$olid - A dynamic self-liquidated token that solve liquidity problem

Kwan Justin Chi Yee, Jayson Rahmlow

Abstract. Cryptocurrencies being traded nowadays could run out of liquidity from time to time, due to massive withdrawal and deposit in a short period of time, it causes huge volatility that people who provide liquidity ends up losing money slippage. We provide a mechanism such that high volatility could be avoided through a dynamic token-staking algorithm which would balance out liquidity when trading.

1. Introduction

A simple trading formula,x \* y = k. proposed by uni-swap, is used to determine the price of token based on the supply and demand. However, this leads to whales with multi-millions worth of cryptocurrency pump and dump the price due to its delayed adjustment of price. Hence, we proposed a more advanced algorithm to avoid so.

This paper will discuss on such mechanism and test that work, we will not ensure

This method is proposed as an experiment, an algorithm that is proposed in this paper could be changed as in response of the community.

By introducing a global value to store individual coin's price, only liquidity of individual coin is needed to be added into the liquidity pool (not pair), and adjusting such value according to the inflow and outflow of the coin, user will be able to trade from coin to coin without the needs of liquidity mapping for all individual coin pair.

1. Implementation

Given as the supply of the token, the price of the token, , is determined by the following formula:

(slippage allowed)

In real-life scenario, when a buyer buys the token, the buyer cannot buy all the token he wants in the current price. Exchanges mimic the effect that, the token’s price goes up as the buyer buys in more coin in a dynamic way, which is called price impact. Price impact normally could be deduced to a simple formula since market normally adjust price linearly.

All price impact could be calculated with definite integral given as the formula for calculating price

Since the proposed method does not adjust the price in a linear way, therefore, we would need to deduce our own price impact formula

Let α be the initial supply of the token, β be the final supply of the token

And let ẟ = β - α, which is the total amount of token bought by the buyer, the below formula is used to calculate the total price of all tokens that the buyer needs to pay for buying ẟ token(s).

)

Mechanism

* The only way coins are minted and burnt is through buying and selling to the contract.
* Coin is always minted when bought and burnt when sold to the contract.

When buyers buy the coin on the blockchain, the coin is minted and the coin is sent to the buyer's wallet, total supply is increased by the amount that is bought in.

Vice versa, when sellers sell the coin on the blockchain, the coin is burnt, total supply is decreased by the amount that is sold.

Each coin structure is mapped with the following attribute:

* Amount of coin in the liquidity pool(T)
* Price of the coin(P)

Let d(Delta) be the number of coin being bought/sold, and T be the amount of coin in the liquidity pool

Price of an individual coin is adjusted by the formula: P = P(d+T)/T.

For instance, coin A has a supply of 100 in liquidity pool (alpha) and price of 1.2, and coin B has a supply of 200 in liquidity pool (beta) and price of 1.

coin A{

T = 100

P = 1.2

}

coin B{

T = 200

P = 1

}

If a user swap 30 coin A for coin B, price ratio would be (P of coin A / P of coin B)=1.2/1=1.2 , hence, amount of coin B swapped in = 1.2\*30, = 36, coin A liquidity pool will +30 coin, and coin B liquidity pool will -36 coin, resulting in 100+30 = 130 for coin A, and 200-36 = 164 for coin B. Price will be changed accordingly.

Mapping of coin liquidity pool

Since token has its own contract address, by mapping the token address into the a pointer structure that holds the information of the token,

1. Result
2. Summary

Dynamic gas according to bid-ask