Bounded Loss of Uniswap and Senswap Models

October 31, 2020

1 Uniswap Bounded Loss

Uniswap is an automated market maker based on *constant function market maker*. In this model, the cost function is described as follows:

$$C(x) = x_1 * x_2 = k, \text{where k is a constant.}$$

For this market, the price function for each token is defined as:

$$P_1(x) = \frac{\partial C(x)}{\partial x_1} = x_2.$$

$P_2(x)$ is defined in similar way. Hence, if x_1 grow up as fast then the price of x_2 as small.

No-arbitrage conditions: The marginal price of coin α in Uniswap is defined as the price of an infinitesimally small trade. In other words, the marginal price is the exchange rate at the trading time. This price is described by the following formula:

$$\frac{d\Delta_2}{d\Delta_1} = \frac{1}{\gamma}\frac{x_2}{x_1} = \gamma^{-1}m_u,$$

where $m_u = x_2/x_1$. From the above formula, we see that $d\Delta_2 = \gamma^{-1}m_u d\Delta_1$. Thus, we can always make a nonzeoro profit if the Uniswap marginal price of α, $\gamma^{-1}m_u$ is smaller than the market marginal price m_p. In other words, the following inequality holds: $m_p\Delta_1 \geq \gamma^{-1}m_u\Delta_1$ after performing a small enough trade. Assume there is no arbitrage, this means we must have:

$$m_u \geq \gamma m_p.$$

Similarly, by swapping α for β we get the following bounds on the Uniswap market price:

$$\gamma m_p \leq m_u \leq \gamma^{-1}m_p.$$

In the case of no transaction fee, this means $\gamma = 1$, to obtain the no arbitrage conditions, we must have $m_p = m_u$.

1.1 Bounded Loss Condition

Bounded loss is the condition that makes a market maker can control the loss in the worst-case. It is normally defined as:

$$\sup[\max_i(x) - C(x)] < \infty$$

The worse-case loss of a cost function C which begins from initial payout vector x^0 is

$$\sup[\max_i(x_i) - C(x) + C(x^0)]$$

In the Uniswap model, the market maker can't be suffered loss from trading exclude the possible loss from the difference of exchange rate between two tradings at two different times. Therefore, if we want to define a bounded loss for Uniswap, we can regard the payout which pays for traders of the market maker as the number of tokens that are sent for traders after each trading.

With the above-defining payout along with the Uniswap formula, we noticed that the maximum payout that the market maker has to pay for traders in the worst case is to equal with either all reserve of x_1 or x_2. Hence, using the definition of bounded loss condition, we have:

$$\sup[\max_i(x_i) - C(x)]$$

either equal to

$$x_1 - C(x) = x_1 - k < \infty, \text{ with } k \text{ is the constant}$$

or

$$x_2 - C(x) = x_2 - k < \infty, \text{ with } k \text{ is the constant}$$

and the worse-case loss is to equal to either x_1 or x_2.

2 Senswap Bounded Loss

It's similar to Uniswap, our model Senswap is also based on constant functions class. Concretely, we chose the constant ellipse function for our model rather than the constant product function for its scalability and highly adapted options. The constant-ellipse cost function is defined as follows:

$$C(x) = (x_1 - a)^2 + (x_2 - a)^2 + bx_1x_2 = k.$$

Thus, the price formula for each token has the following form:

$$P_1(x) = \frac{\partial C(x)}{\partial x_1} = 2(x_1 - a) + bx_2.$$

Similarly, $P_2(x) = 2(x_2 - a) + bx_1$. The rate exchange between two coins is

$$P_{12} = \frac{2(x_2 - a) + bx_1}{2(x_1 - a) + bx_2}$$

2.1 Liquidity

We will survey the liquidity of this model. Assume that in the initial state, the reserves for coins A and B respectively are x_1 and x_2. Then, the rate exchange between A and B might be

$$P_{12} = \frac{2(x_2 - a) + bx_1}{2(x_1 - a) + bx_2} = k$$

We always expect this rate exchange to be constant number k at all the time. Hence, if there are some trades which happened then the reserves x_1^t, x_2^t should satisfy the following equation:

$$P_{12}^t = \frac{2(x_2^t - a) + b^t x_1^t}{2(x_1^t - a) + b^t x_2^t} = k$$

By transforming, we have

$$a = \frac{x_2(2k - b^t) + x_1(b^t k - 2)}{2k - 2}$$

When the rate exchange reaches to 1, the value of a can not be defined. Therefore, in some cases, we should fix the value of a and adjust the value of b as desired. But in reality, the same price between two coins is barely occurrence. Thus in this model, we just consider the stable of liquidity based on the change of variable a.

2.2 Bounded Loss and Impermanent Loss

Since Senswap is constructed from constant-ellipse function, it's sure that the Bounded Loss Condition always exists as the same as the Uniswap model. And the remaining part, we should evaluate the impermanent loss of this model at some times.