FreeSwap Exchange Protocol

Daoru Lu
ldru@163.com
January 2021

Abstract

FreeSwap protocol is a decentralized exchange protocol that does not charge any exchange fees. Based on the formula "constant-product invariant", Freeswap creates two one-way-swap sub-pools for each pair of tokens. As the swap going, when the token price deviation in the two sub-pools reaches a certain extent, the sub-pools carry out arbitrage operation with each other. This arbitrage operation can rebalance the token prices in the sub-pools and make profits for liquidity providers. This paper mainly describes the Freeswap arbitrage mechanism and its basic rules, expounds its win-win characteristics, and evaluates the arbitrage profit level quantitatively through a theoretical model. It is shown that Freeswap can achieve the arbitrage profit equivalent to 2.488 ‰ swap fee while arbitraging at sub-pool price deviation of 1%.

0 Introduction

In 2020, "constant-product invariant" formula [1] made a great success in decentralized exchanges. This formula is extremely concise, but it can automatically determine the token price in the liquidity pools. DEXs based on the this formula, such as Uniswap[2][3] and SushiSwap, can provide competitive swapping liquidity and trading depth, and attract a large number of cryptocurrency investors.

However, in addition to paying the gas fee of blockchain network, DEX transactions also need to pay a certain percentage of enchange fees, which is some higher than centralized exchanges. The purpose of Freeswap protocol is to design a completely free exchange protocol that does not require to pay any exchange fees. It is expected that by removing exchange fees, more users can be attracted to join DEXs and leave centralized exchanges.

To solve the problem of "front running attack", Vitalik Buterin once proposed to set up two one-way trading sides for a token pair[4]. Freeswap protocol refines his proposal and defines the implementation details. Since the two onw-way trading pools can only exchange tokens in one direction, this will inevitably lead to the reverse deviation of token prices between two trading pools. This deviation can not only stop "front-running attack" of miners, but also provide a way to make profits for liquidity providers. Freeswap protocol makes full use of the arbitrage opportunities generated by the price deviations between the two sub-pools to provide returns to liquidity providers, and further, to serve the exchange users totally with no trade fee.

1 Basics of Liquidity Pool

1.1 Swap pair definition

The Swap pair consists of two different tokens that can be exchanged, represented by $TokenA$ and $TokenB$, respectively. When creating a swap pair, the user needs to deposit corresponding number of $TokenA$ and $TokenB$ with equal value according to the actual market price. If the price differs from the market price, the pair will be arbitraged and the creator will suffer a loss.

Assuming P_A and P_B are respectively the price of $TokenA$ and $TokenB$ in any legal currency, $P_{A\to B}$ is defined as the price exchanging $tokenA$ for $TokenB$, which is the amount of $TokenB$ equivalent to $1\ TokenA$:

$$P_{A\to B} = \frac{P_A}{P_B} \tag{1.1.1}$$

Similarly, define $P_{B\to A}$, which is the price at which $TokenB$ is traded to $TokenA$:

$$P_{B\to A} = \frac{P_B}{P_A} \tag{1.1.2}$$

In theory, without considering trading costs, the following relationship exists:

$$P_{A\to B} * P_{B\to A} = 1 \tag{1.1.3}$$

Assuming N_A and N_B are the quantities of two digital assets $TokenA$ and $TokenB$ at any time within the trade pair, we express the trade pair as:

$$(N_A \mid N_B) \tag{1.1.4}$$

The trade pair always assume two internal tokens have equal market values all the time, and so, following formula are equivalent:

$$N_A * P_A = N_B * P_B \tag{1.1.5}$$

$$N_A = N_B * P_{B\to A} \tag{1.1.6}$$

$$N_B = N_A * P_{A\to B} \tag{1.1.7}$$

$$P_{A\to B} = \frac{N_B}{N_A} \tag{1.1.8}$$

$$P_{B\to A} = \frac{N_A}{N_B} \tag{1.1.9}$$

1.2 Constant-product invariant

While performing token exchange, since the external prices of $TokenA$ and $TokenB$ cannot be easily obtained, it is needed to design a mechanism to determine the exchange ratio between $TokenA$ and $TokenB$. The DEXs of Automated Market Maker (AMM) type uses the "constant-product invariant" formula [1] to determine the change of the number of tokens in the trading pool before and after the exchange, also the amount of tokens traded in and out accordingly.

For the pair of $(N_A \mid N_B)$, "constant-product invariant" formula is represented as:

$$N_A * N_B = K \tag{1.2.1}$$

In this formula, the value of K changes only when the users deposit tokens to, or remove tokens from the trading pair, and alway remains constant during the trading process.

Suppose in a exchange oparetion, the user trades $TokenA$ for $TokenB$, and the input amount of $TokenA$ is Δ_A, the output amount of $TokenB$ is Δ_B. ("Trade in" and "trade out" here after are from the point of liquidity pool, if from the point of users, the relationship "trade in" and "trade out" is completely opposite). According to the "constant-product invariant" formula, there are:

$$(N_A + \Delta_A) * (N_B - \Delta_B) = (N_A * N_B) \tag{1.2.2}$$

$$\Delta_B = \frac{\Delta_A}{N_A + \Delta_A} * N_B \tag{1.2.3}$$

It can be seen that there are three different prices here, before the trade, after the trade, and happend in the trade.

The token price of the pool before the trade (expressed as $P^0_{A \to B}$):

$$P^0_{A \to B} = \frac{N_B}{N_A} \tag{1.2.4}$$

The actual token price taken in the trade (represented as $P^1_{A \to B}$) is:

$$P^1_{A \to B} = \frac{\Delta_B}{\Delta_A} = \frac{N_B}{N_A + \Delta_A} \tag{1.2.5}$$

The token price after the trade finished (represented as $P^2_{A \to B}$) is:

$$P^2_{A \to B} = \frac{N_B - \Delta_B}{N_A + \Delta_A} \tag{1.2.6}$$

Obviously:

$$P^0_{A \to B} > P^1_{A \to B} > P^2_{A \to B} \tag{1.2.7}$$

It means, the price of trade-in token $TokenA$ relative to the trade-out token $TokenB$ has slipped after the trade has finised. The price of $TokenA$ in the pair before the trade starts is higher than the actual price the trade adopts, and the actual price the trade adopts is higher than the price of $TokenA$ in the pair after the trade is completed. That is, after the token trade is completed, the price of $TokenA$ relative to $TokenB$ in the pair has fallen, and accordingly, the relative price of $TokenB$ has risen.

1.3 Trade arbitrage

The exchange operation happens at the moment the transaction block is confirmed on the blockchain. The external prices of $TokenA$ and $TokenB$ will not change at that moment. Due to the constraint of "constant-product invariant", the token price will slide within the exchange, which will cause users to suffer

a certain marginal exchange loss. The exchange loss in the amount of $TokenB$ is calculated as follows:

$$\begin{aligned} Lost_{A\to B} &= \Delta_A * \frac{N_B}{N_A} - \Delta_A * \frac{N_B}{N_A+\Delta_A} \\ &= \Delta_A * \frac{N_B}{N_A} \Big/ (1+\frac{N_A}{\Delta_A}) \end{aligned} \tag{1.3.1}$$

It can be seen that the greater the ratio (Δ_A/N_A) the user trade $TokenA$ into the pool, the greater the exchange loss it will suffer.

After the exchange is completed, if another user performs reverse exchange, according to the "constant-product invariant" formula, he only needs to pay $TokenB$ in the amount of Δ_B, and he can get $TokenA$ of Δ_A. After this reverse exchange, the asset price in the pool will return to the initial price $P_{A\to B} = N_B/N_A$. That is, this user completes the exchange at the higher $TokenB$ to $TokenA$ price, and realizes the arbitrage of the previous user's exchange loss, the arbitrage profit is equal to the previous user's exchange loss.

1.4 Impermanent loss

When providing liquidity, the provider needs to inject into the trading pool two tokens of corresponding amounts with the same total value. With the trade on going, the number of tokens corresponding to the user's liquidity will change. Assuming that the user's initial investment of $TokenA$ is X_A, when the user exits liquidity, if the market price of $TokenA$ rises relative to $TokenB$, then ${X'}_A$, the amount of $TokenA$ that the user can get from the pool may be less than X_A. If the liquidity service income obtained by providing liquidity are not enough to compensate for the loss caused by the decrease in the amount of $TokenA$, in this case, the liquidity provider will suffer losses relative to simply holding $TokenA$ and $TokenB$ instead of providing liquidity services. In fact, no matter whether the price of $TokenA$ and $TokenB$ goes up or down, as long as their price deviates from the time providing liquidity, the user will suffer losses. The amount of the loss is completely determined by the market price variation, beyond any user's control, which is also called "Impermanent Loss" [5].

Taking $TokenB$ as the price benchmark, following quantitative analysis of impermanent loss is conducted without considering the exchange fee income. Assuming that the initial number of tokens that the user put into the trade pair is $(X_A|Y_B)$, the price of $TokenA$ relative to $TokenB$ is P_1, and at the end, the number of tokens that the user withdraws from the trade pair is $(X'_A|Y'_B)$, and the price of $TokenA$ relative to $TokenB$ is P_2, if the user does not provide liquidity for the trading pool, but simply holds $(X_A|Y_B)$ tokens, the final value is:

$$V_1 = X_A * P_2 + Y_B \tag{1.4.1}$$

Since the user provides liquidity for the trading pool, the actual value of $(X'_A|Y'_B)$ tokens is:

$$V_2 = X'_A * P_2 + Y'_B \tag{1.4.2}$$

Considering:

$$\begin{cases} X_A * Y_B = X'_A * Y'_B \\ P_1 = Y_B / X_A \\ P_2 = Y'_B / X'_A \end{cases} \tag{1.4.3}$$

Then there is [6]:

$$\begin{aligned} \frac{V_2}{V_1} &= \frac{2\sqrt{P_1 * P_2}}{P_1 + P_2} \\ &= \sqrt{\frac{4P_1P_2}{(P_1 - P_2)^2 + 4P_1P_2}} \leq 1 \end{aligned} \tag{1.4.4}$$

So, as long as $P_1 \neq P_2$, V_2 is always less than V_1, that is to say, the users's total token value denominated in $TokenB$ must be reduced compared to simply holding two tokens. Similarly the total token value denominated in $TokenA$ must also be reduced compared to simply holding two tokens. Therefore, the so-called "Impermanent Loss" is actually a permanent loss that is bound to occur, only the amount of loss may change with the price ratio.

2. FreeSwap exchange protocol

2.1 FreetSwap protocol goal

The goal of freeswap protocol is to implement a decentralized exchange protocol with no trade fees at all. The "constant-product invariant" mechanism inevitably leads to token price slippage, from which Freeswap could make profits for liquidity provider by arbitraging automatically.

2.2 FreeSwap trade pair setup

FreeSwap trade protocol sets up two independent one-way sub-pools for the token pair $(Token_A|Token_B)$, expressed as:

$$(N_{AA}|N_B) \ || \ (N_A|N_{BB}) \tag{2.2.1}$$

Where, the sub-pool $(N_{AA}|N_B)$ (called as A sub-pool, or A pool hereinafter) can trade in $TokenA$, trade out $TokenB$, N_{AA} is the number of $TokenA$, and N_B is the number of $TokenB$ in A pool.

The sub-pool $(N_A|N_{BB})$ (called as B sub-pool, or B pool hereinafter) can trade in $TokenB$, trade out $TokenA$, N_{BB} is the number of $TokenB$, and N_A is the number of $TokenA$ in B pool.

When the liquidity provider adds tokens into the trading pool, he can specify which sub-pool to add, or he can add into both the sub-pools at the same time with specified ratio.

2.3 FreetSwap token exchange

When the user trades $TokenA$ and $TokenB$ with the trading pool, there are two opposite trading operations, one is to trade in $TokenA$, and trade out $TokenB$; the other is to trade in $TokenB$, and

trade out $TokenA$. Here, "trade in" and "trade out" are relative to the trading pool.

If the user trades $TokenA$ for $TokenB$, FreeSwap will pass this trade to A pool, the $(N_{AA}|N_B)$ sub-pool to process. The A pool always trades in $TokenA$ and swap out $TokenB$, so the number of N_{AA} always goes up, and the number of N_B always goes down, which will lead to the monotonous decrease of $tokenA$ price denominated by $TokenB$, and finally resulting $TokenA$ price deviating from actual market price.

Similarly, if the user trade in $tokenB$ for $tokenA$, B pool will handle this trade. B pool always trades in $tokenB$, and swaps out $tokenA$, so the amount of N_{BB} will keep rising, and the amount of N_A keep falling, which will also cause that B price denominated in $TokenA$ decreases monotonically and eventually deviates from the actual market price.

The sub-pool $(N_{AA}|N_B)$, $(N_A|N_{BB})$ are two one-way trading pairs in opposite directions, they are always fixed at one token in and the other token out. As the trades happening, the prices of the pair tokens always slide in opposite directions. After some trade volume accumulation, the price difference of the same token in the two sub-pools will exceed a certain level, which is too large to prevent users from participating the exchange. At this time, it is necessary to internally exchange the tokens between the sub-pools to smooth the token price difference. The process of internal token exchange within the sub-pools is actually a arbitrage process. Trading users conduct trade in two one-way trading pools, resulting token prices one-way slidding, and users will suffer trade losses from this price slippage. Through the internal exchange between two sub-pools, price slippage can be repaired, and the arbitrage to user trade slippage losses can be realized, which may make profits for the liquidity providers of the pool.

2.4 FreeSwap exchange arbitrage

As said above, FreeSwap sets up two independent sub-trading pools:

$$(N_{AA}|N_B)\ ||\ (N_A|N_{BB}) \tag{2.4.1}$$

The prices of $TokenA$ and $TokenB$ in the two sub-pools will deviate as exchange occurs. When the price deviates to some extent, the two sub-pools need to internally exchange tokens to restore the token price. The token exchange between two sub-pools can restore the token prices of both side, and also realize arbitrage between the pools. The analysis following:

Before arbitrage, $TokenA$ prices in two pools are:

$$P_{AA\to B} = \frac{N_B}{N_{AA}} \tag{2.4.2}$$

$$P_{A\to BB} = \frac{N_{BB}}{N_A} \tag{2.4.3}$$

The deviation ratio is:

$$R_{PA} = \frac{P_{A\to BB}}{P_{AA\to B}} = \frac{N_{AA} * N_{BB}}{N_B\ *\ N_A} \tag{2.4.4}$$

Similarly, $TokenB$ prices in the two pools are:

$$P_{BB\to A}=\frac{N_A}{N_{BB}} \tag{2.4.5}$$

$$P_{B\to AA}=\frac{N_{AA}}{N_B} \tag{2.4.6}$$

The deviation ratio is:

$$R_{PB}=\frac{P_{B\to AA}}{P_{BB\to A}}=\frac{N_{AA}*N_{BB}}{N_B*N_A} \tag{2.4.7}$$

It can be seen that the price ratios of $TokenA$ and $TokenB$ in the two sub-pool are the same, namely:

$$R_{PA}=R_{PB}=R_P=\frac{N_{AA}*N_{BB}}{N_B*N_A} \tag{2.4.8}$$

The FreeSwap protocol specifies that when the token prices of two sub-pool deviate to a certeain threshold γ, such as the value of R_P is greater such as 101%, the internal arbitrage exchange is automatically executed to restore the token price.

Arbitrage operation is to exchange one token with more amount in one sub-pool for the other token with more amount in the other sub-pool by equivalent value, that is, $(N_{AA}|N_B)$ sub-pool swap $TokenA$ for $TokenB$ with same value in $(N_A|N_{BB})$. From another perspective, it is for $(N_A|N_{BB})$ sub-pool to exchange $TokenB$ for $TokenA$ in $(N_{AA}|N_B)$ sub-pool with equal value.

To ensure exchange fairness, the exchange price is set as the average price of all tokens in the two sub-pools, namely:

$$P^e_{A\to B}=\frac{N_B+N_{BB}}{N_A+N_{AA}} \tag{2.4.9}$$

$$P^e_{B\to A}=\frac{N_A+N_{AA}}{N_B+N_{BB}} \tag{2.4.10}$$

$$P^e_{A\to B}*P^e_{B\to A}=1 \tag{2.4.11}$$

After the arbitrage completed, the two sub-pools can be expressed as:

$$(N_{AA}-L_A\mid N_B+L_B)\;||\;(N_A+L_A\mid N_{BB}-L_B) \tag{2.4.12}$$

That is, in this trade arbitrage, A pool exchanges $TokenA$ with a quantity of L_A for $TokenB$ with a quantity of L_B. Considering the exchange value is equal, the following equation exists:

$$L_B=L_A*P^e_{A\to B} \tag{2.4.13}$$

After the arbitrage operation, $TokenA$ price of the A pool (priced in $TokenB$) will increase, but it should not exceed the average price of the overall trading pool $P^e_{A\to B}$ for the reason of fairness. Similarly, $TokenB$ price (priced in $TokenA$) in B pool will also rise, but either should not exceed the average price $P^e_{B\to A}$, That is, the following relationship exists:

$$\frac{N_B+L_B}{N_{AA}-L_A}\le P^e_{A\to B} \tag{2.4.14}$$

$$\frac{N_A + L_A}{N_{BB} - L_B} \leq P^e_{B \to A} \tag{2.4.15}$$

Combining $(2.4.13) - (2.4.15)$, following relationships can be obtained:

$$L_A \leq \frac{N_{AA} * N_{BB} - N_A * N_B}{2 * (N_B + N_{BB})} \tag{2.4.17}$$

$$L_B \leq \frac{N_{AA} * N_{BB} - N_A * N_B}{2 * (N_A + N_{AA})} \tag{2.4.18}$$

In order to reduce the times of arbitrage operation, arbitrage between sub-pools should exchange as many tokens as possible to smooth token price deviations to the greatest extent. So L_A, L_B should take the maximum vaue given by $(2.4.17)$ and $(2.4.18)$. It is easy to find, in this case, the token prices of the two sub-pool both happen to be restored to the average asset price of the whole trade pool $P^e_{A \to B}$, which is also the price at which the two sub-pools exchange tokens.

In general, Freeswap arbitrage can be expressed as follows:

$$\begin{cases} \frac{N_{AA} * N_{BB}}{N_A * N_B} \ \geq \ \gamma \\ P^e_{A \to B} \ = \ \frac{N_B + N_{BB}}{N_A + N_{AA}} \\ L^e_A \ = \ \frac{N_{AA} * N_{BB} - N_A * N_B}{2 * (N_B + N_{BB})} \\ L^e_B \ = \ \frac{N_{AA} * N_{BB} - N_A * N_B}{2 * (N_A + N_{AA})} \end{cases} \tag{2.4.19}$$

Among the above equation, γ is the condition for triggering the arbitrage operation, which means that arbitrage is executed when the price deviation of the two sub-pools is greater than or equal to $\gamma - 1$. L^e_A, L^e_B are the amount of tokens exchanged between the two sub-pools within arbitrage operations. Arbitrage is taken by exchanging L^e_A, L^e_B amount of $TokenA$, $TokenB$ at the price of $P^e_{A \to B}$, on arbitrage completion, the token prices of the two sub-pools are exactly the same, both are $P^e_{A \to B}$.

3 FreeSwap arbitrage analysis

3.1 FreeSwap arbitrage analysis

From the perspective of the liquidity provider, for the internal exchange of the two sub-pools, ie. arbitrage operations, following considerations should be taken:

1. Pool A and Pool B must both obtain positive returns in arbitrage operations. If one party gains and the other loses, the arbitrage mechanism cannot be established;
2. Pool A and Pool B both expect to maximize their profits in the arbitrage operations. The ideal goal of the arbitrage mechanism is to maximize the profits of both sub-pools simultaneously.
3. The arbitrage return of two sub-pools should be balanced reasonably. If one side gains more and the

other side gains less, the arbitrage mechanism is also not perfect.

FreeSwap arbitrage protocol can meet the above three requirements, that is, it can realize the maximum positive profit for both sides of the trading pools, and the profits of both sides are equal.

The arbitrage profit can be measured by the changes of "constant-product invariant" of the sub-pool before and after the arbitrage operation. After the arbitrage completion, the changes in the K value of the two sub-pools are analyzed below.

For the A sub-pool, the change in K value after arbitrage is:

$$\begin{aligned}\Delta K_A &= (N_{AA} - L_A) * (N_B + L_B) - N_{AA} * N_B \\ &= N_{AA} * L_B - L_A * N_B - L_A * L_B \\ &= -P^e_{A\to B} * L_A^2 + (N_{AA} * P^e_{A\to B} - N_B) * L_A \\ &= -P^e_{A\to B} * \left(L_A - \frac{N_{AA} - N_B * P^e_{B\to A}}{2}\right)^2 + \frac{(N_{AA} - N_B * P^e_{B\to A})^2}{4 * P^e_{B\to A}}\end{aligned} \tag{3.1.1}$$

It can be seen that for A pool, while $L_A = L_A^M$, K value increases the most, that is, A pool gets the maximum arbitrage profit as:

$$\begin{aligned}L_A^M &= \frac{N_{AA} - N_B * P^e_{B\to A}}{2} \\ &= \frac{N_{AA} * N_{BB} - N_A * N_B}{2 * (N_B + N_{BB})}\end{aligned} \tag{3.1.2}$$

It is obvious that L_A^e in $(2.4.19)$ equals to L_A^M, it means that FreeSwap arbitrage mechanism can increase K value of the A pool to its maximum:

$$\begin{aligned}\Delta K_A^M &= \frac{(N_{AA} - N_B * P^e_{B\to A})^2}{4 * P^e_{B\to A}} \\ &= \frac{(N_{AA} * N_{BB} - N_A * N_B)^2}{4 * (N_A + N_{AA}) * (N_B + N_{BB})}\end{aligned} \tag{3.1.3}$$

Similarly, for the B sub-pool, the K value changes after arbitrage as follows:

$$\begin{aligned}\Delta K_B &= (N_{BB} - L_B) * (N_A + L_A) - N_{BB} * N_A \\ &= N_{BB} * L_A - L_B * N_A - L_A * L_B \\ &= -P^e_{B\to A} * L_B^2 + (N_{BB} * P^e_{B\to A} - N_A) * L_B \\ &= -P^e_{B\to A} * \left(L_B - \frac{N_{BB} - N_A * P^e_{A\to B}}{2}\right)^2 + \frac{(N_{BB} - N_A * P^e_{A\to B})^2}{4 * P^e_{A\to B}}\end{aligned} \tag{3.1.4}$$

Similarly for the B pool, when $L_B = L_B^M$, the K value has the largest increase, that is, the B pool gets the maximum arbitrage profit:

$$
\begin{aligned}
L_B^M &= \frac{N_{BB} - N_A * P_{A\to B}^e}{2} \\
&= \frac{N_{AA} * N_{BB} - N_A * N_B}{2 * (N_A + N_{AA})}
\end{aligned} \tag{3.1.5}
$$

Samely, L_B^e in $(2.4.19)$ is equal to L_B^M, that is, FreeSwap arbitrage mechanism can increase K value of B pool to the maximum:

$$
\begin{aligned}
\Delta K_B^M &= \frac{(N_{BB} - N_A * P_{A\to B}^e)^2}{4 * P_{A\to B}^e} \\
&= \frac{(N_{AA} * N_{BB} - N_A * N_B)^2}{4 * (N_A + N_{AA}) * (N_B + N_{BB})}
\end{aligned} \tag{3.1.6}
$$

Comparing $(3.1.3)$, $(3.1.6)$ and $(2.4.19)$, we can see that:

$$
\Delta K_B^M \equiv \Delta K_A^M \equiv L_A^e * L_B^e \tag{3.1.7}
$$

This means that FreeSwap arbitrage protocol can simultaneously maximize the increment of K Value both for the A pool and B pool by arbitrage, and the K value increment in the two sub-pool are exactly the same, which equals to the product of the amount of tokens internaly exchanged within arbitrage between the sub-pools.

3.2 FreeSwap arbitrage profit

To simplify the calculation, it is assumed that the Freeswap trading and arbitraging process is as follows:

1. The A pool and the B pool initially have the same amount of two tokens, which are represented by X and Y respectively;
2. In A pool, the user exchange $TokenA$ with a quantity of x for $TokenB$ with a quantity of y, causing the token price of the A pool to slide. And the sliding range is γ, triggering arbitrage operation between A pool and B pool;
3. After arbitrage, the number of tokens in the A pool and the B pool becomes $(X + x_1' \ , \ Y - y_1')$ and $(X + x_2' \ , \ Y - y_2')$;

The changes in the amount of $Tokens$ during the trade and arbitrage process are shown in the following table:

$Sub\ Pool:$	$A\ pool$	$B\ Pool$
$Token\ State:$	$(N_{AA} \ , \ N_B)$	$(N_A \ , \ N_{BB})$
$Initial\ State:$	$(X \ , \ Y)$	$(X \ , \ Y)$
$Post\ Trade:$	$(X + x \ , \ Y - y)$	$(X \ , \ Y)$
$PostArbitrage:$	$(X + x_1' \ , \ Y - y_1')$	$(X + x_2' \ , \ Y - y_2')$

According to the "constant-product invariant" formula, there are:

$$
(X + x) * (Y - y) = X * Y
$$

$$y = \frac{x}{X + x} * Y \tag{3.2.1}$$

According to the arbitrage triggering conditions in $(2.4.19)$, there are:

$$(X + x) * Y = \gamma * X * (Y - y)$$

$$x = (\sqrt{\gamma} - 1) * X \tag{3.2.2}$$

According to $(3.1.3)$ and $(3.1.6)$, after arbitrage the maximum K value increasement of two sub-pools are:

$$\begin{aligned} \Delta K_A^M = \Delta K_B^M &= \frac{((X + x) * Y - X * (Y - y))^2}{4 * (X + (X + x)) * ((Y - y) + Y)} \\ &= \frac{(x * Y + y * X)^2}{4 * (2X + x) * (2Y - y)} \\ &= \frac{(\sqrt{\gamma} - 1)^2}{4\sqrt{\gamma}} * X * Y \end{aligned} \tag{3.2.3}$$

After arbitrage, the increase ratio of K value both in A pool and B pool will be:

$$\delta K_A = \delta K_B = \frac{(\sqrt{\gamma} - 1)^2}{4\sqrt{\gamma}} \tag{3.2.4}$$

According to $(2.4.19)$, the amount of tokens exchanged between the two sub-pools during arbitrage operation can be calculated as follows:

$$L_A^e = \frac{x}{2} \tag{3.2.5}$$

$$L_B^e = \frac{x}{2(X + x)} * Y = \frac{y}{2} \tag{3.2.6}$$

It can be seen that after arbitrage, the number of tokens in the two sub-pools becomes:

$$(X + \frac{x}{2}\ ,\ Y - \frac{y}{2})\ ||\ (X + \frac{x}{2}\ ,\ Y - \frac{y}{2}) \tag{3.2.7}$$

That is, the number of tokens in the two sub-pools is equal, achieving a complete balance.

The above analysis is based on the assumption that the amount of tokens in A pool and the B pool are exactly the same. Let's analyze below, if the amount of tokens in A pool and the B pool are different, how much is the K value increase of the two sub-pools after arbitrage according to FreeSwap protocol.

In this case, the changes in the amount of tokens during the trade and arbitrage process are shown in the following table:

$Sub-Pools$:	$A\ Pool$	$B\ Pool$
$Token\ State:$	$(N_{AA}\ ,\ N_B)$	$(N_A\ ,\ N_{BB})$
$Initial\ State:$	$(X_1\ ,\ Y_1)$	$(X_2\ ,\ Y_2)$
$Post\ Trade:$	$(X_1\ +\ x_1\ ,\ Y_1\ -\ y_1)$	$(X_2\ ,\ Y_2)$
$PostArbitrage$:	$(X_1\ +\ x_1'\ ,\ Y_1\ -\ y_1')$	$(X_2\ +\ x_2'\ ,\ Y_2\ -\ y_2')$

Assuming that the K Value ratio of A pool and B pool is represented by β, the following relationship exists:

$$\begin{cases} \frac{Y_1}{X_1} = \frac{Y_2}{X_2} & (Same\ Price) \\ \frac{X_2 * Y_2}{X_1 * Y_1} = \beta & (K\ Value\ ratio) \\ (X_1 + x_1) * (Y_1 - y_1) = X_1 * Y_1 & (Constant\ Invariant) \\ (X_1 + x_1) * Y_2 = \gamma * X_2 * (Y_1 - y_1) & (Trigger\ Arbitrage) \end{cases} \tag{3.2.8}$$

Jumping the dedcution process, it can be concluded that:

$$\Delta K_A^M = \Delta K_B^M = \frac{(\gamma - 1)^2}{4\gamma\left(\sqrt{\gamma} + \frac{1}{\sqrt{\beta}}\right)\left(\frac{1}{\sqrt{\gamma}} + \frac{1}{\sqrt{\beta}}\right)} * X_1 * Y_1 \tag{3.2.9}$$

So after arbitrage, the K value increase ratio of A pool and B pool are:

$$\delta K_A = \frac{(\gamma - 1)^2}{4\gamma\left(\sqrt{\gamma} + \frac{1}{\sqrt{\beta}}\right)\left(\frac{1}{\sqrt{\gamma}} + \frac{1}{\sqrt{\beta}}\right)} \tag{3.2.10}$$

$$\delta K_B = \frac{(\sqrt{\gamma} - 1)^2}{4\sqrt{\gamma}(\sqrt{\beta\gamma} + 1)(\sqrt{\beta} + \sqrt{\gamma})} \tag{3.2.11}$$

It can be seen from $(3.2.10)$ and $(3.2.11)$ that δK_A increases monotonically with β, while δK_B decreases monotonically with β, which means that when the amount of funds in the two sub-pools is unbalanced, the more the amount of tokens in the sub-pool relative to the other one, the less the increase in K value during arbitrage. On the contrary, the less the amount of tokens in the sub-pool, the more the increase in K value during arbitrage. Therefore, when users add liquidity, it is more advantageous to choose to join the sub-pool with smaller amount of tokens, and on the round it helps improve the liquidity of this smaller pool. This internal regulation mechanism of FreeSwap protocol can make the amount of tokens in the two sub-pools to keep in a dynamic balance.

3.3 Equivalent transaction fee rate

If it is not through trading arbitrage, but by charging exchange fees, how to set the exchange fee rate to obtain the same K value increase as $(3.2.4)$?

Let's represen this trade fee rate as α. Since the FreeSwap protocol has two independent one-way sub-pools, considering the equivalence, when calculating α, the single two-way trade pool should have the same token amount, which can be expressed as: $(2X, 2Y)$. The user exchanges $TokenA$ with the

same amount of x as $(3.2.2)$ for some amount of $TokenB$, denoted by y'. According to the "constant-product invariant" formula, there are:

$$(2X + (1-\alpha)x) * (2Y - y') = 2X * 2Y \tag{3.3.1}$$

After the completion of the exchange, because the exchange fees do not participate in "constant-product invariant" calculation, just directly enter the liquidity pool, resulting in an increase in the K value of the overall token pool:

$$\begin{aligned} \Delta K &= \alpha x * (2Y - y') \\ &= ax * \frac{2X * 2Y}{2X + (1-\alpha)x} \end{aligned} \tag{3.3.2}$$

Compared with the K value before the exchange, the K Value increase of the whole pool is as follows:

$$\delta K = \frac{ax}{2X + (1-\alpha)x} \tag{3.3.3}$$

Combining with $(3.2.2)$, $(3.2.4)$, let $\delta K = \delta K_A$, we have:

$$\alpha = \frac{\sqrt{\gamma} - 1}{\sqrt{\gamma} + 1} \tag{3.3.4}$$

As the calculation example, when γ = 1.01, $\alpha \approx$ 2.488‰, that is, if the arbitrage is automatically conducted while the price deviation of the two sub-pools reaches 1%, the profit of the liquidity pool provider is equivalent to collect exchange fee of 2.488‰ from trading users. Currently, UniSwap [2],[3] charges a exchange fee of 3‰ from trading users, so FreeSwap can achieve approximately 83% of UniSwap's profit rate through only exchange arbitrage. Taking into account that the FreeSwap protocol can provide users with totally free exchange, it can attract more exchange users and increase exchange volume. By the increase in exchange volume, it is possible for FreeSwap to achieve the same profit level as UniSwap, or even exceed UniSwap while completely waiving any exchange fees.

It can also be deduced from $(3.3.4)$

$$\sqrt{\gamma} = \frac{1+\alpha}{1-\alpha} \tag{3.3.5}$$

According to the calculation, $\gamma \approx$ 1.0121‰ while α = 3‰, that is, if FreeSwap performs arbitrage when the price deviation of the two sub-pools reaches 1.21%, it can obtain equivalent 3‰ of exchange fee income same as UniSwap.

3.4 Not applicable exchanges

Since Freeswap protocol relies on exchange arbitrage when the price of sub-pools deviate to a certain level, Freeswap is not suitable for liquidity pools with both stable tokens exchanged. The exchange of stable tokens needs to minimize the price deviation as much as possible, it will not help attract exchange users by accumulating price deviation and then arbitraging.

Freeswap is also not applicable to exchange deflationary tokens. As intenal exchange will occur between

two sub-pools for arbitrage, and the transfer of deflationary token will cause token deflation, which is not beneficial to the sub-pool swapping in deflationary token. Moreover, arbitrage will increase the transfer times of deflationary token, finally accelerate token deflation, this way violate the original intention of deflationary token.

4. FreeSwap protocol conclusion

Based on the "constant-product invariant" formula, freeswap creates two one-way exchange sub-pools for each pair of tokens. One-way exchange will cause the token prices of two sub-pools to slide in opposite direction. When the the price slippage reaches the preset threshold, the two-pools will exchange with each other internally and automatically for arbitrage. This arbitrage could provide revenue for the liquidity providers, and also smooth the price devation of two sub-pools.

The arbitrage mechanism of the FreeSwap protocol can maximize the profit for the two sub-pools at the same time, which ensures fairness and achieves a win-win situation. Calculations show that if arbitrage is performed when the price deviation of the sub-pool reaches 1%, FreeSwap's arbitrage mechanism can achieve the income equivalent to charge 2.488 ‰ of exchange fees while charging no fees at all.

By implementing a completely free decentralized exchange protocol，it is expected that FreeSwap can help attract more and more users to join in decentralized exchange and take part in more and more Defi activities.

References

1、Vitalik Buterin, **Let's run on-chain decentralized exchanges the way we run prediction markets**, Dec. 2016. URL: https://www.reddit.com/r/ethereum/comments/55m04x/lets_run_onchain_decentralized_exchanges_the_way.

2、 Hayden Adams, **Uniswap Whitepaper**. URL: https://hackmd.io/C-DvwDSfSxuh-Gd4WKE_ig.

3、 Hayden Adams et al, **Uniswap V2 Core**. URL: https://uniswap.org/whitepaper.pdf.

4、 Vitalik Buterin, **The x*y=k market maker model**. URL: https://ethresear.ch/t/improving-front-running-resistance-of-x-y-k-market-makers.

5、Pintail, **Understanding Uniswap Returns**. URL: https://medium.com/@pintail/understanding-uniswap-returns-cc593f3499ef.

6、Pintail, **Uniswap: A Good Deal For Liquidity Providers?**. URL: https://medium.com/@pintail/uniswap-a-good-deal-for-liquidity-providers-104c0b6816f2.