$CUT: an algorithmic multi-chain rewards system

0xHisato

Keywords: Blockchain, Multi-Chain, Rewards System

1. $CUT Concept

$CUT is the native algorithmic rewards smart contract and token issued to users of the Steakwallet ecosystem as a reward for staking. Users of the Steakwallet ecosystem will receive $CUT for staking assets listed on the Steakwallet platform as a yield boost. The proposed system has three objectives:

1. Reward users for various transactions performed within a network (in its initial configuration for staking).
2. Function across blockchains and networks to allow multi-chain participation.
3. Provide high rewards during times of lower participation and lower rewards during high participation in order to mitigate inflation risks.

While $CUT will be deployed in the Steakwallet ecosystem, other projects could feasibly seek to participate by registering as an oracle in the $CUT contract.

2. Designing the Rewards Function

Table 1: Dataset of total amount staked vs. reward rate the users receive.

Ethereum Staked (TVL)	**Reward rate** ($RRate$)
250000	30.0
500000	20.0
2000000	10.0
4000000	7.5
6000000	6.0

We are starting the design of the $CUT incentive system by building a convex rewards function that takes into consideration Total Value Locked (TVL) measured in Ethereum (ETH) in an ecosystem and a Reward Rate at which users receive yield. Since the reward token is designed to be inflationary in nature, the objective is to provide incentives for users to stake assets in the ecosystem when TVL is low and to incrementally reduce rewards as

TVL increases to lower rates of inflation. In order to build this convex yield function, we turn to the rewards schedule proposed for Ethereum V2. The training dataset is constructed manually and presented in Table 1.

The dataset consists of 5 observations, 1 input/independent variable called Total Value Locked (TVL), and 1 output/dependent variable called *RRate*. The amount staked is considered as the independent variable while the reward rate is used as the dependent variable in order to study the impact of growth in TVL on reward rates users receive. To better illustrate the relationship among the input and output variables, the dataset is plotted in Figure 1. The relationship in the figure shows that the reward rate incrementally decreases as we increase TVL. The reward rate reduces rapidly as the amount is increased but the rate of reduction decreases as TVL approaches 6,000,000 units of ETH.

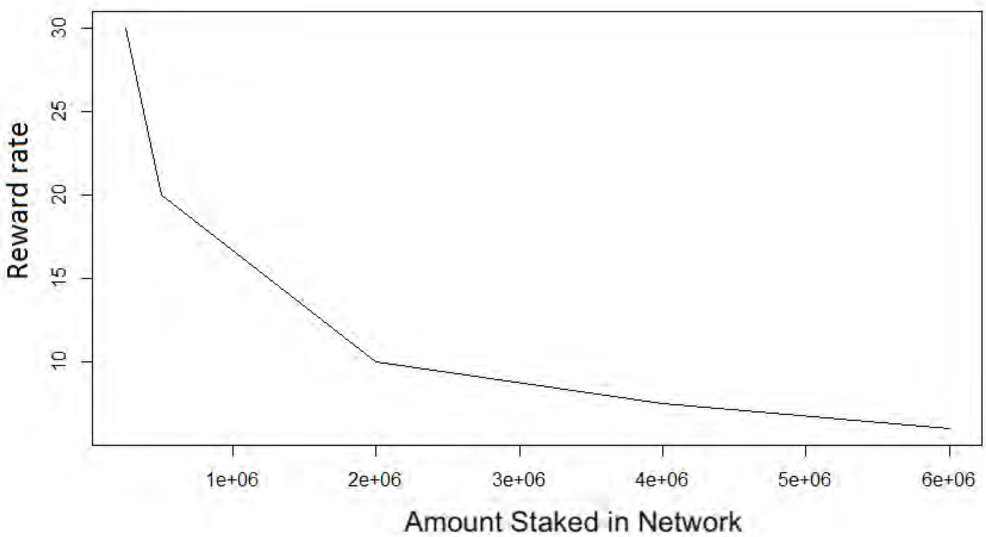

Figure 1: Illustration of the relationship among the total amount staked and reward rate they receive.

Although, Figure 1 gives good insight into the relationship of TVL and the associated reward rate it is important to further model it for comparatively larger amounts of TVL. To forecast the reward rate for TVL amounts in millions of units of ETH, a regression model is fitted on the dataset presented in Table 1.

2.1. Reward rate forecasting using log-transformed linear model

To forecast the reward rate for comparatively large TVL amounts, a linear regression model is fitted; however, by taking into account the assumption of violation in constant variance of data the dependent variable *RRate* is log-transformed before model fitting. It is important to mention that in regular linear models the independent variable has an additive relationship with the dependent variable; however, the log-transformed model built has a multiplicative relationship among the independent and dependent variables. For clarification, the log-transformed model is presented in Eq. (2).

$$y = \alpha e^{\theta_2 x} + e^{\theta_1} \quad (1)$$

It is determined through the model fitting that α = 22.17293, θ_1 = 0.40546510810816 and θ_2 = -2.973163e-07.

To forecast the reward rate large TVL numbers, the model is tested on numbers ranging from 1,000,000 to 100,000,000. NB that predictions for the test data are exponentiated because we used a log transformed dependent variable in the model fitting process. The predicted reward rates are plotted against TVL amounts in Figure 2.

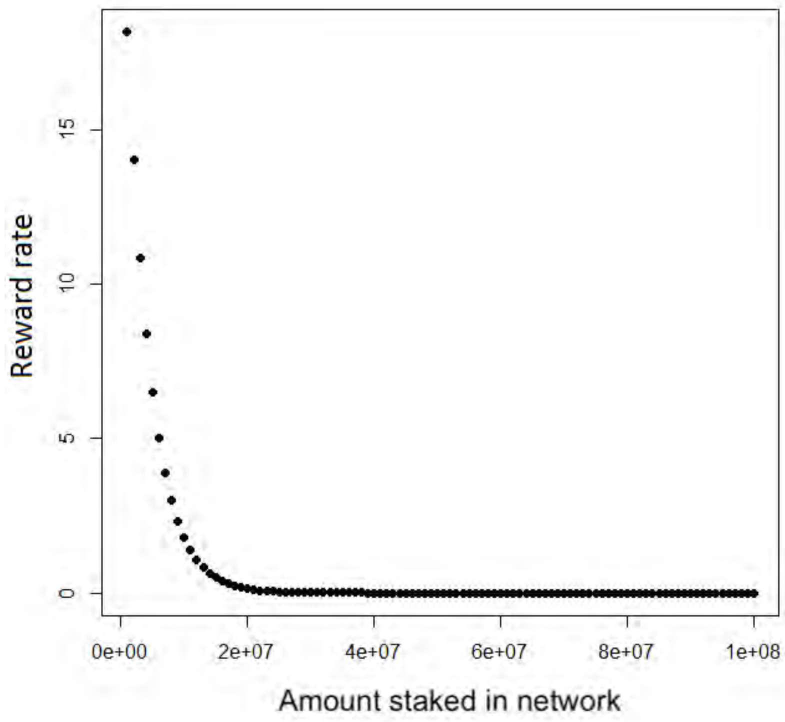

Figure 2: Forecast of the relationship among larger amounts and reward rates.

We can observe that the reward rate is very high for small amounts but quickly drops as the TVL amounts increase. For example, the reward rate is 17.97025% for 1,000,000 units of ETH and the reward rate reaches 2.6334% for 10,000,000 units of ETH. However, the reward rate starts stabilizing after 10,000,000 and reaches to 1.5% for 100,000,000 units of ETH (or equivalent).

In order to confirm the exactness of the modeled curve, the original curve is plotted on the modeled curve in Figure 3. It can be seen in Figure 3 that both curves overlap. The curve in red shows the original curve and the dotted curve in black color the modeled curve. Hence, it can be concluded that the predictions are probable.

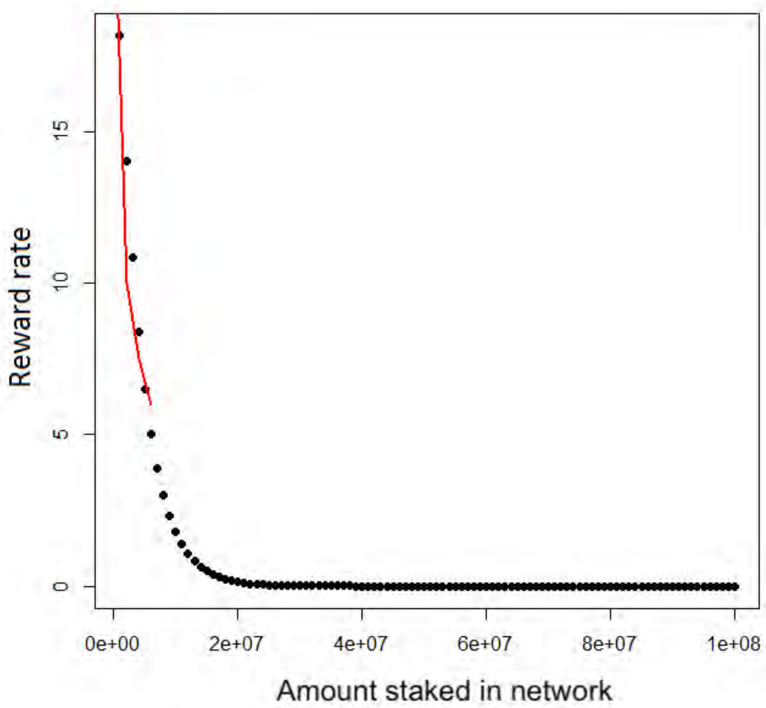

Figure 3: Mapping of the original curve on modeled curve.

3. Inflation Adjustment Coefficient

In order to mitigate the degree of inflation in the system due to tokens being minted as rewards are claimed, we are introducing an adjustment coefficient called k which can be manipulated to reduce the slope of the rewards curve, thereby decreasing the amount of TVL needed to approach the asymptote of 1.5%. The value for k can be utilized as the primary lever to adjust inflation in the \$CUT smart contract. In order to allow for a simplification of the conversation around the inflation adjustment, we are denominating k as an integer, yielding the following equation:

$$y = \alpha e^{(\theta_2 - k[10^{-7}])x} + e^{\theta_1} \tag{2}$$

In order to account for smaller amounts of TVL in the early days of the ecosystem, we are setting k=20, resulting in rewards and thus inflation decreasing at a faster rate as can be seen in Figure 4.

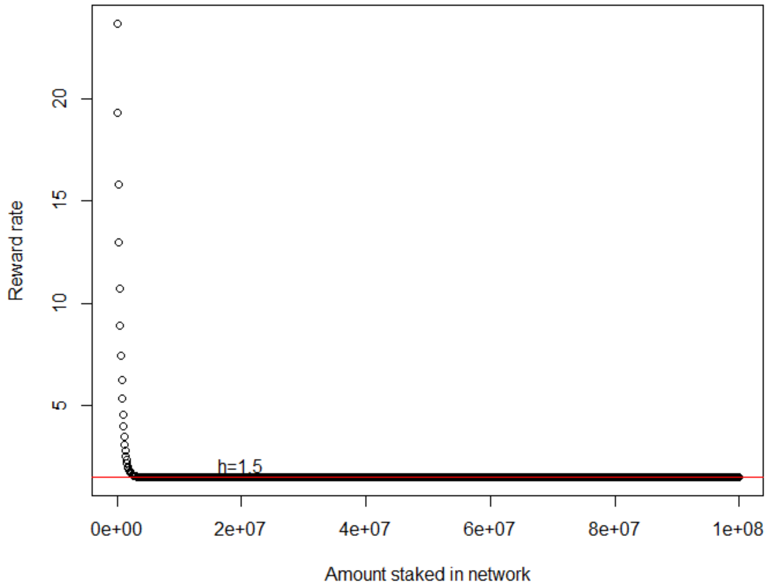

Figure 4: Modeled curve with adjustment coefficient k.

4. Reward calculation method

Next, we are formulating the reward function for participating ecosystems, in this case Steakwallet which will act as the first oracle in the $CUT contract. The user will receive rewards in $CUT in accordance with a boost rate ($BRate$) that is applied to a given Annual Percentage Reward (APR), which will increase linearly with the number of projects a user stakes on. This linear growth coefficient will remain constant once 10 projects have been staked on. In this section, it is discussed how the total reward can be calculated from total units of Ethereum (TVL) in the system.

4.1. Calculation of the boost rate

The boost rate encourages users to invest in more than one projects. The boost for pth project can be calculated by using the following equation:

$$BRate(p) = RRate * p \tag{3}$$

while total boost can be computed using the following formula:

$$BRate = \sum_{i=1}^{n} BRate(i) \tag{4}$$

where n denotes total number of projects a user is staking in, $BRate$ denotes the boost rate, $RRate$ denotes the reward rate computed in the previous section through the regression model, and n represents the number of projects the user is currently staking in. It is important to mention here that the maximum value of n is N, which is the total number of projects in the network.

4.2. Calculation of the $CUT rate

As described earlier, $CUT is the extra reward that user receives by stakeing. The formula to calculate $CUT rate for any project/currency p is presented in the following equation:

$$CRate(p) = APR(p) * BRate \tag{5}$$

where $CRate(p)$ is the $CUT rate for project p, $APR(p)$ is the Annual Percentage Return for the project p, and $BRate$ is the boost rate calculated in Eq. (3).

4.3. Calculation of the total reward

Total reward is the sum of the actual reward and extra reward. The extra reward is calculated using $CUT rate. Below is the equation to calculate the extra reward:

$$CReward(p) = ETH(p) * CRate(p) \tag{6}$$

where $ETH(p)$ denotes the amount staked in terms of units of Ethereum (or equivalent) in project p and $CReward(p)$ is the $CUT reward for project p. The total reward can be calculated in terms of Ethereum by using the following equation:

$$Reward(p) = ETH(p) * APR(p) + CReward(p) \tag{7}$$

$$Reward(p) = ETH(p) * (APR(p) + CRate(p))$$

where $Reward(p)$ is the total reward that user will receive for staking for project p. The total $CUT rewards for all projects can be computed using the following equation:

$$CReward = \sum_{p=1}^{n} CReward(p) = \sum_{p=1}^{n} ETH(p) * CRate(p) \tag{8}$$

where n is the number of projects the user staking in. Similarly, the total reward for all projects can be computed as:

$$Reward = \sum_{p=1}^{n} Reward(p) = \sum_{p=1}^{n} ETH(p) * (APR(p) + CRate(p)) \tag{9}$$