv:2112.01472* (2021).
[14] Ulrich Pferschy and Joachim Schauer. "The Knapsack Problem with Conflict Graphs." In: *J. Graph Algorithms Appl.* 13.2 (2009), pp. 233–249.
[15] Tim Roughgarden. "Intrinsic robustness of the price of anarchy". In: *Journal of the ACM (JACM)* 62.5 (2015), pp. 1–42.
[16] Tim Roughgarden. *Selfish routing and the price of anarchy*. MIT press, 2005.
[17] Roy J Ruffin. "Cournot oligopoly and competitive behaviour". In: *The Review of Economic Studies* 38.4 (1971), pp. 493–502.
[18] Danny Ryan et al. *consensus-specs/validator.md at dev · ethereum/consensus-specs · GitHub*. 2022. URL: https://github.com/ethereum/consensus-specs/blob/dev/specs/phase0/validator.md.
[19] Harvey M Salkin and Cornelis A De Kluyver. "The knapsack problem: a survey". In: *Naval Research Logistics Quarterly* 22.1 (1975), pp. 127–144.
[20] Alejo Salles. "On the Formalization of MEV". In: *https://writings.flashbots.net/research/formalization-mev/* (2021).
[21] 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.
[22] *The All-Pay Auction with Complete Information*. June 1990. URL: https://pure.uvt.nl/ws/portalfiles/portal/1145032/MBDKCV5620413.pdf.

11 Appendix

11.1 Notation table

Table 2: Notation

$\texttt{ev}$	=	extraction value
$\texttt{rv}$	=	miners revenue
m	=	gas price
Δ	=	latency of gossip network
Δ_i	=	latency of player i
Δb	=	increment of tokens balance
$\mathcal{P}$	=	set of players
Pr	=	probability measure
$\mathbb{E}$	=	expectancy
u_i	=	utility function of player i
b	=	tokens balance
γ_i	=	gas efficiency of reverted transactions of searcher i
h	=	hash function
A	=	set of valid actions
$\mathcal{D}$	=	domain
$\texttt{prt}$	=	protocol
$\mathcal{P}(A)$	=	set of all subsets of A
supp X	=	support of a random variable X
$\texttt{st}$	=	state of domain $\mathcal{D}$
s	=	strategy
S	=	tuple of strategies

11.2 MEV games proofs

Lemma 11.1. Let $X \sim \text{Exp}(\lambda)$, then $\mathbb{P}(X \in [t, t+\Delta) \mid X \geq t) = 1 - e^{-\lambda\Delta}$.

Proof: Using the definition of conditional probability, we have,

$$\Pr(X \in [t, t+\Delta) \mid X \geq t) = \frac{\mathbb{P}(X \in [t, t+\Delta])}{\mathbb{P}(X \geq t)} = \frac{\int_t^{t+\Delta} \lambda e^{-\lambda x}\, dx}{\lim_{a\to\infty} \int_t^a \lambda e^{-\lambda x}\, dx} = \frac{e^{-\lambda t} - e^{-\lambda(t+\Delta)}}{e^{-\lambda t}} = 1 - e^{-\lambda\Delta}.$$

Proof 4.2: Assume $S_1 = S_{BR}$ (with $m_0 = m_{\min}$ here w.l.o.g.) or $S_1 = S_{RC}$, and $u_1(S_1, \varnothing) \geq \eta(1+\tau)^k g_1$, where $\frac{1}{(1+\tau)^k g_1} \gg \eta > 0$ for all k, giving $\eta(1+\tau)^k g_1 \approx 0$. Let $S_2 = S_2(k\phi) = (m_{\min} + \eta)(1+\tau)^k$ (a BR strategy) where $\phi = \delta$ if $S_1 = S_{BR}$ and $\phi = \Delta_1$ if $S_1 = S_{RC}$. We use induction on k to show: for each $k \in \mathbb{N}$, at time $t = k\phi$ we have $S_1 = m_{\min}(1+\tau)^k < S_2 = (m_{\min} + \eta)(1+\tau)^k$, and $S_1 < S_2$ for $t \in [0, k\phi]$.

When $k = 0$, $S_1 = m_{\min}(1+\tau)^0 = m_{\min} < S_2 = (m_{\min} + \eta)(1+\tau)^0 = m_{\min} + \eta$. So, $k = 0$ holds.

Assume k holds. That is, at $t = k\phi$, $S_1 = m_{\min}(1+\tau)^k < S_2 = (m_{\min} + \eta)(1+\tau)^k$, and $S_1 < S_2$ for $t \in [0, k\phi]$.

It follows that, for $k+1$, at $t = (k+1)\phi$, if $S_1 = S_{RC}$ then $\mathcal{P}_1$ observes $\mathcal{P}_2$'s bid $S_2(k\phi) = (m_{\min} + \eta)(1+\tau)^k > S_1(k\phi) = m_{\min}(1+\tau)^k$ and immediately bids $S_1((k+1)\phi) = \max\{m_1 \times (1+\tau), m_0 + \eta\} = m_{\min}(1+\tau)^{(k+1)}$ (since $(m_{\min} + \eta)(1+\tau)^k + \eta \approx m_{\min}(1+\tau)^k < m_{\min}(1+\tau)^{(k+1)}$) while $\mathcal{P}_2$ bids $S_2((k+1)\phi) = (m_{\min} + \eta)(1+\tau)^{(k+1)}$ on schedule. $S_1 = S_{BR}$ bids the same at $t = (k+1)\phi$ without the observation.

So, when k holds, it follows that at $t = (k+1)\phi$, $S_1 = m_{\min}(1+\tau)^{(k+1)} < S_2 = (m_{\min} + \eta)(1+\tau)^{(k+1)}$, and $S_1 < S_2$ for all $t \in [0, (k+1)\phi]$. This proves the inductive step. Since $k = 0$ and the inductive step hold, the statement holds for all $k \in \mathbb{N}$ by induction.

Hence, $S_2(t) > S_1(t)$ for all t, and $u_1(S_1, S_2) = 0$ if $\gamma_1 = 0$, and $u_1(S_1, S_2) < 0$ otherwise. That is, $u_1(S_1, S_2) \le 0$ for all t. By assumption, we have $\texttt{ev} - m_{\min}(1+\tau)^k g_1 \ge \eta(1+\tau)^k g_1$. That is, $\texttt{ev} - g_1(m_{\min} + \eta)(1+\tau)^k \ge 0$. Then, since $g_2 \le g_1$, there exists η where $u_2(S_2, S_1) \ge 0$. This concludes the proof.

Proof 4.7 Clearly, any deviation below $m + \eta$ (where $\eta \ge 0$ is the minimum gas bid tick of the protocol) is suboptimal except bidding 0. Therefore, we have to check if the $\varnothing$ strategy and deviation strategy (bidding $m + \varepsilon$ for some $\varepsilon > \eta$,) are optimal answers. Observe that the $S_C(m)$ strategy is a Nash equilibrium if and only if, for all t,

$$\mathbb{E}(r_0 \mid (r_0, ..., r_n) \leftarrow \texttt{Exec}(S_C(m); \Delta, g, \gamma) \mid X \ge t) \ge \mathbb{E}(r_0 \mid (r_0, ..., r_n) \leftarrow \texttt{Exec}(\texttt{Deviate}; \Delta, g, \gamma) \mid X \ge t) \quad (17)$$

Using Lemma 11.1 to obtain the RHS probabilities, equation 17 is equivalent to

$$(\texttt{ev} - g_i m)\frac{1}{n} - \gamma_i m \frac{n-1}{n} \ge (\texttt{ev} - g_i(m+\varepsilon))(1 - e^{-\lambda \Delta_{i,\min}}) - \gamma_i(m+\varepsilon)e^{-\lambda \Delta_{i,\min}}$$

where $\Delta_{i,\min} = \min\{\Delta_j : j \ne i\}$. Provided $\texttt{ev} - m g_i > -m\gamma_i$ (a reasonable assumption), this is equivalent to,

$$\frac{1}{n} \ge 1 - e^{-\lambda \Delta_{i,\min}} - \varepsilon\left(\frac{g_i(1 - e^{-\lambda \Delta_{i,\min}}) + \gamma_i e^{-\lambda \Delta_{i,\min}}}{\texttt{ev} + m_i(\gamma_i - g_i)}\right) \quad \text{for all } \mathcal{P}_i.$$

Now, the bracketed formula in the ε term is always positive. So, if the inequality holds for $\varepsilon = \eta$, it holds for all $\varepsilon > \eta$. Hence, equation 17 is equivalent to,

$$\frac{1}{n} \ge 1 - \exp(-\lambda \Delta_{\min}) - O(\eta) \quad \text{for all } \mathcal{P}_i,$$

where $\Delta_{\min} = \min\{\Delta_i\}$. Similarly, observe that the $S_C(m)$ strategy dominates the $\varnothing$ strategy if and only if,

$$\frac{1}{n}(\texttt{ev} - g_i m) - \frac{n-1}{n}\gamma_i m \ge 0, \quad \iff \frac{\texttt{ev}}{g_i + \gamma_i(n-1)} \ge m \quad .$$

This concludes the proof.

Proof4.8: Let $\mathcal{P}_i \ne \mathcal{P}_1$. Recall the notation **??** $u_i(S_i, S_{-i})$. If a searcher $\mathcal{P}_i$ deviates and tries to gas bid $m_i > m$, their expected payoff is,

$$u_i(m_i, S_{-i}) = (1 - e^{-\lambda \Delta_1})(ev - g_i m_i) - e^{-\lambda \Delta_1}\gamma_i m_i.$$

Now, suppose that $u_i(m_i, S_{-i}) \le 0$ for all $\mathcal{P}_i$. This is equivalent to,

$$m_i \ge \frac{\texttt{ev}(1 - e^{-\lambda \Delta_1})}{g_i(1 - e^{-\lambda \Delta_1}) + \gamma_i e^{-\lambda \Delta_1}}, \text{ where } \frac{\texttt{ev}(1 - e^{-\lambda \Delta_1})}{g_i(1 - e^{-\lambda \Delta_1}) + \gamma_i e^{-\lambda \Delta_1}} \le m_{\min} = \frac{\texttt{ev}(1 - e^{-\lambda \Delta_1})}{g_{1,\min}(1 - e^{-\lambda \Delta_1}) + \gamma_{1,\min} e^{-\lambda \Delta_1}},$$

for all $\mathcal{P}_i$. Therefore, we can say that, if $m \ge m_{\min}$ then $u_i(m_i, S_{-i}) \le 0$ for all $\mathcal{P}_i$, and the searchers $\mathcal{P}_i$ will follow the null strategy. Now, searcher $\mathcal{P}_1$ does not have an incentive to bid less than $m_{\min}$, since the other searchers will follow the cooperative strategy and outbid their bid. So, searcher $\mathcal{P}_1$ will at least bid $m_{\min}$, using the censor strategy.

Proof 4.17: Assume that at time $t = 0$, searchers $\mathcal{P}_i$ have access to a public correlating device $D \in \mathbb{N}$ where,

$$\Pr[D = i] = \frac{v_i}{\texttt{ev} - g_i m}, \text{ for all } i, \quad \text{and } \Pr[D = 0] = 1 - \sum_{i=1}^{n} \frac{v_i}{\texttt{ev} - g_i m} \quad .$$

Let S be the tuple of strategies that will lead to a correlated Nash equilibrium. If the output of D is a positive integer i, then let $S = S_+ = S_i \cup \bigcup_{j \ne i} S_j$ where $S_i = S_{CE}(m)$ and $S_j = \varnothing^\star(m, S_C, \Delta_j)$. If the correlating device outputs zero, then let $S = S_0 = \bigcup_i^n S_{CE}(0)$. Now, for all i, we have,

$$\mathbb{E}(u_i(S) \mid D) = u_i(S)\Pr[D = i] + u_i(S)\Pr[D \ne i] = (\texttt{ev} - g_i m)\frac{v_i}{ev - g_i m} = v_i \quad .$$

That is, $\boldsymbol{u}(S) = \boldsymbol{v}$. Given S_+ can be mapped to the strategy tuple in proposition 4.8, and that our chosen $m \geq m_{\min}$ defined there, it follows that this correlated strategy is a Nash equilibrium. S_0 can be interpreted as searchers not choosing to extract the $\mathtt{ev}$, which is a Nash equilibrium by definition.

Proof 4.12: Analogue to proposition 1 of [22].

Proof 4.19: First, let's assume that all players send a transaction and fix a player i. If the others searchers invest l_{-i} resources, then the best response (maximum of $u_i(l_i, l_{-i})$) is $l_i = \sqrt{(\mathtt{ev} + (\gamma-1)b_{min})l_{-i}} - l_{-i}$. The unique solution of the system of equations is $l_i = \frac{n-1}{n^2}(\mathtt{ev} + (\gamma-1)b_{min})$. The utility u_i is positive if and only if $\mathtt{ev} \geq b_{min} + (n-1)\gamma b_{min}$.

Proof 6.8: Let $F_1, ..., F_n$ be the distribution functions of a mixed Nash equilibrium $\sigma = (\sigma_1, ..., \sigma_n)$. Assume there exists i, j such that $u_i(\sigma), u_j(\sigma) > 0$. Observe that

$$u_i(x) = (\mathtt{ev} - g_i x)A_i(x), \text{ for all } x \in \operatorname{supp}(\sigma_i) \tag{18}$$

where $A_i(x) = \Pr[\bigcup_{j=1, j\neq i}^{n} \{\sigma_j < x\}]$.

Step 1: $\underline{s}_i = \underline{s}_j =: \underline{s}$.

Step 2: $\underline{s}$ is not a mass point of F_i and F_j.

Step 3: $F_i(\underline{s}) \neq 0$ contradiction.

At most exist a player i with positive utility.

If $k \geq 1$

Notation 11.2. Let F be a function, we define $F^-(y) := \lim_{x \to y^-} F(x)$.

Proof 4.18: Let's fix a player i and assume that other players have send a total of x_{-i} transactions to extract the MEV opportunity. Then, if the player i sends x_i the transaction, the expected payoff is:

$$u_i(x_i, x_{-i}) = \frac{x_i}{x_i + x_{-i} + 1}(\overline{\mathtt{ev}} - (x_i - 1)c) - x_i c\left(1 - \frac{x_i}{x_i + x_{-i} + 1}\right) \tag{19}$$

$$= \frac{x_i}{x_i + x_{-i} + 1}(\overline{\mathtt{ev}} - x_i c) - x_i c \tag{20}$$

where $\overline{\mathtt{ev}} = \mathtt{ev} - gm$ and $c = \gamma m$. Assume for now that x_i, x_{-i} can be any positive real number. One can check that $\frac{\partial^2 u_i(x_i, x_{-i})}{\partial x_i^2} < 0$ and that $\nabla u = 0$ has a unique solution $\overline{x}$, and this solution is a diagonal solution. This proves that if the space of actions is $\mathbb{R}_{\geq 0}$, then there is a unique NE and is the solution of $\nabla u = 0$. However, the action space is discrete (players have to choose the number of transactions they send). To prove that the NE is in a set V_z we will invoke the theorem 3.1 of [8]. We have seen already that the game is concave and by construction is symmetric. Therefore, we need to check that conditions 1) and 2) hold. These conditions hold for $n >> 1$, leading to the desired result. Moreover, one can prove that $z = \lfloor \overline{x} \rfloor$. If $n >> 1$, then $z = 0$ and there exists i such that $s_i = 0$. Let Y be the number of j such that $s_j \neq 0$, then

$$\frac{\overline{\mathtt{ev}}}{Y + 1} - \gamma m < 0,$$

so $Y \geq \overline{\mathtt{ev}}/\gamma m - 1$. Then, we deduce that the PoA is $\Omega(\frac{\mathtt{ev} - gm}{\gamma m})$.

11.3 Proof 4.22

Assume that $(\sigma, ..., \sigma)$ is a symmetric Nash equilibrium. Let F be the cumulative distribution functions of the strategies, and assume that F is continuous, therefore $F^-(x) = F(x)$. Now assume that the player P_1 bids x, then the expected payoff is

$$\mathbb{E}(u_i | b_i = x) = p(\mathtt{ev} - x) \Pr[x \text{ is the highest bid}] \tag{21}$$

$$= p(\mathtt{ev} - x)\left(\sum_{i=0}^{n} \binom{n-1}{i} p^i (1-p)^{n-1-i} F(x)^i\right) \tag{22}$$

$$= p(\mathtt{ev} - x)(pF(x) + 1 - p)^{n-1} \tag{23}$$

Since F is a Nash equilibrium, for all $x, y \in \text{Supp } \sigma$ holds $\mathbb{E}(u_i|\$b = x) = \mathbb{E}(u_i|\$b = y)$.
Claim: Supp $\sigma \subseteq [0, \overline{s}]$ for some $\overline{s} < 1$.
Assume that the claim does not hold. Then exists $\{x_n\}_{n\in\mathbb{N}} \subseteq \text{Supp}\sigma$ such that $x_n \to 1$. We deduce that

$$C = p(\texttt{ev} - x_k)((1-p) + pF(x_k))^{n-1}$$

for all $k \in \mathbb{N}$. Taking limits, we deduce that $C = 0$. Therefore, we would have that the $\mathbb{E}(u_1 \mid (\sigma, ..., \sigma)) = 0$. On the other hand, the $\mathbb{E}_i(u_i \mid (0, \sigma, ...\sigma)) \geq p(1-p)\texttt{ev}$, leading to a contradiction.
Now, let's prove that there does not exist any better response strategy. To do so, it is enough to prove that there is no bid $\$b > \overline{s}$ such that $\mathbb{E}(u_1 \mid (\$b, \sigma, ..., \sigma)) > \mathbb{E}(u_1 \mid (\sigma, ..., \sigma))$. We have that for any $\$b > \overline{s}$, it holds $\mathbb{E}(u_1 \mid (\$b, \sigma, ..., \sigma)) = p(\texttt{ev} - \$b)$. Also, we have that

$$\mathbb{E}(u_1 \mid (\sigma, ..., \sigma)) = p(\texttt{ev} - \overline{s}) \tag{24}$$

So, we deduce that the player has no incentives to deviate. Also, we must have that $F(0) = 0$, otherwise we would have that $\frac{1}{n}p\texttt{ev}F(0) = \lim_{\varepsilon\to 0^+} \mathbb{E}(u \mid (\varepsilon, ..., \sigma)) = \mathbb{E}(u \mid (0, \sigma, ..., \sigma)) = p\texttt{ev}F(0)$, leading to a contradiction. One can easily prove that $0 \leq F(x) \leq 1$ for all x. Moreover, F is increasing and right continuous, therefore, F is a cumulative distribution function of a random variable.

11.3.1 Characterization

Let $\mathcal{N} = \{p_1, ..., p_n\}$ be a set of players and $F_1, ..., F_n$ be the cumulative distribution functions of a Nash equilibrium.

Notation 11.3. We make the following notations.

- $F^{i_1,...,i_k} = \prod_{l=1}^{k} F_{i_l}$.
- The probability that i wins the bid if he finds ev is $A_i = \sum_{k=1}^{n-1} \sum_{\{i_1,...,i_k\}\subseteq\mathcal{N}\setminus\{i\}} p^k(1-p)^{n-1-k}F^{i_1,...,i_k}$. Observe that for $p = 1$, we have that $A_i = F^{1,...,n}/F_i$.

Observation 11.4. $A_i(x) \leq A_j(x)$ implies that $F_j(x) \leq F_i(x)$.

Lemma 11.5. Let $F_1, ..., F_n$ be Nash equilibrium in the Flashbots auction, then:

1. If $\overline{s}_i = \texttt{ev}$ for some i, then the Nash equilibrium is the one determined in the proposition 6.8.
2. If $\overline{s}_i < \texttt{ev}$ for all i, then $\overline{s}_i = \overline{s}$ for all i.
3. If $\overline{s}_i < \texttt{ev}$ for all i, then $\underline{s}_i = \underline{s}$ for all i.
4. $\underline{s_i} = 0$ for all i.

Proof. 1. Assume that $\overline{s}_i = \texttt{ev}$ for some i. Then, the expected player utility is 0. If all the other players have $\overline{s}_j < \texttt{ev}$, then we would have that exist $\varepsilon > 0$ such that $\texttt{ev} - \varepsilon > \overline{s}_j$. Clearly, if player i plays $\texttt{ev} - \varepsilon$, would have a positive utility, leading to a contradiction. So exist another player with strategy ev.

2. If there is an $\overline{s}_i < \texttt{ev}$ then for all j we have that exist a neighborhood $(\varepsilon_j - \overline{s}_i, \varepsilon + \overline{s}_i)$ such that F_j is constant (otherwise the right and left limits of the utility would not coincide, leading to a contradiction). Therefore, if F_j is constant for all j in this neighborhood, the player i, will decrease $\overline{s}_j$ maximizing his payoff, leading to a contradiction.

3. Analogous.

4. Trivial.

□

Lemma 11.6. The expected payoff is the same for every player, i.e. $u_i^\star = u_j^\star$ for all i, j.

Proof. Suppose $u_i^\star < u_j^\star$. Then $u_i^\star < u_j^\star = u_j(\overline{s}, F_{-j}) \leq \lim_{x\to\overline{s}} u_i(x_i, F_{-i})$, which leads to a contradiction. □

Lemma 11.7. There are no mass points on $[0, \texttt{ev}]$.

Proof. Assume that a F_i has a mass point in $x_i \in (0, \overline{s}]$. First, let's assume that $x_i \neq 0$. Since $A_{i,j}$ is an increasing function, we have that $F_i A_{i,j}$ has an upward jump at x_i. That is $\lim_{x_i \to x^-} A_j < \lim_{x_i \to x^+} A_j$. On the other hand, there exists j such that F_j is not constant in a neighborhood $(x_i - \varepsilon, x_i + \varepsilon)$ of x_i, otherwise would not be a Nash equilibrium since the player i will move the mass point x_i to $x_i - \varepsilon/2$ to increase it's expected payoff. Now let j be an element that is non-constant in a neighborhood of x_i. Then, for all $x \in \text{Supp}\sigma_i \cap (x_i - \varepsilon, x_i + \varepsilon)$, we have that

$$\mathbb{E}(u_j \mid \$b_j = x) = p(\texttt{ev} - x)A_j(x) \tag{25}$$

is constant, but this does not hold since A_j has a discontinuity in x_i. Therefore, there are no mass points with $x > 0$. Now let's prove it for $x = 0$. First assume that there are two players with a mass points in $x = 0$, then one player has incentives to deviate since if both players bid 0 they will split $\texttt{ev}$ probabilistically, while deviating with bid ε would take the $\texttt{ev}$. Now if just a player is playing it, he will always lose the auction, therefore would have incentives to move its mass point to $\overline{s} + \varepsilon$ to increase its utility. □

Lemma 11.8. $F_i(x) = F_j(x)$ for all i, j and $x \in \text{Dom}F_i \cap \text{Dom}F_j$ such that x is a point of increase of F_i and F_j.

Proof. Let $B_i(x) = p(\texttt{ev} - x)A_i(x)$. We know that this function is constant in the strictly increasing points (since this functions coincide with the expected utility in $x \in \text{Supp } \sigma_i$. Therefore, from lemma 11.6 we deduce that $A_i = A_j$. Using the observation 11.4, one can deduce that this implies that $F_i = F_j$. □

Lemma 11.9. If F_i is strictly increasing in some open interval (a, b) then F_i is strictly increasing in $(a, \overline{s}]$.

Proof. Wlog, assume that there exist an interval $(b, b + \varepsilon)$ such that F_i is not strictly increasing. Then, we have that $F_i(b) = F_i(b + \varepsilon)$. Clearly, exists k, l such that F_k, F_l are strictly increasing in $(b, b + \varepsilon)$ (otherwise a player change its distribution function to increase its revenue). Also, we have that $B_l(b) = B_i(b)$ by 11.6. Since $x \in (b, b + \varepsilon)$ does not lie in the i's strategy support, we have that $B_i(x) \leq B_l(x)$, implying that $A_i(x) \leq A_l(x)$, and hence that $F_l(x) \leq F_i(x)$, a contradiction with the fact that F_h is increasing, F_i is constant and $F_i(b) = F_l(b)$.

□

Proof 4.23 : **Case 1** $n \to +\infty$: Observe that the expected revenue of the miner is $\mathbb{E}(\max\{\$b_1, ...\$b_n\})$. Fix $\varepsilon > 0$ such that $1 - \varepsilon < \overline{s}$.

$$\lim_{n \to \infty} \Pr(\max\{\$b_1, ...\$b_n\} < 1 - \varepsilon) = \lim_{n \to \infty} \Pr(\$b < 1 - \varepsilon)^n = 0$$

Therefore $\mathbb{E}(\max\{\$b_1, ...\$b_n\}) \geq 1 - \varepsilon$. Since $\overline{s} \to \texttt{ev}$ when $n \to +\infty$, we have that $\lim_{n \to \infty} \mathbb{E}(\max\{\$b_1, ...\$b_n\}) = 1$.

Case 2 $p \to 1$: We have that $|F(x) - \texttt{ev}|_2 \to 0$ if $p \to 1$. Then, $\mathbb{E}[u_i] \to 0$. Since $\texttt{rv} = \texttt{ev} - \sum_{i=1}^{n} \mathbb{E}[u_i]$, we deduce that $\texttt{rv} \to \texttt{ev}$.