

Arbitrage Model for AMM Liquidity Pool structure.

Due to the simplicity of the AMM pricing model, cyclic arbitrage opportunities can be derived from the following model:

Given an arbitrage route of n – Tokens $\{T_1, T_2, T_3, \dots, T_n\}$

Where $T_n = T_1$ for route to be cyclic

And let Lp_n be the liquidity pool for Pair T_n / T_{n-1} (or vice versa)

And let Ta_n be Token A for Lp_n & Tb_n be Token B for Lp_n

And let Ra_n be the reserves of Ta_n for Lp_n & Rb_n be the reserves of Tb_n for Lp_n

And let μ_n be a "reverse" flag where:

$$\mu_n = 0 \text{ if } Ta_n = T_{n-1}$$

$$\mu_n = 1 \text{ if } Ta_n = T_n$$

And let γ_n to be the fee associated with Lp_n and the product of fees to be:

$$\omega(k) = \prod_{i=1}^k \gamma_i$$

The output Δn of the route, given input Δa considering all the fees of varying CEXs involved in the pool, will be:

$$\Delta n = \frac{Eb_n \omega(n) \Delta a}{Ea_n + (\Delta a \omega(n-1))} \quad (\text{eq. 5})$$

$$\text{Where } Ea_n = \frac{Rj_n Ea_{n-1}}{Rj_n + (Eb_{n-1} \omega(n-1))} \text{ for } n > 1 \text{ and } Ea_n = Rj_n \text{ for } n = 1 \quad (\text{eq. 6})$$

$$\text{And } Eb_n = \frac{Rk_n Eb_{n-1}}{Rj_n + (Eb_{n-1} \omega(n-1))} \text{ for } n > 1 \text{ and } Eb_n = Rk_n \text{ for } n = 1 \quad (\text{eq. 7})$$

$$\text{Where } Rj_n = \begin{cases} Ra_n & \text{when } \mu_n = 0 \\ Rb_n & \text{when } \mu_n = 1 \end{cases} \quad (\text{eq. 8})$$

$$\text{And } Rk_n = \begin{cases} Rb_n & \text{when } \mu_n = 0 \\ Ra_n & \text{when } \mu_n = 1 \end{cases} \quad (\text{eq. 9})$$

The Optimal Input for the Arbitrage Route can be determined when the derivative of the return function $f(n)$, where $f(n) = \Delta n - \Delta a$, equals to 0

$$\frac{df}{da} = \frac{Ea_n * Eb_n * \omega(n)}{(Ea_n + (\Delta a * \omega(n-1)))^2} - 1 = 0$$

$$\text{Hence optimal input amount } \Delta a_{\text{optimal}} = \frac{\sqrt{Ea_n Eb_n \omega(n)} - Ea_n}{\omega(n-1)} \quad (\text{eq. 10})$$

Therefore maximum ouptut amount, without knowing the Optimal input amount:

$$\Delta n_{maximum} = \frac{Eb_n \omega(n)(\sqrt{Ea_n Eb_n \omega(n)} - Ea_n)}{\omega(n-1)\sqrt{Ea_n Eb_n \omega(n)}} \quad (eq. 11)$$

Hence, difference between Optimal Input & Maximum Ouptut amount (λ), without knowing the Optimal input amount :

$$\lambda = \frac{\Delta n_{maximum}}{\Delta a_{optimal}} = \frac{Eb_n \omega(n)(\sqrt{Ea_n Eb_n \omega(n)} - Ea_n)}{\omega(n-1)\sqrt{Ea_n Eb_n \omega(n)}} * \frac{\omega(n-1)}{\sqrt{Ea_n Eb_n \omega(n)} - Ea_n}$$

$$\lambda = \frac{\Delta n_{maximum}}{\Delta a_{optimal}} = \frac{Eb_n \omega(n)}{\sqrt{Ea_n Eb_n \omega(n)}} = \sqrt{\frac{Eb_n \omega(n)}{Ea_n}} \quad (eq. 12)$$

Knowing the Optimal Input Amount we can then simply calculate the maximum output amount:

$$\Delta n_{maximum} = \Delta a_{optimal} * \lambda = \Delta a_{optimal} * \sqrt{\frac{Eb_n \omega(n)}{Ea_n}} \quad (eq. 13)$$

Finally, we can intrepret the unit difference between input amount and output amount as :

$$\beth = \Delta n_{maximum} - \Delta a_{optimal} = \Delta a_{optimal} * (\lambda - 1)$$