Evmos

Token Economics Redesign

Key aspects of token issuance and staking rewards distribution to enhance sustainability in the ecosystem.

POL

pol.finance

Evmos Token Economics

POL Finance[1],
Federico Kunze Küllmer[2], and
Daniel Burckhardt[3]

[1] *dipa@pol.finance, agustin@pol.finance*
[2] *federico@evmos.org*
[3] *daniel@evmos.org*

June 13, 2024

Abstract

This report presents a proposal for redesigning the economy of the native tokens on the Evmos chain. The initiative aims to address key aspects of token issuance and staking reward distribution to enhance sustainability, decentralization, and fairness in the ecosystem. It is suggested to adjust the current annual emission rates to a daily emission that adjusts dynamically to changing market conditions. Furthermore, changes are proposed in the staking reward structure for validators, considering implementing a quadratic distribution system and a duty-based scoring system through a base factor to incentivize key activities that validators perform. This approach seeks to optimize validator participation, promote decentralization, and ensure the healthy growth of the network. Finally, the implementation of a burning process is crucial to ensure a controlled future emission rate, fostering a long-term deflationary expectation that supports an increase in the valuation of the EVMOS token. This also aligns the EVMOS delegated Proof-of-Stake (dPoS) with the standard PoS mechanism used in Ethereum. The proposals are supported by mathematical analysis, simulations, and technical considerations to provide a comprehensive view of the potential impact on the Evmos ecosystem.

Disclaimer

The following disclaimer is intended to clarify the purpose and limitations of the report produced by POL Finance and the Evmos Core Team for the study of the Evmos protocol. Please read this disclaimer carefully before reviewing the report.

No Financial Advice: The information provided in this report is for informational purposes only and should not be construed as financial advice, investment recommendations, or an invitation to engage in any buying, selling, or holding of any specific asset. POL Finance, Evmos Foundation and the authors do not provide financial advice, and the content of this report should not be considered as a sole or definitive source for making decisions or relying upon.

Limitation of Liability: POL Finance, Evmos Foundation and the authors assume no responsibility or liability for any financial losses, direct or indirect, incurred from using or interpreting the information provided in this report. Readers are responsible for their own investment decisions and should undertake their due diligence.

Information Obsolescence: It is crucial to acknowledge that the information provided in this report may become obsolete due to dynamic market conditions, regulatory shifts, technological advancements, or other factors. Readers should be aware that the report's content is subject to change, and it is advisable to conduct timely research and analysis before making any investment decisions.

Contents

Chapter 1

Introduction

A robust economic system is of paramount importance for the proper functioning of a blockchain. A blockchain operates as an incentive machine, motivating agents to perform specific actions by rewarding them with tokens. As constraints are introduced, this system becomes increasingly intricate. Evmos recognizes the significance of robust token engineering, which is why we collaborate to achieve it.

In this specification, we propose a redesign of the original token economics that were introduced with the launch of the Evmos chain. We first outline the problems and historical events associated with the current state of the Evmos token economics. These include an overestimated token emission schedule, ineffective token distribution, and liquidity issues. For more information, please refer to this blog post.

Next, we discuss the potential of implementing changes on the core protocol level to improve decentralization through validator fairness and incentivize demand for applications by normalizing staking rewards.

Chapter 2

Evmos Tokenomics v1

In this chapter, we will describe the problems with the current design of Evmos tokenomics.

During its first year, Evmos attracted a significant user base, mostly due to its token economics and highly inflationary token emissions. However, after Terra's collapse, the Nomad hack, and other major events in the ecosystem (such as TAC, FTX fraud, SEC court filings, and the macroeconomic recession environment), a liquidity crunch occurred not only in the decentralized market but in the market as a whole. This resulted in users continuously disposing of their EVMOS tokens rather than depositing them in DeFi apps to build up Total Value Locked (TVL).

There are different issues related to the initial design of the Evmos Token Economics that come into play.

2.1 Emission Problem

EVMOS tokens are minted daily according to the emission schedule (aka. release schedule) that follows the **Half Life** function.

$$f(t) = a(1-r)^t \times \left[1 + v\left(1 - \frac{b}{b^*}\right)\right] + c$$

where

	Name	Description	Initial value
`t`	time	time period (year)	0
`a`	initial value	Initial token supply	300,000,000
`r`	decay factor	Emissions change over period	0.5
`v`	max variance	Max amount to increase inflation	0.0
`b`	staking ratio	The ratio of staked vs total supply of tokens	N/A
`b*`	target staking ratio	Optimal Staking Ratio	0.66
`c`	constant	Minimum long term supply	9,375,000

This emission function takes into account the ratio of staked tokens to the total token supply, represented by the letter b, and the Target Staking Ratio b^*.

- The number of tokens emitted was subject to a constant emission rate, which, unfortunately, proved to be inflexible in adapting dynamically and promptly to adverse market conditions. This resulted in the emission rate lagging behind and becoming misaligned with the average global inflation, both within and outside the Cosmos ecosystem, during the aforementioned negative market events.

- High emission created a liquidity crunch in the market that incentivized users to dispose of their EVMOS tokens instead of utilizing them in DeFi apps or locking them in said applications to build up TVL. As it became much more attractive to stake EVMOS tokens for a high return rather than using them as a payment method for consuming services on the network, the high issuance led to a widespread disinterest in the use of EVMOS dApps.

Remark. At the time of writing, the emission (inflation) rate on Evmos is around 36% whereas top Cosmos chains APR's in comparison are: Osmosis 12%, Cosmos Hub 10%, Crypto.com 9%, Injective 16%, Stargaze 25%; Other Chains: Polygon 1%.

Recommendation: We recommend developing an emission function that dynamically adjusts based on the most relevant market's variables for network health, such as the staked ratio and network demand or traffic. Simultaneously, we propose that emissions and adjustments occur daily to allow for greater flexibility than the current annual emission scheme provides. Additionally, by incorporating a daily hard cap, we can avoid the cannibalization effect on dApps, provided that this hard cap is calculated based on an ecosystem-representative emission rate. Finally, a daily design reduces the computation required to mint and distribute tokens from a per-block to a daily basis. Details in 3.1.

2.2 Distribution Problems

Once the tokens are minted, they are allocated to different categories of pools. Each pool has its own internal logic for how to distribute the tokens it receives on each epoch.

please, decide if use capital first letter in ordered list and if using period at the end of each item. check There are two topics to consider when it comes to the token distribution:

1) The distribution percentages of the tokens to the different pools, i.e where tokens go after minting.

2) The internal distribution and allocation of tokens that are liquid, i.e. the tokens that are immediately available to end users.

The community treasury (aka Community Pool) currently holds more than 115M EVMOS tokens at the time of writing. It can be spent through governance proposals such as to fund community-built projects.

The Incentives pool account also holds more than 115M EVMOS tokens at the time of writing. However, the usage incentives have never been adopted. Reasons for this might be

1) The lack of education around how to activate incentives.

2) Lack of need from the user wanting to get back their transaction fees, as transaction fees on Evmos are very cheap anyways.

In summary, we need to solve both problems: the user pool's funds are not used at all, and the Community pool has already taken enough funds.

Recommendation: Regarding the inefficiency of using the community pool to incentivize both users and developers, we will address this in the following sections. Considering the quantity of tokens in both pools and the need to issue tokens to pay validators under the new incentive scheme, we propose the following distribution categories.

- Community pool → 0%
- Staking → 100%
- Usage incentives → 0%

Please, decide if use ”European style title” or ”American Style Title” and use only one. check

2.3 Ineffective Incentives

2.3.1 Inefficient Validation Incentives

- As participants accumulate more voting power in the network, they are proportionally allocated more resources, creating a dynamic where individuals with more resources tend to accumulate even more, leading to imbalances and wealth concentration. This 'rich-get-richer' scheme was provoked due to the fact that allocation is pro-rata to the amount of consensus voting power they hold in the network.
- Validators must generate more income than expenses to maintain the network's viability. Being ”cash-flow positive” implies that validators are capable of producing a surplus in their financial transactions, ensuring their economic sustainability and contributing to the continuous operation of the network.

we should also define what cash-flow positive means. On average? Do we have data for this? done

Recommendation: Here, we encounter two fundamentally distinct and even contradictory issues. On one hand, to discourage the concentration of power that the current linear rewards scheme generates, we propose adopting a quadratic distribution scheme. This approach will reduce the profit rate for larger validators and increase it for smaller validators. On the other hand, to ensure the profitability of validators, we propose reducing validation costs and, if possible, utilizing tokens available in the community pool or even new inflows of tokens derived from transaction fees on other blockchains that have incorporated the Evmos module. Finally, to align the behavior of validators towards one that improves the network's health in the long term, we suggest incorporating the concept of a Base Factor. Details in 3.3.1.1 and 3.4.2.

2.3.2 Usage Incentives

Usage incentives allow users and projects to register an on-chain incentive via governance which, upon approval, tracks the amounts of gas from EOAs that interact with the contract address. The incentives would have a duration and a % of the daily EVMOS rewards allocated to the incentives pool.

Usage incentives were not used due to multiple reasons:

- DApp builders are not familiar with Cosmos Governance. There was no (and still is no) UI to create proposals in a user-friendly manner.

- Unawareness from the users about the benefits of submitting the incentives.
- No direct integration of the incentives with smart contracts.
- Mechanism design was ineffective as it
  - Was not specifically targeted to bootstrap TVL.
  - Had no lock/unlock period.
  - Had no vesting for the token reward.

Recommendation: We propose to enhance the visibility and broaden the reach of grants and Hackathons. A well-structured Communication and Marketing schedule is essential for this. However, these incentives can be burned to decrease the Total Supply and boost confidence in the token's value, thereby potentially attracting more users to hold the token.

2.3.3 Ineffective Incentives for Developers

Evmos has a revenue share model on which developers can earn a percentage of the transaction fees for all the transactions that interact with their registered smart contract.

There main points of failure for this model were the following:

- The fees distributed to developers from the revenue module are insufficient to cover operational expenses (OPEX).
- Developers did not have a way to integrate the revenue registration process in the smart contract constructor function.
- Developers have no monitoring tool to track how much fees have been earned in a given period.

Recommendation: Part of the insufficiency of fees to cover developer costs is attributed to the insufficient demand for the EVMOS token. Therefore, increased demand for Evmos and its applications resulting from the impact of the changes suggested by this tokenomics report will enhance developer revenue. Furthermore, we propose resolving or improving the monitoring and integration technologies of registered smart contracts. Details in 3.3.2

2.3.4 Low Demand for EVMOS Token

Demand for the token is low because DeFi on Evmos never reached considerate stability. Bridge hacks, a market recession, poor business development, and ineffective incentives set forth major roadblocks for DeFi adoption on Evmos and resulted in the lack of a successful DeFi application.

Recommendation: The redesign of the network's tokenomics to ensure that the aforementioned issues and objectives are achieved in the medium to long term should instill more confidence in the utility and value of the token. Moreover, the drastic reduction in emissions and the change in the distribution of newly issued tokens will exert stricter control over supply growth, thereby alleviating downward pressure on the token price and enhancing the final token valuation of the community and the ecosystem.

2.4 Lack of a Clear Burning Mechanism

On Evmos, fees are currently not burned, which means that the token is fully inflationary.

Recommendation: We propose introducing the burning of fees on Evmos to reduce the circulating supply of Evmos tokens such as: Base Fee from EVMOS Transfers (EOA → EOA), Base Fee from unregistered contracts, Outposts, all the transaction fees from Cosmos transactions and the tokens in the Incentives Pool . Details in 3.4.1.

Chapter 3

Evmos Tokenomics v2

3.1 Solution to Emission Problem

We consider it is essential to incorporate other market variables apart from the staked ratio to work with an issuance function that has the following properties:

- **Elasticity**: The ability to adapt to the network environment and resistance to shocks in case of a loss of liquidity, massive unbonding of staked tokens, or a decrease in demand. (Where ”liquidity” refers to Total Reachable Liquidity (TRL), ”tokens staked” refers to b, and ”demand” involves the base fee, average transactions per block, and the growth in the number of deployed contracts).
- **Dynamism**: The capacity to behave in an inflationary or deflationary manner according to network conditions. The function will be defined as depending on the following
  - Staking ratio (daily): staked tokens vs total.
  - Demand of the network (daily): base fee value.
  - Weights: Used for parametrization of the staking ratio and demand.

We propose the following issuance function

$$\begin{cases} f(0) = a, \\ f(t) = (1 + R(t)) * f(t-1), (t \geq 1) \end{cases} \tag{3.1}$$

where

$$R(t) = \max\left\{0, \min\left\{r_s, \omega_s\left(1 - \frac{s(t)}{s^*}\right) + \omega_b e^{-\alpha_{\text{BaseFee}} \mathit{BaseFee}(t)}\right\}\right\}. \tag{3.2}$$

and the parameters and variables involved in the dynamic part of R are

$$\begin{aligned}
a := 436643.8356 = \mathit{last\_registered\_emission} * r_s &: \text{Initial token supply},\\
t &: \text{Time period (day)},\\
s &: \text{Staking ratio (at the end of the day)},\\
s^* = 0.66 &: \text{Target staked ratio},\\
BaseFee &: \text{BaseFee daily average},\\
\alpha_{\text{BaseFee}} = 10^{-9} &: \text{Normalize to a manageable scale},\\
r_s = 0.0086 &: \text{Upper emission bound (daily)},\\
\omega_s = 0.000233 &: \text{Staking weight},\\
\omega_b = 0.000249 &: \text{BaseFee weight.}
\end{aligned}$$

Finally, the initial supply is being determined utilizing the agent-based model.
The calculation of r_s comes from the following. We have that

$$(1 + R(t)) \leq 1 + r_s \Rightarrow \prod_{t=1}^{365} (1 + R(t)) \leq (1 + r_s)^{365}$$

Then we ask r_s to be such that

$$\frac{(1 + r_s)^{365} * a}{\mathit{initial\_total\_supply}} \leq 0.02 \Leftrightarrow r_s \leq \left(0.02 * \frac{\mathit{initial\_total\_supply}}{a}\right)^{1/365} - 1 \sim 0.0086$$

The decision to take emissions on a per-day basis is justified by the fact that the underlying mathematical and computational optimization problem to find the optimal weights (ω_s, ω_b) becomes complex and, in some cases, unsolvable or with significantly erroneous solutions.

Note that the maximum ensures that emissions can never be negative, and the minimum guarantees that we will never emit more than r_s.

In the dynamic part of $R(t)$, the first term represents the impact of staking on issuance. Given a target staking ratio b^*, if the relationship between staked tokens and the total token supply (b) is greater than b^*, it indicates that there are more staked tokens than anticipated, resulting in a reduction in issuance. Conversely, if there are fewer staked tokens than expected, this means that more tokens are in circulation, allowing for increased issuance.

The last term aims to represent the impact of demand on issuance. Since the base fee is proportional to demand, when the base fee increases, miner tips also increase, meaning miners receive higher compensation. Consequently, issuance must be reduced. In summary, an increase in token demand results in lower issuance, while a decrease in demand reduces the base fee, which, in turn, implies an increase in issuance to cover validator costs.

We can be sure that the emission rate will not exceed a daily hard cap of r_s on the overall annual emission. The parameter tuning will be guided by the constraint of not exceeding that r_s. We will show this analysis in the Section 4.

A final comment is that in Section 4 we find the optimal weights for a more general constrain in which we would like to issue at most $r_s = \frac{0.08}{365}$ but also at least $r_i = \frac{0.02}{365}$. In other words, for each simulated path, we will find the weights by asking them to satisfy

$$r_i \leq \omega_s \left(1 - \frac{b(t)}{b^*}\right) + \omega_b e^{-\alpha_{\text{BaseFee}} BaseFee(t)} \leq r_s, \quad (\forall t, 1 \leq t \leq 30). \tag{3.3}$$

what is r_i?

It would be more restrictive to define R by taking the maximum between r_i and the other term, than taking the maximum against 0. We prefer to allow the function to not issue any token, allowing it to move freely between 0 and r_s, if market conditions demand such behavior for f.

3.2 Solution to Distribution Problems

In response to the current situation, we recommend the following adjustments to the token distribution for EVMOS:

- User Incentives Burn: Since user incentives are not being utilized, we propose burning all of them. This measure aims to decrease the total supply of EVMOS tokens, thereby enhancing community confidence in its intrinsic value.
- Community Treasury Allocation: Considering that the community treasury currently holds 20% of the total supply, we suggest abstaining from distributing any newly issued EVMOS tokens to this pool at the present moment."

Based on the previous arguments, the proposed distribution categories are:

- Community pool → 0%
- Staking → 100%
- Usage incentives → 0%

Staking will be the sole remaining category, while the community pool will be filled proportionately based on transaction volume, accruing all fees generated from the EVM Extensions.

3.3 Solution to Inefficient Incentives

3.3.1 Solution to Inefficient Validation Incentives

3.3.1.1 Decentralization

One of the major problems we currently have is that there are only a few big validators that earn money and a lot of small validators that are not profitable at all. To change this situation and promote a more decentralized, equally distributed validation distribution we propose to incorporate the Quadratic Distribution payoff design.

Quadratic Distribution

Currently, all validators receive a pro-rata reward allocation based on the total deposits (amount delegated) to their validator. However, over time, this can lead to more centralization in the network as the top validators can compound their rewards easily without worrying about covering their running costs.

To prevent the rich from getting richer and introduce a fairer distribution of staking rewards amongst validators, a **Quadratic Distribution (QD)** for staking rewards is proposed as part of the new Token Economics. QD improves the *one share, one vote* model preventing potential scenarios where:

- Large token holders (*aka*. “whales” or oligopolies) accrue a majority of issued tokens.
- Large-size validators that accrue a majority of delegations. Explorers commonly sort them by descending order based on total voting power or deposits.
- Large validators that can censor transactions. The block proposer selection algorithm in CometBFT is based on a round-robin sorted by voting power (aka. consensus power).
- Large token validators or token holders influence over governance, through the voting power their tokens conveys.

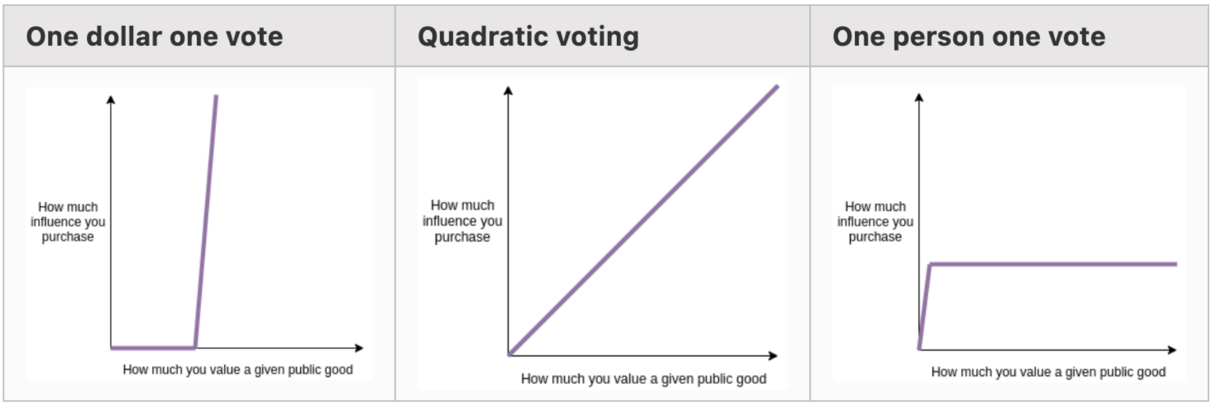


Figure 3.1: Quadratic Examples

In QD, staking rewards are based on the square root of the deposits, so the higher the number of deposits to a validator, the lower the increase of rewards for the validator.

3.3.1.2 Reward Function

Currently, Evmos utilizes a fixed total reward rate, where tokens are distributed according to the Voting Power (i.e deposit) of the validator

$$B(D_i) = \frac{D_i}{\sum_{j=1}^{n} D_j}$$

For the distribution of the staking rewards, we propose a quadratic distribution of the token.

The proposed Base Reward using Quadratic Distribution is the following:

$$QD(D_i) = \frac{D_i^{0.5}}{\sum_{j=1}^{n} D_j^{0.5}},$$

The final reward function proposal for the validators is as follows

$$R_i(t) = f(t) * \mathit{base\_factor}_i * QD(D_i) * V_c,$$

where t is the day of emission and V_c is the validator comission. The logic behind the value 0.5 will be analyzed in Section 4.

Remark. In the case where the total Base Factor percentage is lower than 100%, we propose to burn the difference. It means if $\sum_i \mathit{base\_factor}_i < 1$, that is, it does not exceed 100%, we burn the remaining tokens.

3.3.2 Solution to Ineffective Incentives for Developers

Beyond the problems mentioned in Chapter 2, this revenue split model proposed and implemented by Evmos has been a successful narrative in other EVM chains that have adopted Contract Secured Revenue (CSR) that is the same concept but implemented as a smart contract, thus providing native EVM support via NFTs and a direct integration for smart contracts that can register the integration via a constructor.

Therefore, we propose the following solutions to improve the incentives for developers

- Fixing the problem of not having a way to integrate the revenue registration process in the smart contract constructor function.
- Adding a monitoring tool to track how much fees had been earned in a given period.

3.4 Transaction Fee Market Structure

3.4.1 Burning fees

On Evmos fees are currently not burned, which means that the token is fully inflationary. Instead of relying on the economics of Ethereum for the fee market, Evmos has implemented a modified version of the EIP-1559 standard.

The key takeaways from the current implementation are (see (Spec)):

- Just as EIP-1559 on Ethereum, effective fees are calculated for all Ethereum-type transactions on Evmos
  - 'EffectiveGasPrice = min(PriorityFee + BaseFee, GasFeeCap)' (EIP-1559 for Dynamic Fee transactions).
  - 'EffectiveGasPrice = GasPrice' (EIP-2930 for AccessList transactions and Legacy transactions).
- If devs register their contract to revenue, the Effective Fee ('TotalFee = EffectiveGasPrice * Gas') is allocated to devs according to the developer share parameter (see (Code)).

  The current values are the following:

  - **Developer Revenue Share**: 95% of transaction fee.
  - Staker Fee (i.e validators and delegators): 5% of transaction fee.

- If contract is not registered, the BaseFee on Evmos is **not burned**, and instead is distributed to validators and delegators (stakers).

  This is done by transferring it to a protocol-owned account (FeeMarket) where the tokens are pooled and then distributed to the protocol-owned account (‘Distribution‘) from which the rewards are collected.

We propose introducing the **burning of fees on Evmos** to reduce the circulating supply of Evmos tokens. The following sections outline the burn and fee allocation suggestions for the different transaction categories.

Ethereum Transaction

- Priority Tip (per gas) is allocated to validators by transferring it to the ‘feeCollector‘ module account. This is then distributed to the delegators and validators.
- BaseFee (per gas) is allocated or burned according to the case. The cases for how to handle the BaseFee are 5:

Case	Action
EVMOS transfers (EOA → EOA)	Burn BaseFee
EVM Extensions (except Outpost)	Allocate BaseFee to Community
Outposts	• Allocate BaseFee*DeveloperShares to contract withdrawal address • Burn BaseFee*(1-DeveloperShares)
Registered Smart Contracts	• Allocate BaseFee*DeveloperShares to contract withdrawal address • Burn BaseFee*(1-DeveloperShares)
Unregistered Smart Contracts	Burn BaseFee

Table 3.1: Burning fees mechanism

Cosmos Transactions

Evmos supports both Cosmos and Ethereum transactions. Cosmos transactions interact directly with core protocol functionalities such as staking, governance, and token transfers, bypassing the EVM. Ethereum transactions, on the other hand, interact with dApps via smart contracts deployed on the EVM on Evmos. maybe here worth specifying that, cosmos transactions interact DIRECTLY with core WITHOUT passing through the EVM check

We plan to deprecate Cosmos Transactions (Link) in the long-term, i.e., route all transactions through the EVM. Although we already support access to some core protocol functionalities from the EVM via our Extensions, such as staking and IBC transfers, it will take some time to fully support all core protocol functionalities.

For the time being, all the transaction fees from Cosmos transactions will be burned.

Incentives Pool

As explained in detail in 2, the incentives pool was not able to capture usage incentives to increase the TVL or other measures.

As part of the burning component of the Token Economics v2, it is important to reduce the circulating supply of tokens, which includes these tokens in the calculation even though they are not effectively liquid.

Suggestion: all but a minor amount, eg. 500k EVMOS tokens (the 'leftoverTokens') from the incentives pool will be burned, the leftover tokens can be used to sponsor gasless transactions for new users that wish to onboard to Evmos.

3.4.2 Base Factor

Having a qualified group of validators running the network is essential for its security. Currently, **staking rewards for validators are based solely on their participation in consensus**. However, additional qualifications can contribute to the network's security.

To incentivize more qualified validators, a reward system based on their participation in the network can be introduced. This will add additional variables with the goal of increasing overall security in the network, providing incentives to critical infrastructure operators (eg. IBC relayers) and incentivizing validator participation.

We propose to redefine the score associated with each duty to make it proportional to the behavior of each validator, thus incentivizing the best delivery to achieve the maximum possible reward.

The base factor structure from Ethereum design has the following disadvantages

- For duties that depend on a threshold, paying a fixed amount only to those validators who are above that threshold is a disincentive.
    - The behavior of validators who are below that threshold since they have no incentive to make an effort if they are below the thresholds.
    - The behavior of validators who are above the threshold because once the threshold is reached there is no incentive to try harder since they will not earn more.
- Inefficient punishments if the distribution of tokens is not block by block.
- Downtime slashing is extreme punishment. With jail and the block-by-block token distribution redesign, the opportunity cost of being in jail is enough of a disincentive.

We propose to make the score of each duty proportional to the validator's performance above the threshold.

For Validator Uptime we propose a minimum threshold of 50%, that is, if x is the performance of the validator,

$$\text{Duty Value} = \begin{cases} 0 & if \quad x < 50\% \\ 20x & if \quad x \geq 50\%. \end{cases}$$

We reduced the threshold by 50% to reward validators who have worked above half of their capacity, incentivizing them to contribute more, as Duty Value will be proportionally representative. Proposing a lower threshold would benefit validators working less than half of their capacity, potentially increasing the number of validators with such performance, which could compromise the network's security.

In Validator Governance Participation we continue with the same logic, we propose a minimum threshold of 50%, that is, if x is the performance of the validator

$$\text{Duty Value} = \begin{cases} 0 & if \quad x < 50\% \\ \\ 10x & if \quad x \geq 50\%. \end{cases}$$

We reduced the threshold to 50% to encourage more people to vote on the initiatives.

The rationale of this new duty scoring system is that if the score is proportional to the performance, it motivates validators who want to have a higher score to make an effort to the limit of their capacity. On the other hand, scoring 0 below the threshold encourages validators who want to have a representative Base Factor to make at least a minimum effort.

The final design is

Duty	**Proposed Duty Value**	**Proposed Threshold**	**Target Threshold (i.e. 100% of Duty Value**
Validator Uptime	20	50%	95%
Validator Governance Participation	10	50%	80%
Relayer Operator	2	N/A	All or Nothing
Jailed	32	N/A	All or Nothing
Total	64		

Table 3.2: Base Factor Scores

Chapter 4

Numerical Analysis

4.1 Linear rewards vs Quadratic rewards

Impact in Profit distribution

In this section, we will examine the impact of the changes mentioned throughout the report on validator revenue distributions. Since the Validator Revenue and Validator Cost Analysis data provided by Evmos did not have the same information in terms of quality and quantity, we decided to work with a new dataset created by combining both datasets. The idea was to fill in the data for those validators that appeared in both sheets simultaneously, and for those that did not, we filled in missing data using the median of the available data. The median is a robust measure that is not affected by outliers in data with non-symmetric or skewed distributions. Since we have data with extremely high values that are not representative of the majority of cases, the median is a better choice.

Let's compare how validator rewards would appear when considering the reward function for different values of p, $p = 0.3, 0.5, 0.9$, and 1. Once again, let's remember that the case $p = 1$ represents the linear reward scheme used by Evmos today.

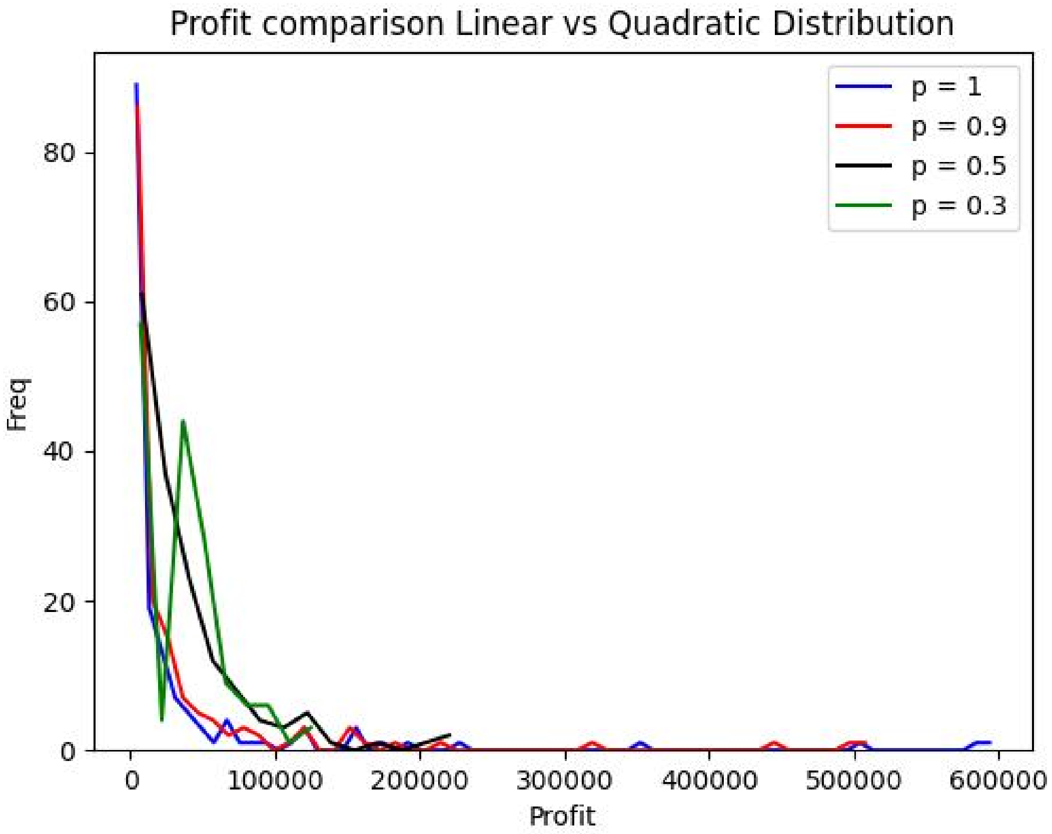


Figure 4.1: Simulated Profit

As illustrated in Figure 4.1, it's worth noting that when p is lower, the distribution of earnings becomes more decentralized, benefiting those who stake less in comparison to those who have more. Furthermore, a higher concentration of profits can be observed at the lower values of the dataset, implying a drastic reduction in the final APR for the validators.

Impact in Validator's ROI

Next, we will showcase the impact of the Return on Investment (ROI) calculated as $ROI = \frac{Profit}{Cost}$ for representative agents from three distinct groups of validators. The first will be a validator whose stake participation percentage is dominant; the second will be an average agent, i.e., whose deposits are at the mean values of the data; and finally, we will show the impact on the revenue of smaller agents. For this, we assume an annual emission of 36% of the Total Supply, as indicated by the data at the time of the study. While the absolute numbers may differ, in making relative comparisons, the analysis maintains its significance even if the emission percentage is changed.

In Figure 4.2, we observe how the shift to the quadratic distribution rewards design has a significant impact on their earnings. We can see that for $p = 1$, there are profit percentages around 4%, decreasing to 2.75% for $p = 0.9$, 1.30% for $p = 0.5$, and 0.25% for $p = 0.3$. That is, being a major player in the new design would strongly affect the final profits, potentially reducing them by up to a maximum of 90% if the lowest possible exponent $p = 0.3$ is used.

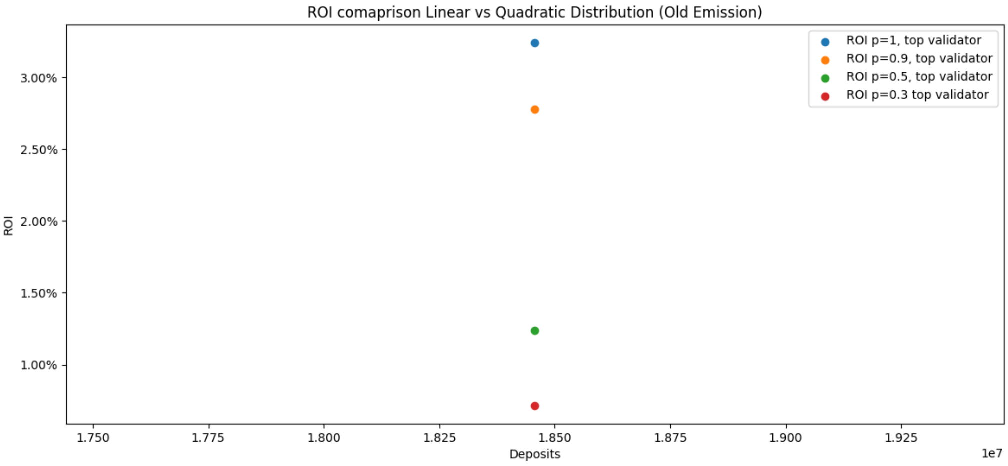


Figure 4.2: ROI top validator with old emission

On the other hand, when we look at the impact of this new design on the representative agent of an average validator, as shown in Figure 4.3, the impacts on profits mean going from an ROI of 2% for $p = 1$ to an ROI of 4.25% for $p = 0.9$, 13% for $p = 0.5$, and 19.5% for $p = 0.3$. Unlike the top validator, this agent would benefit under the new scheme, experiencing improvements of up to 975% for $p = 0.3$.

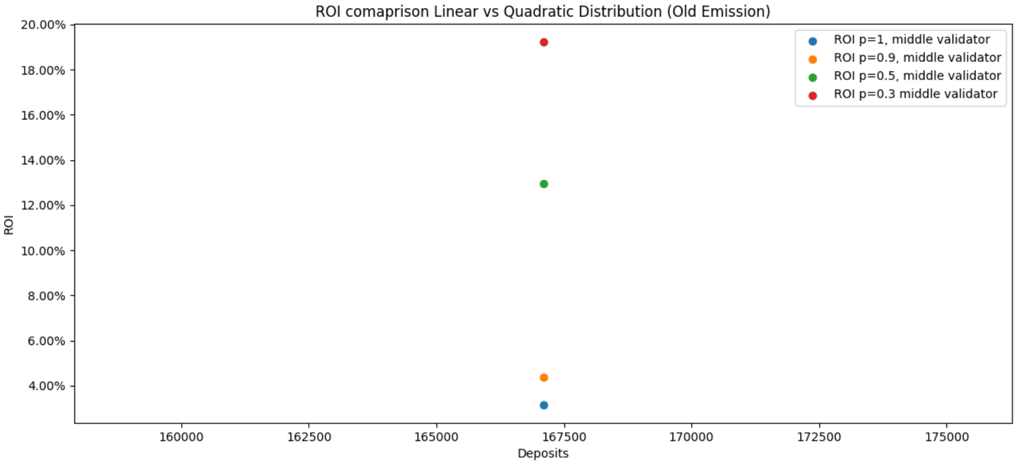


Figure 4.3: ROI medium validator with old emision

Finally, analyzing the case of a validator in the group with the least significant total deposits, as shown in Figure 4.4, we observe an ROI of 1% for $p = 1$, 4% for $p = 0.9$, 10.5% for $p = 0.5$, and 15% for $p = 0.3$. The conclusion is that being a small validator is a good idea in the quadratic approach, in relation to the linear approach, with the potential for an ROI increase of up to 1500% if $p = 0.3$.

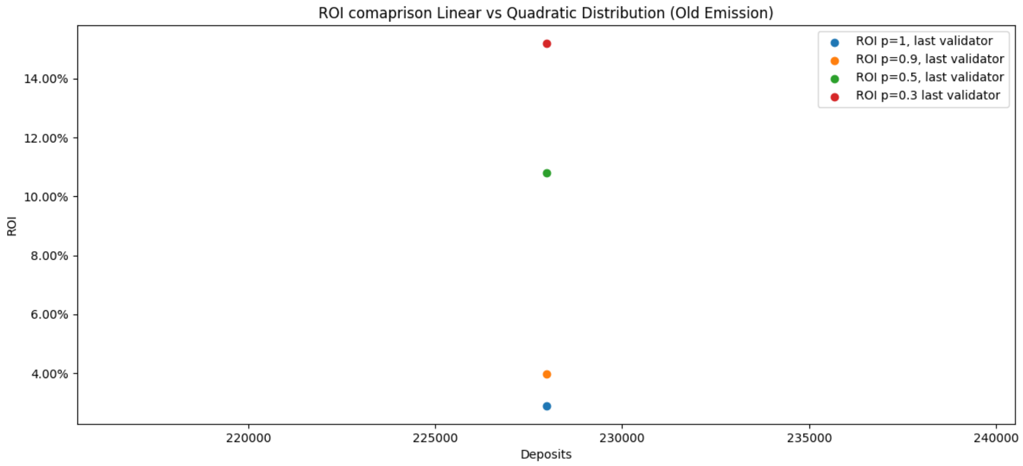


Figure 4.4: ROI last validator with old emision

We restrict ourselves to the cases $p = 0.3$, $p = 0.5$, and $p = 0.9$ as they show the broad impact of the quadratic distribution initiative, covering a range from a low value to a value closer to 1 and going through the middle point.

CONCLUSION: In summary, with the implementation of the new Quadratic Distribution system, top validators would experience a significant decrease in their earnings, leading to a much lower APR. Meanwhile, small to medium validators will experience an increase in APR. This suggests that the most profitable strategy under the new reward scheme is to be an average to small validator. The POL team proposes to choose $p = 0.5$ as it represents a reasonable compromise, reducing the maximum APR for the largest validators without affecting them excessively and, at the same time, rewarding small validators the most.

4.2 Weight Selection

In this section, we present the mathematical and statistical analysis used to find the weights we consider optimal, as well as what we understand by optimal. Furthermore, we will briefly explain the computational algorithm used to find them.

Rationale

The goal is to achieve a dynamic adaptation of the APR, based on market information, and keep it within a previously established range, $[r_i, r_s]$. Based on the emission rates of the Cosmos ecosystem and considering that the new design would significantly impact validator revenue, **we propose taking $\boldsymbol{r_i = \frac{0.02}{365}}$ and $\boldsymbol{r_s = \frac{0.08}{365}}$**. However, the algorithm used to find the weights is stated in terms of any r_i and r_s, so the choice of r_i, r_s is arbitrary. We suggest selecting both rates subject to the following factors.

- Maximum annual emission r_s competitive relative to the Cosmos ecosystem's emission.
- Minimum annual emission r_i that ensures a minimum payment to validators.
- Maximum and minimum emissions voted on by the community.

It is evident that we cannot guarantee that the dynamic part of $R(n)$ stays within the range $[r_i, r_s]$ for all possible market scenarios. Therefore, we propose to find the weights $\omega = (\omega_s, \omega_b)$ that minimize the annual average error of being away from r_i and r_s. Finally, **the weights will be updated once a month**.

Intuitions

The methodology to find ω is to simulate 50 market scenarios to represent the variables on which our emission function depends. Specifically, each scenario will consist of 30 daily realizations for the staking ratio b and the gas fee $BaseFee$. The idea of taking 50 scenarios is to attempt to find weights robust enough to be left unmodified for an entire month, covering a wide variety of scenarios.

For a given scenario, we will find the weights ω that ensure the daily emission for each realization of b and $BaseFee$ remains between r_i and r_s. We repeat this process for each scenario.

Once this is done, we will have 50 weights that are optimal but only for the scenario for which they were defined. We developed two ways to then choose a weight ω that is most suitable for all scenarios at once:

- Minimize daily emission: The first error metric is to select the weight that minimizes the daily emission for each scenario. That is, we will calculate the error for each scenario and take the average over all scenarios.
- Minimize monthly emission: The second error metric will be to minimize the monthly emission for the entire scenario, for each scenario. This means, instead of comparing the daily emissions of each scenario, we compare their sum and check if it falls between r_i and r_s on a monthly basis. Once this is done for each scenario, we take the average of the errors.

Finally, we will choose weights that meet the following criteria:

- Not be null, $w_s, w_b \neq 0$, so that we can ensure dynamic emission with respect to both the stake ratio and BaseFee.
- Minimize the minimum of the two previous error metrics.

Technical Definitions

In this section, we will technically define the previous intuitive ideas and present the results of the algorithm.

Firstly, the parameters used for the simulations of staked ratios are:

$$\begin{aligned}
b_0 &= 0.5 : \text{Initial ratio,} \\
dt &= 1 : \text{Time step (1 day),} \\
T &= 30 : \text{Time horizon (30 days),} \\
\mu &= drift\_staking\_diario : \text{Based on data,} \\
\sigma &= volatilidad\_staking\_diario * 10 : \text{Based on data,} \\
seed &= 2 \text{ (Seed used for reproducibility of randomly generated paths).}
\end{aligned}$$

The parameters used for the simulations of BaseFees are:

$$\begin{aligned} BaseFee_0 &= 1\_000\_000\_000 : \text{Initial ratio},\\ dt &= 1 : \text{Time step (1 day)},\\ T &= 30 : \text{Time horizon (30 days)},\\ \mu &= drift\_baseFee\_diario : \text{Based on data},\\ \sigma &= volatilidad\_baseFee\_diario : \text{Based on data},\\ seed &= 12 \text{ (Seed used for reproducibility of randomly generated paths).} \end{aligned}$$

The parameters of the emission function are:

$$\begin{aligned} a := f(0) &= last\_registered\_emission * r_s : \text{Initial token supply},\\ b^* &= 0.66 : \text{Optimal staked ratio},\\ \alpha_{\text{BaseFee}} &= e^{-9} : \text{Normalice to a manageable scale},\\ r_i &= \text{Lower (daily emission) bound voted},\\ r_s &= \text{Upper (daily emission) bound voted.} \end{aligned}$$

Let $1 \leq i \leq 50$ be a simulated scenario. Let

$$\begin{aligned} b_i &= (b_i(1), ..., b_i(30)),\\ BaseFee_i &= (BaseFee_i(1), ..., BaseFee_i(30)) \end{aligned}$$

be the realizations of the stake ratio b_i and BaseFee BaseFee_i for this fixed scenario, respectively.

We seek (ω_s^i, ω_b^i) that satisfy the following inequalities:

$$r_i < \omega_s^i \left(1 - \frac{b_i(t)}{b^*}\right) + \omega_b^i e^{-\alpha_{\text{BaseFee}} BaseFee_i(t)} \leq r_s, \quad (\forall t, 1 \leq t \leq 30). \tag{4.1}$$

This ensures that the dynamic part of $R(t)$ (using the weight (ω_s^i, ω_b^i)) is within the interval $[r_i, r_s]$.

Next, for each (ω_s^i, ω_b^i), we can calculate the verification percentage of equations (4.1) in the other 49 scenarios using two error metrics. Let

$$P_j^i(t) = \omega_s^i \left(1 - \frac{b_j(t)}{b^*}\right) + \omega_b^i e^{-\alpha_{\text{BaseFee}} BaseFee_j(t)}$$

be the dynamic rate generated by the weights (ω_s^i, ω_b^i) in the j-th scenario.

- **Average Daily Error**:

  Calculate the average daily error induced by (ω_s^i, ω_b^i)

  $$err_d((\omega_s^i, \omega_b^i)) = \frac{1}{49} \sum_{j=1}^{49} err_j^i,$$

  where

  $$err_j((\omega_s^i, \omega_b^i)) := \frac{1}{30} \sum_{t=1}^{30} \delta^j_{(\omega_s^i, \omega_b^i), r_i, r_s}(t)$$

  with

  $$\delta^j_{(\omega_s^i, \omega_b^i), r_i, r_s}(t) := \begin{cases} 1, & \text{if } r_i \leq P_j^i(t) \leq r_s,\\ 0, & \text{elsewhere.} \end{cases}$$

- **Average Monthly Error**: Calculate the average monthly error induced by (ω_s^i, ω_b^i)

$$err_m((\omega_s^i, \omega_b^i)) = \frac{1}{49}\sum_{j=1}^{49} \delta_{(\omega_s^i,\omega_b^i),30\cdot r_i,30\cdot r_s}(j)$$

with

$$\delta_{(\omega_s^i,\omega_b^i),r_i,r_s}(j) := \begin{cases} 1, & \text{if } r_i * 30 \leq \sum_{t=1}^{30} P_j^i(t) \leq r_s * 30, \\ 0, & \text{elsewhere.} \end{cases}$$

Furthermore, the optimal weights will be w_s^*, w_b^* such that

$$w_s^*, w_b^* = \arg\min_{(\omega_s^i,\omega_b^i)} \min(err_d((\omega_s^i, \omega_b^i)), err_m((\omega_s^i, \omega_b^i))),$$
$$w_s^*, w_b^* \neq 0,$$

Finally, to avoid depending on the seed used in the simulations and to cover a wide range of market events, we decided to choose the optimal weights as the average of the best weights for 100 different seeds. The average is a representative statistic since the observed empirical distribution of the weights closely resembles a normal distribution, as seen in Figure 4.5

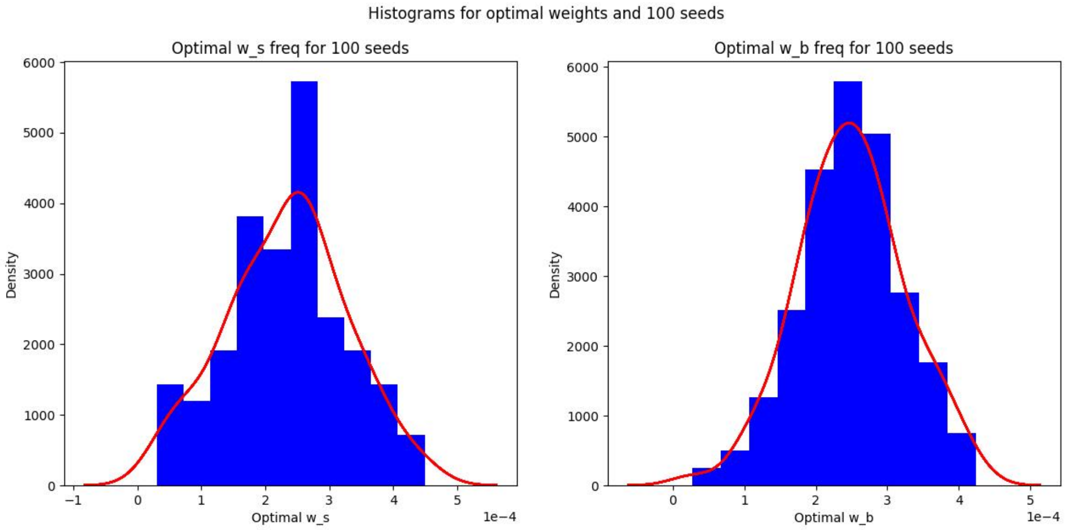


Figure 4.5: Optimal weights histogram across 100 seeds

We found that the optimal weights and most robust weights are

$$w_s^* = 0.000233, w_b^* = 0.000249.$$

This is shown in the Figure 4.6.

```
Optimal w_s = 0.000233, Optimal w_b = 0.000249
```

Figure 4.6: Optimal Weights

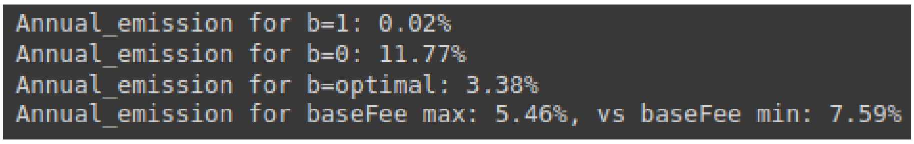
In Figure 4.7 we can appreciate that the mean monthly error across 50 scenarios for 100 different seeds is of 0.59%

```
Mean error for optimal weights across 50 scenarios and 100 seeds is 0.59%
```

Figure 4.7: Mean monthly error across 50 scenarios and 100 different seeds

Finally, as shown in Figure 4.8, we test the emission rate under the following specific scenarios

- We would not like to issue any token in the extreme case in which $b = 1$ and $BaseFee$ is any run, i.e. 100% of outstanding tokens are being used for staking purposes.
- We would like to issue at most r_s in the case $b = 0$ and $BaseFee$ is any run, i.e. no token is being used for staking purposes.
- We would like to issue something lower or equal to r_s if $b = b^*$ and $BaseFee$ is any run, as this would mean that we reached the optimal staking ratio.
- We would like the emission amount to be a decreasing function of the BaseFee. To check this, we compare the case $BaseFee = \max(BaseFee_j)$ vs the case $BaseFee = \min(BaseFee_j)$, and b any run, where we check whether $f(0.5, \max(BaseFee_j)) \leq f(0.5, \min(BaseFee_j))$ [if BaseFee is higher then Validators are earning higher miner tips, therefore we would not need to emit as many tokens as if the BaseFee was lower].



```
Annual_emission for b=1: 0.02%
Annual_emission for b=0: 11.77%
Annual_emission for b=optimal: 3.38%
Annual_emission for baseFee max: 5.46%, vs baseFee min: 7.59%
```

Figure 4.8: Behaviour checking for specific and extreme situations