Stackelberg Attack on Protocol Fee Governance

Alex Hajjar*
Butter

February 2024

Abstract

Abstract. We establish a Stackelberg attack by Liquidity Providers against Governance of an AMM, leveraging forking and commitments through a Grim Forker smart contract. We produce a dynamic, block-per-block model of AMM reserves and trading volume in the presence of competing forks, derive equilibria conditions in the presence of protocol fees, and analyze Stackelberg equilibria with smart contract moves.

Contents

1 Introduction **2**

2 Model **3**
- 2.1 Automated Market Maker . . . 3
- 2.2 Trader volume allocation . . . 4
- 2.3 LP reserves allocation . . . 6
- 2.4 Aggregate allocation . . . 6

3 Equilibria analysis **8**
- 3.1 No fee scenario: network effects . . . 8
- 3.2 Leader with fee scenario: equilibrium condition on fee . . . 9

4 Stackelberg attack **10**
- 4.1 Governance player . . . 11
- 4.2 Grim Forker contract . . . 11
- 4.3 Stackelberg equilibrium . . . 12
- 4.4 Market-sourced equilibrium condition . . . 13

5 Conclusion **14**

*lajarre@buttery.money

1 Introduction

We introduce the concept of an *absolute commitment*, where agents have a more 'absolute' ability to commit to strategies than is usually the case in games. In particular

Protocol fees are used by diverse DeFi protocols as a means to allow value to be captured by the governing DAO [1, 5]. Fees can then be used to produce dividends for token holders, or to accumulate treasury value [6]. In both cases, they allow the governance token to gain value, thus justifying investability in the protocol. Importantly, some major protocols, like Uniswap, haven't activated the protocol fee to date.

Protocol fees are taken from some protocol participants revenue. In the case of Uniswap, protocol fees are deduced from Liquidity Provider (LP) fees [1]. This dynamic creates an adversarial setting where LPs have diverging interests from the DAO (Governance).

Related work. [8] produces conditions on Stackelberg equilibria between stakeholders of a stablecoin system, including governance as a player and protocol parameter setting as action space. Such models are still to be studied in context of AMM protocol fee governance. In this paper, we aim at producing a framework for deriving Stackelberg equilibria between LPs allocating reserves among an AMM and its fork and Governance of the original AMM setting a protocol fee.

[7] introduces contracts as Stackelberg equilibria where smart contract moves are introduced in a game-theoretic setting. Drawing on it, [9] studies some Stackelberg attacks based on commitments enabled by smart contracts. We define Grim Forker smart contracts as a case of Stackelberg attacks. Such contracts are used by LPs to change the equilibria in their favor, to the detriment of Governance.

We focus on Constant Function Market Makers. We briefly introduce the core components of these systems and refer to [2] for further details.

This paper. We formalize a game-theoretic model of competition among two forks of the same Automated Market Maker (AMM) pool, whereby we observe rational behavior of AMM users (LPs, traders and Governance).

We study these competition games in a in a dynamic system setting, block-per-block, with protocol fee rate and on initial conditions as parameters.

We finally adjunct a metagame played by LPs against Governance, where they can commit to strategies that automatically deploy their reserves among the competing forks, depending on conditions. We introduce the Grim Forker contract as an instance of a Stackelberg attack.

This produces new Stackelberg equilibria to the competition game. This allows us to define which protocol-fee-setting strategies Governance should rationally follow.

This paper will be limited to the analysis of a single pool in an AMM like Uniswap v2, but the results can be generalized to multiple pools and to Uniswap v3-like AMM tick-per-tick.

2 Model

2.1 Automated Market Maker

We define an Automated Market Maker as being a 2-token pool, similar to Uniswap v2.

Definition 1 (Automated Market Maker (AMM)). *An Automated Market Maker (AMM) is defined by the tuple* $(\alpha, \beta, R, V, \gamma, \phi)$*, where:*

- α *is the first token type,*
- β *is the second token type,*
- R *represents the amounts in reserves,*
- V *is the traded volume over a period of time,*
- $1 - \gamma$ *is the percentage fee,*
- ϕ *is the protocol fee.*

By convention, R denotes the reserves in α, and V denotes the volume in α. For the rest of this paper, any quantity that denotes an asset will use the same convention.

In the following allocation games, we will consider two AMMs which are in the most direct kind of competition, or *Competing AMM Forks*.

Definition 2 (Competing AMM Forks). *Two AMMs* $(\alpha, \beta, R_a, V_a, \gamma, \phi_a)$ *and* $(\alpha, \beta, R_a, V_a, \gamma, \phi_a)$ *are competing when:*

- *they share the same token types* α *and* β*,*
- *they share the same fee level* γ*,*
- *they are operated by the same smart contract code on the same blockchain, thus resulting in identical blockchain transaction costs for performing a trade.*

In the rest of this paper, when considering Competing AMM Forks, $\mathcal{A}_x$ will systematically denote $(\alpha, \beta, R_x, V_x, \gamma, \phi_x)$.

For the purposes of analyzing equilibria under competition, we will want to observe two competing AMM forks where one is a clear market leader with the following condition.

Definition 3 (Market Leader Condition). *Two Competing AMM Forks* $\mathcal{A}_a$ *and* $\mathcal{A}_b$ *verify the Market Leader Condition when* $V_a > V_b$ *and* $R_a > R_b$*.*

This will also enable our analysis to focus on cases where the ratio of reserves always starts strictly higher than 50% in favor of the larger AMM.

2.2 Trader volume allocation

We now model traders and LPs so as to reason about how they will behave when presented with the option to allocate their trades or reserves to any of two competing AMM forks.

We assume traders to be profit maximizing. Their net value received from a trade on a given AMM is modeled as $\mathsf{TradeValue} - \mathsf{Fees} - \mathsf{PriceImpact} - \mathsf{TxCost}$, with:

- TradeValue: the payoff from trading, specific to the trader's preferences and independent of the AMM,
- Fees: the AMM trading fees,
- PriceImpact: the cost resulting from the price impact of the trade, specific to the AMM reserves and to the size of the trade,
- TxCost: the blockchain transaction costs, specific to the AMM, but equal when comparing Competing AMM Forks.

The PriceImpact term can be derived from [2] equation (7) where we consider the price impact to be equal to the price gap compared to a perfect market with infinite reserves. For a given AMM $\mathcal{A}_a$:

$$\mathsf{PriceImpact} = m_u \gamma^{-1} \left(\frac{(\delta\Delta)^2}{R_a} + O\left(\frac{(\delta\Delta)^2}{R_a^2} \right) \right)$$

with m_u the AMM price of coin α and Δ the amount of coin α traded.

The TxCost term plays an important role in that it contributes an incentive for traders to use a single AMM for their trades rather than spread them freely over multiple AMMs. This incentive will impact the equilibrium condition and result in network effects appearing.

This term assumes the traders will pay the AMM tx costs for each interaction independently. Other options would exist like using Dex aggregators which would produce a different cost profile, but it is reasonable to assume any such cost profile would always result in an incentive to use a single smart contract interaction rather than multiple ones.

Definition 4 (Trader Allocation Game)**.** *A trader aims to exchange tokens α for tokens β, with the choice to allocate trading volume Δ among two competing AMM forks $\mathcal{A}_a$ and $\mathcal{A}_b$. Let $\delta \in [0,1]$ represent the proportion allocated to $\mathcal{A}_a$, with the remaining fraction $(1-\delta)$ allocated to $\mathcal{A}_b$.*

The trader's utility is defined as

$$
\begin{aligned}
U_t(\Delta, \delta) = \mathit{TradeValue}(\Delta) \\
-\mathit{Fees}(\Delta) \\
-m_u\gamma^{-1}\left(\frac{(\delta\Delta)^2}{R_a} + \frac{((1-\delta)\Delta)^2}{R_b} + O\left(\frac{(\delta\Delta)^2}{R_a^2}\right) + O\left(\frac{((1-\delta)\Delta)^2}{R_b^2}\right)\right) \\
-\begin{cases} c_0 & \text{if } \delta \in \{0,1\} \\ 2c_0 & \text{otherwise} \end{cases}
\end{aligned}
$$

where c_0 the transaction cost for performing a swap on either of the two AMMs, considered constant, and m_u the ratio between reserves in token β and token α, considered equal across AMMs (no-arbitrage condition).

Note that the blockchain transaction costs are doubled if the allocation is split.

As we are only looking for equilibria in an allocation game, let's use a simplified utility keeping only terms that vary with the allocation proportions and simplifying out fixed multiplicative terms. Let's also consider only small enough trades so that quadratic terms are negligible.

Definition 5 (Small Trader Allocation Simplified Utility)**.** *Assuming two competing AMM forks, we define the trader's simplified utility, assuming Δ fixed, as*

$$
u_t(\delta) = -\frac{\delta^2}{R_a} - \frac{(1-\delta)^2}{R_b} - \begin{cases} c & \text{if } \delta \in \{0,1\} \\ 2c & \text{otherwise} \end{cases} \tag{1}
$$

with $c = c_0\gamma m_u^{-1}\Delta^{-2}$.

This is a concave function on $(0, 1)$ discontinuity at $\delta = 0$ and $\delta = 1$. We can easily see that the maximum on $(0, 1)$ will be the proportions of reserves.

Lemma 1.

$$
\underset{(0,1)}{\arg\max}\; u_t = \frac{R_a}{R_a + R_b}
$$

Thus we observe that under the Market Leader Condition, some traders will prefer to allocate all their trade volume to the leading AMM rather than seek the maximum described above, as long as the blockchain transaction costs are large enough to them. This will be further studied below in our equilibria analysis.

Let's denote σ, the proportion of such traders. We will assume that this proportion is constant throughout Trader Allocation Games under the Market Leader Condition in this paper. This reasonable assumption will be enough to enable network effects.

Definition 6 (Sensitive Traders Proportion)**.** *In any Trader Allocation Game under the Market Leader Condition, σ in $[0, 1]$ is defined so that:*

- σ *is the proportion of traders for whom* $\underset{[0,1]}{\arg\max}\ u_t = 1$*,*
- $(1-\sigma)$ *is the proportion of traders for whom* $\underset{[0,1]}{\arg\max}\ u_t = \frac{R_a}{R_a+R_b}$*.*

2.3 LP reserves allocation

Definition 7 (LP Allocation Game)**.** *A Liquidity Provider (LP) aims to allocate reserves (assumed only in token* α *for simplicity), with the option to allocate a total of* r *among two competing AMM forks* $\mathcal{A}_a$ *and* $\mathcal{A}_b$*. Let* $r_a \in [0, r]$ *represent the amount allocated to* $\mathcal{A}_a$*, with the remaining* $r_b = r - r_a$ *allocated to* $\mathcal{A}_b$*.*

The LP's utility is given by

$$U_l(r, r_a) = (1-\gamma-\phi_a)V_a\frac{r_a}{R_a+r_a} + (1-\gamma-\phi_b)V_b\frac{r-r_a}{R_b+r-r_a}$$

U_l is based on the definition of LP returns and protocol fee in [1].

This results in the Taylor expansion:

$$U_l(r, r_a) = (1-\gamma-\phi_a)V_a\frac{r_a}{R_a} + (1-\gamma-\phi_b)V_b\frac{r-r_a}{R_b} + O\left(\frac{r_a^2}{R_a^2}\right) + O\left(\frac{(r-r_a)^2}{R_b^2}\right)$$

Let's thus approximate the utility for small LPs, who have reserves negligible compared to the total reserves of any of the two AMMs.

Definition 8 (Small LP Utility)**.**

$$u_l(r, r_a) = (1-\gamma-\phi_a)V_a\frac{r_a}{R_a} + (1-\gamma-\phi_b)V_b\frac{r-r_a}{R_b} \tag{2}$$

Leveraging u_t and u_l will allow us to derive equilibria for both allocation games.

2.4 Aggregate allocation

We want to further study the evolution through time of competing AMM forks $\mathcal{A}_a$ and $\mathcal{A}_b$. For that, we want to observe the evolution of V_a/V and R_a/R.

This will enable analyzing which parameters influence equilibria and which conditions produce (resp. prevent) a self-reinforcing network effect.

To that end, we define two sequences that capture the dynamic of repeated games through blocks and model the occurrence of trades and reserves allocation as a series of sequential allocation games within a block

We further want to assume that the aggregate volume and reserve allocated by traders at any time step are constant, so as to focus our inquiry on the dynamics between the two AMMs.

Definition 9 (Block Allocation Game)**.** *Blocks are denoted by their indexes $i \in \mathbb{N}$.*

We consider $\mathcal{A}_a$ and $\mathcal{A}_b$ which verify:

- *the total allocated trade volume per block, $V = (V_a)_i + (V_b)_i$ is constant for all $i \geq 0$,*
- *the total amount of reserves $R = (R_a)_i + (R_b)_i$ is constant, for all $i \geq 0$,*

with AMM notation expanded to include $((V_.)_i)_{i \geq 0}$ the trade volume per block and $((R_.)_i)_{i \geq 0}$ the reserves amount per block.

A Block Allocation Game is constructed by sub-games for each blcok $i \geq 0$ in the following order:

- *Block Traders Allocation Game: repeated sequential Trader Allocation Games so that the sum of trades allocated is equal to the volume per block V,*
- *Block LPs Allocation Game: repeated sequential LP Allocation Game so that the sum of reserves allocated is equal to R.*

From here onward, any mention of AMM forks will refer to this definition.

Note that the entirety of reserves, R, is allocated anew by LPs at each block. This simplifying measure will enable us focusing on how equilibria emerge, even if sacrificing modeling accuracy.

Taking into account multiple forms of transaction costs for LPs would result in only a fraction of R being reallocated at each round, which makes the dynamic system move slower, but arguably still in the same direction.

More generally, picking a different definition for Block Allocation Games might yield slightly different results down the line, thus might be worth exploring to refine the model.

To further simplify, we also assume that all traders playing the Block Traders Allocation Games make small-enough trades. Hence, for the purpose of analyzing equilibria, we will define their utility as the Small Trader Allocation Simplified Utility u_t. Equivalently, we assume that all LPs entering any Block LPs Allocation Game have reserves negligible compared to the total reserves so we can define their utility as the Small LP Utility u_l.

Note that these simplifications will only matter as a way to calculate equilibria, thus need only be good local approximations. For example, if we assume that in an average Ethereum block a large proportion of LP reserves allocation are of the small kind, these will be robust.

Nevertheless, we might be losing modeling accuracy with respect to blocks where a single large LP is making a large move. Such events could produce sudden variations, which could be modeled as stochastic terms. It is outside of the scope of this simple model and could be suggested as a refinement.

Next, we define our dynamic system based on allocation ratios rather than volume and reserves amounts, leveraging the Block-Competing AMM Forks constraints on V and R being constant thus unnecessary to include in our model.

Definition 10 (Allocation Ratios)**.** *Given two Block-Competing AMM forks $\mathcal{A}_a$ and $\mathcal{A}_b$, along with an initial block indexed* 0 *(by convention),* $(T_i)_{i\geq 0}$ *and* $(L_i)_{i\geq 0}$ *are sequences with values in* $[0,1]$ *defined by the recurrence relations:*

- $T_0 = \frac{(V_a)_0}{(V_a)_0+(V_b)_0}$
- $L_0 = \frac{(R_a)_0}{(R_a)_0+(R_b)_0}$
- $T_i = \mathit{BlockTradersAllocation}(T_{i-1}, L_{i-1})$
- $L_i = \mathit{BlockLPsAllocation}(T_i, L_{i-1})$

where

- *BlockTradersAllocation represents the outcome of the Block Traders Allocation Game as a function of previous block state*
- *BlockLPsAllocation represents the outcome of the Block LPs Allocation Game as a function of the outcome of Block Traders Allocation Game and of previous block state.*

(T_i) represents the aggregate swap volume allocation ratio to $\mathcal{A}_a$. (L_i) represents the aggregate the aggregate reserves allocation ratio to $\mathcal{A}_a$.

3 Equilibria analysis

3.1 No fee scenario: network effects

If two AMM forks follow the Market Leader Condition, we want to observe how network effects apply and make the leader a monopoly. For that, we analyze the dynamic system described by (L_i) and (T_i).

We will assume that $(L_i) > 0.5$ and $(T_i) > 0.5$ throughout this section unless otherwise specified.

Proposition 1 (Block Traders Allocation Rule)**.** *In a Block Allocation Game under the Market Leader Condition,*

$$\forall i > 0, T_i = \sigma + (1-\sigma)L_{i-1}$$

Proof. Follows from the Sensitive Traders Proportion definition applied to every Trader Allocation Game in any Block Traders Allocation Games. □

Proposition 2. *In a Block Allocation Game under the Market Leader Condition with $\phi_a = \phi_b = 0$*

$$\forall i > 0, L_i = T_i$$

Proof. Follows from observing that the equilibrium condition for any LP Allocation Game in a Block LPs Allocation Game is $\frac{\partial u_l}{\partial r_a} = 0$ which is equivalent to $\frac{R_a}{V_a} = \frac{R_b}{V_b}$ hence $\frac{R_a}{R} = \frac{V_a}{V}$. □

Proposition 3. *In a Block Allocation Game under the Market Leader Condition with $\phi_a = \phi_b = 0$*

$$\lim_{i\to\infty} L_i = 1$$

Proof. $L_i = \sigma + (1-\sigma)L_{i-1}$ thus (L_i) is strictly increasing. And it is bounded by 1. □

3.2 Leader with fee scenario: equilibrium condition on fee

If the market leading AMM enables a protocol fee $\phi_a > 0$ while the fork doesn't, we expect that some players will allocate less to the leading AMM. We want to look into when it produces a non-monopoly equilibrium.

In this sub-section, we assume $\mathcal{A}_a$'s protocol fee is permanently set to ϕ_a.

Proposition 4. *In a Block Allocation Game under the Market Leader Condition with $\phi_a > 0$ and $\phi_b = 0$*

$$\forall i > 0, L_i = \frac{1-\gamma-\phi_a}{1-\gamma-\phi_a T_i} T_i$$

Proof. Follows from deriving the maximum u_l as a function of r_a, shown in equation (2), with $\phi_b = 0$. □

We can observe that, as $T_i \in [0,1]$, for $i > 0$ we have $L_i < T_i$. This corroborates the idea that introducing the fee in $\mathcal{A}_a$ reduces its attractiveness for LPs and thus hinders the network effects.

Proposition 5 (Leader Fee Equilibrium)**.** *In a Block Allocation Game under the Market Leader Condition with $\phi_a > 0$ and $\phi_b = 0$, and assuming that T_0 is not null, the dynamic system will be at equilibrium either*

- *when $T_i = 1$,*
- *when $\phi_a = \sigma(1-\gamma)T_i^{-1}$.*

Proof. $T_0 \neq 0$ induces $T_i \neq 0$ for all i. Then the equation follows from solving the equilibrium equation constructed by combining the Leader Fee Block LPs Allocation Rule with the Block Traders Allocation Rule. □

Note that whenever $T_{i_0} = 1$, it results that $T_i = S_i = 1$ for any $i > i_0$ and the system won't move. This notably means that introducing any fee at this point will not make the system move at all.

Typical values are $\gamma = 0.003$ and $\phi = 0.0006$ as per [1]. We assume $\sigma = 0.2$ to be a reasonable approximation. In these conditions, $\sigma(1-\gamma)\phi_a^{-1} > 1$ thus making the second equilibrium condition unachievable. In such cases, equilibrium can only be achieved when $T_i = 1$.

To illustrate aggregate players behavior depending on how the actual protocol fee is set, let's suppose it is set so that $\phi_a < \sigma(1-\gamma)T_i^{-1}$. This produces additional interest from LPs to allocate to $\mathcal{A}_a$, thus making both reserves and volume allocated to $\mathcal{A}_a$ increase through time until equilibrium is reached.

Also, whenever the fee is high enough, trade volume and reserves allocation to $\mathcal{A}_a$ will eventually become lower than half of the total, canceling the Market Leader Condition. It is easy to observe that in that case, a mirror situation where $\mathcal{A}_b$ becomes market leader in the Block Traders Allocation Rule thus producing reserved network effects in favor of it. This will help define an upper bound on ϕ_a for this not to happen.

Formally:

Theorem 1 (Asymptotic Allocation)**.** *Under the same premises as the Leader Fee Equilibrium proposition,*

- $2\sigma(1-\gamma) < \phi_a$ *implies* $(T_i)_{i\geq 0}$ *and* $(L_i)_{i\geq 0}$ *are decreasing and* $\lim_{i\to\infty} T_i = 0$.
- $\sigma(1-\gamma)T_0^{-1} < \phi_a < 2\sigma(1-\gamma)$ *implies* $(T_i)_{i\geq 0}$ *and* $(L_i)_{i\geq 0}$ *are decreasing and* $\lim_{i\to\infty} T_i = \sigma(1-\gamma)\phi_a^{-1}$
- $\phi_a < \sigma(1-\gamma)T_0^{-1}$ *implies* $(T_i)_{i\geq 0}$ *and* $(L_i)_{i\geq 0}$ *are increasing and* $\lim_{i\to\infty} T_i = \min(1, \sigma(1-\gamma)\phi_a^{-1})$

Proof. First point follows observing that by monotonicity $\exists k > 0$ where the Market Leader Condition becomes reversed and updating the Traders Allocation Rule for $i \geq k$ with $T_i = (1-\sigma)L_{i-1}$. Second and third point follow from assuming Market Leader Condition doesn't change, applying both Allocation Rules and noting $T_i \in [0, 1]$. □

4 Stackelberg attack

Having established the basic equilibrium condition for such AMM forks where the leader imposes a protocol fee, let's now include the governance of the AMM (Governance) as a player who can change the AMM protocol fee at any block.

We assume for the sake of the argument that Governance is profit maximizing and that the protocol fee is entirely directed to itself as a payoff. Note that Governance would usually be comprised of governance token holders.

As a new starting point, we consider only one AMM without forks yet.

Intuitively, it appears that, as long as LPs' preferences are entirely represented by their LP Allocation Game utility (notably, if they don't own governance tokens), then they would be ready to fork the AMM and start using the feeless fork, any time when it becomes more profitable to do so.

We assume that LPs can commit to new strategies, including deploying smart contracts that may allocate reserves on their behalf. In line with [7], [9], we observe Stackelberg equilibria emerging from smart contract moves.

We will derive the Grim Forker, a simple smart contract that allows LPs to commit to fork the AMM and lock their reserves there, whenever the fee is higher than a defined threshold. This aims at forcing Governance to not move fees higher than the threshold.

Let's further assume for simplicity that deployment and interaction of such smart contracts incurs negligible costs to LPs.

4.1 Governance player

Definition 11 (Governance Fee Setting Game)**.** *Using notation from the Block Allocation Game, a Governed AMM $\mathcal{A}_{gov}$'s protocol fee is denoted by a sequence evolving through time $(\phi_{gov})_{i\geq 0}$.*

The Governance player's action space consists of updating $(\phi_{gov})_i$ at any block i with a value in $[0, 1-\gamma)$.

Governance's payoff at each block i is given by

$$(u_g)_i = (\phi_{gov})_i (V_{gov})_i$$

And its utility is represented by the expected value of payoffs

$$U_g = \sum_{i=i_0}^{\infty} (\phi_{gov})_i (V_{gov})_i \delta^{-i}$$

with δ a discount factor in $(0, 1)$.

4.2 Grim Forker contract

Let's describe Grim Forker, a smart contract which constitutes a Stackelberg attack in the sense of [9]: it enables LPs to commit to a specific strategy based on some parameters, with the goal of influencing Governance to reduce the fee as much as possible.

The proposed contract consist of:

- a vault (in line with [10]) which acts as a LP itself on $\mathcal{A}_{gov}$ and which funds are kept available for withdrawal before the fork, but are locked after the fork happens,

- a clause to fork the AMM when some key conditions are met, notably depending on a maximum fee value $\phi_{threshold}$ which is a parameter of the contract.

Note that the locking of the funds after the fork aims at producing a large enough penalty on Governance expected earnings. The duration of this lock can be modulated as a parameter as well.

Let's describe the forking pseudocode, which is going to be run at each block:

Algorithm 1 Forking Routine

```
if φ_gov > φ_threshold and R_GrimForker > R_gov/2 then
    Deploy fork of AMM
    Withdraw reserves from AMM
    Allocate reserves to fork
    Prevent withdrawal (for some period)
end if
```

with $R_{GrimForker}$ the amount of reserves that have been allocated to this contract.

We can define a new, adapted, allocation game to account for the fork.

Definition 12 (Fork Block Allocation Game)**.** *Given an AMM $\mathcal{A}_{gov}$ and a its newly forked $\mathcal{A}_{fork}$, a Fork Block Allocation Game is constructed by sub-games in the following order:*

- *Grim Forker LPs Auto-Allocation: execution of the forking routine, allocating all $R_{GrimForker}$ to $\mathcal{A}_{fork}$,*
- *Block Traders Allocation Game,*
- *Block LPs Allocation Game.*

4.3 Stackelberg equilibrium

We can now describe the metagame that happens between LPs and Governance where LPs add the deployment of specific Grim Forker instances to their action space.

Definition 13 (Grim Forker Game)**.** *Players are defined by*

- *Governance, whose utility is U_g,*
- *LPs who participate in Grim Forker, or GFLPs.*

The game happens in the following steps:

- *GFLPs deploy a version of Grim Forker with some parameters,*
- *Governance adjusts ϕ_{gov} accordingly.*

Assuming that more that half of reserves are allocated to Grim Forker, as Governance aims at maximizing its utility U_g, it is presented with two choices:

- prevent Grim Forker from launching the fork while maximizing U_g, thus setting $\phi_{gov} = \phi_{threshold}$ and gaining $U_g = \sum_{i=i_{fork}}^{\infty} \phi_{threshold} V \delta^{-i} = \phi_{threshold} V (1-\delta)^{-1}$,
- let Grim Forker launch the fork if the clause is met and maximize the yield of $U_g = \sum_{i=i_{fork}}^{\infty} (\phi_{gov})_i (V_{gov})_i \delta^{-i}$ by playing on ϕ_{gov}.

The second option requires more complex modeling but we know that $(V_{gov})_i$ tends towards 0 in this case thanks to Theorem-1. It might still be the rational option if the discount factor is chosen particularly low as part of Governance's preferences.

Low δ could reflect a large proportion of short-term-minded governance participants, who would thus try to influence Governance in making the second choice.

We can now derive the conditions for participation of LPs.

Theorem 2 (Grim Forker Participation)**.** *It is ex-interim individual rational for LPs to allocate their reserves to Grim Forker if*

- *they don't have conflicting interests with Governance: their entire preferences are represented by u_l,*
- *they assume that Governance will prefer preventing the fork.*

Proof. First, the equivalent amount of reserves allocated in Grim Forker or directly in the AMM yields the same returns. Hence, leveraging the assumption that transaction costs are negligible, it is indifferent for LPs to allocate to the AMM directly or to Grim Forker as long as the forking is not triggered.

Second, LPs would prefer allocating their reserves through Grim Forker to keep the protocol fee below $\phi_{threshold}$. □

Further refinements to the model could include refining preferences for LPs who are also part of Governance, adding more costs and risks (e.g., development, legal) and accounting for delays.

4.4 Market-sourced equilibrium condition

The Grim Forker contract depends on $\phi_{threshold}$, the $R_{GrimForker}$ threshold, and the duration of locking funds as parameters.

Further, more involved modeling of these parameters would result in higher participation from LPs. Simple competition among different Grim Forker contracts could be leveraged to let emerge what LPs agree on being the best estimation algorithm. A downside of this approach is liquidity fragmentation among multiple Grim Forker contracts, thus reducing efficiency.

Another approach would be to embed this market sourcing objective within the Grim Forker smart contract, so as to make, for example, the algorithm for $\phi_{threshold}$ evolve through time. This would solve the liquidity fragmentation issue but at the price of higher smart contract complexity.

5 Conclusion

We have produced a dynamic, block-per-block, model for traders volume allocation and LPs reserves allocation among two AMM forks. We have established asymptotic closed-form values for reserves and trade volume allocation ratios. Assuming reasonable real-world constraints on protocol fee rate, this result points towards an indisputable natural monopoly of the market-leading AMM among its forks, thus removing any benefit of fork-based competition for LPs who are thus at risk of losing part or all of their revenue to Governance.

We introduced a Grim Forker contract as a Stackelberg attack of LPs on Governance, and shown that this creates a new equilibrium, basically giving bargaining power to LPs. Depending on the participation rate in the contract, this attack can force Governance into reducing protocol fees, otherwise risking to see the original AMM be drained of all of its reserves.

In turn, Grim Forker contracts, by leaving their participation rate openly auditable, instruct Governance on how to operate their protocol fee management in an optimal way. Introducing obfuscation techniques as in [4, 3] would hinder auditability and change the game dynamics, requiring a different modeling treatment.

Acknowledgments. We thank Vaughn McKenzie-Landell and reviewers for their feedback and suggestions.

References

[1] H. Adams, N. Zinsmeister, M. Salem, R. Keefer, and D. Robinson. Uniswap v3 core, 2021. URL https://uniswap.org/whitepaper-v3.pdf.

[2] G. Angeris, H.-T. Kao, R. Chiang, C. Noyes, and T. Chitra. An analysis of uniswap markets, 2021.

[3] J. Austgen, A. Fábrega, S. Allen, K. Babel, M. Kelkar, and A. Juels. Dao decentralization: Voting-bloc entropy, bribery, and dark daos, 2023.

[4] P. Daian, T. Kell, I. Miers, and A. Juels. On-chain vote buying and the rise of dark daos, 2018. URL https://hackingdistributed.com/2018/07/02/on-chain-vote-buying/.

[5] M. G. Delaunay, P. Frambot, Q. Garchery, and M. Lesbre. Morpho blue whitepaper, 2023. URL https://github.com/morpho-org/morpho-blue/blob/main/morpho-blue-whitepaper.pdf.

[6] GFXlabs. Making protocol fees operational, 2023. URL https://gov.uniswap.org/t/making-protocol-fees-operational/21198.

[7] M. Hall-Andersen and N. I. Schwartzbach. Game theory on the blockchain: A model for games with smart contracts. In I. Caragiannis and K. A. Hansen, editors, *Algorithmic Game Theory*, pages 156–170, Cham, 2021. Springer International Publishing. ISBN 978-3-030-85947-3.

[8] L. Huo, A. Klages-Mundt, A. Minca, F. C. Münter, and M. R. Wind. *Decentralized Governance of Stablecoins with Closed Form Valuation*, page 59–73. Springer International Publishing, 2023. ISBN 9783031186790. doi: 10.1007/978-3-031-18679-0_4. URL http://dx.doi.org/10.1007/978-3-031-18679-0_4.

[9] D. Landis and N. I. Schwartzbach. Stackelberg attacks on auctions and blockchain transaction fee mechanisms, 2023.

[10] J. Santoro, t11s, J. Jadeja, A. C. Cañada, and S. Doggo. Erc-4626: Tokenized vaults, 2021. URL https://eips.ethereum.org/EIPS/eip-4626.