对比 Uniswap，一种新的去中心化交易所的流动性算法

BY he.d.d.shan@hotmail.com

注意：

1. 本文的结论，有些是测试过的，有些是理论推导的。
2. 本文的读者最好了解 Uniswap 的恒定乘积理论。

Uniswap 的算法

这里简单回顾一下 Uniswap 的流动性恒等式算法。为了简化，以 UniswapV1V2 来说明。

对于交易对 TokenA=>TokenB，对应的流动资金是 X 和 Y，则 **X*Y=K**，当交易的时候，X 或 Y 会发生变化，K 值不变。

假设当初的流动性是 X0 和 Y0，用户支付 dY 数量的 TokenB 购买 x 数量的 TokenA，则需要满足流动性性公式：X0 * Y0 =（X0 - x）* (Y0 + dY)，可以推导出

x = X0 -(X0 * Y0)/ (Y0 + dY)- X0=(X0* dY)/ (Y0 + dY)

即（公式 1） **x =(X0* dY)/ (Y0 + dY)**

何氏新算法（连续价格理论）

核心逻辑是：不使用 x*y=k 来推导交易数量，而是使用价格来推导。

在购买后，价格有两种表现形式，购买的价格是： **bPrice = x / dY**；而剩

余流动性资金的价格是：**lPrice= （X0-x） / (Y0+dy)**。当然这两个价格是一样的（这就是核心点），则有 x / dY = （X0-x） / (Y0+dy)，可以推导出

x * （Y0+dy) = （X0-x） * dY

x * （Y0+2 * dy) = X0 * dY

即（公式 2） **x =(X0* dY)/ (Y0 + 2 * dY)**

这么个算法，有人考虑到了吗？我在中文社区没看见有人提出这个算法。其他出名的 DEX 也没采用这个算法，例如 PancakeSwap、dXdY 等。

何氏算法（连续价格理论）扩展一下

如果存在一个交易池，里面有 TokenA，TokenB，TokenC，是不是可以交易这三个 Token 中的任意交易对？那是肯定的。

这种情况就存在一个问题，当交易 TokenA 和 TokenB 的时候，另外两个交易对 TokenA=>TokenC, TokenB=>TokenC，根本就没有发生交易，但价格变化了（一个升高，一个降低），因为对应池子中的 TokenA 和 TokenB 变化了。那这种情况是合理呢，还是不合理?

合不合理，又要对照 Uniswap 来说：

在 Uniswap 中，存在交易对 TokenA=>TokenB, TokenB=>TokenC, TokenC=>TokenA，如果交易对 TokenC=>TokenA 发生了交易，则在交易对闭环（TokenB=>TokenC, TokenC=>TokenA，TokenA=>TokenB）中存在套利机会，这里我不展开说，做过套利的朋友一眼就明白，对此不太明白的朋友可以查询一下相关资料。套利的结果，就是通过路径 TokenA=>TokenB=>TokenC（三个交易对）得到的 TokenA 和 TokenC 的比价，和交易对

TokenA=>TokenC（这一个交易对）的比价，是一样的，最后导致 TokenB 的价格相对 TokenA 和 TokenB 都变化了（一个上涨，一个下跌)。这里忽略了手续费，但不影响逻辑。

所以说，**使用新算法的交易池，交易 TokenA 和 TokenC，导致 TokenB 相对 TokenA 和 TokenB 的价格变化是合理的。**

那么，四个 Token 组成的交易池，能否行得通呢？答案是肯定的，五个六个也可以的。

另外，相对于 Uniswap，其流动性值的定义也有改变，在本算法中更类似于 UniswapV1V2 的定义，是一种权重。

何氏算法（连续价格理论）的一些意义

1，可以规避内部套利

新算法规避了内部的套利。这个在上面说过了。

而 Uniswap 的内部套利很严重，造成大量的套利交易，增加了 Gas 费用，我个人觉得对以太坊生态有一定的负面影响。

极端的看，如果以太坊只有一个使用此算法的去中心化交易所，这个交易所的交易池囊括了所有的 Token，例如 200 个 Token，那以太坊上就不存在套利机会了（当然跨链的套利机会是另外一回事)。

2，提高流动性资金利用率

在多 Token 交易池，一个 Token 对应多个 Token 交易对，那这个 Token 的

资金利用率相对 Uniswap 在理论上就提高了。

3，降低流动性提供者的无常损失

相对于 Uniswap 的公式 1，本算法的公式 2，加快了价格的下降幅度，相当于减少了无常损失，增加了流动性提供者的利润。

当然，这个属于理论上增加利润的一个因素。实际上的利润和很多因素有关，这里不展开说。

我的计划

1. 找个时间，把这个算法提交给 Uniswap，看看他们怎么说。如果能被 Uniswap 采用最好。
2. 如果有必要，我就写一套代码出来，包括合约和客户端，来测试和演示这套想法。我的合约代码可能做不到 Uniswap 的那种完美，但能够保证安全和计算正确，能做到商业化水平。

算法的可能改进

1， 单独 Token 添加流动性。如果池子里面有 30 种 Token，难道用户要一次性添加 30 种 Token 吗？这肯定不合常理。应该可以添加任意种 Token 就可以了，包括只添加一种 Token。

2， 流动性的计算，主要是价格的计算，需要使流动性对应的 Token 数量“虚化”、“样本化”或“分片化”，类似 UniswapV3 的做法，强调某个区间上的流动性分布，这样就能分离各个 Token 的实际数量和价格的对应关系了，使单个 Token 的添加流动性成为可能。

3， 各个交易对的手续费统一最简单。但是不统一呢？可以设置一个手续费等级，一等、二等。一等和一等的 Token 交易手续费是 3；一等和二等、二等和二等的是 5。方法很多，我这里只是列举这种方法。

给 Uniswap 的留言（2023 年 10 月 19 日，Discord）

Hello，我找到一种新算法，暂且叫做连续价格理论。相对于恒定乘积算法（X*Y=K）,本算法注重价格的连续性，也就是说，在交易价格和交易后的静态价格，是相同的；而 Uniswap 的算法无法满足这点。推导过程如下：

对于 TokenA 和 TokenB 的交易，使用 dy 数量 TokenA 购买 TokenB，假设购买到的 TokenB 的数量位 x，则在购买后，价格有两种表现形式，购买的价格是：**bPrice = x / dY**；而剩余流动性资金的价格是：**lPrice= (X0-x) / (Y0+dy)**。当然这两个价格是一样的（这就是核心点），则有 x / dY = （X0-x）/ (Y0+dy)，可以推导出

x * (Y0+dy) = (X0-x) * dY

x * (Y0+2 * dy) = X0 * dY

即（公式 2） **x =(X0* dY)/ (Y0 + 2 * dY)，**

而采用 Uniswap 的恒定乘积公式推导出的公式是：

即（公式 1） **x =(X0* dY)/ (Y0 + dY)**

公式差别不大，但是理论基础不一样，应用差别较大。使用本人的算法（公式 2）可以把多个 Token 装在一个池子里面，形成多个配对，例如把 TokenA，TokenB，TokenC 都装在一个池子里面，那就会形成 TokenA=>TokenB，TokenB=>TokenC，TokenC=>TokenA 三个交易对，这是公式 1 无法做到的。

我的网址：https://learnblockchain.cn/article/6709

希望和 Uniswap 的朋友探讨一下。

Google 译文如下（改进版，和原版有点小出入，把一两个错误地方修改了）：

Hello,

I found a new algorithm, let's call it continuous price theory. Compared with the constant product algorithm (X*Y=K), this algorithm focuses on price continuity, that is to say, the transaction price and the static price after the transaction are the same; Uniswap's algorithm

cannot meet this point. The derivation process is as follows:

For the transaction of TokenA and TokenB, use the dy quantity TokenA to purchase TokenB. Assume that the quantity of TokenB purchased is x. After the purchase, the price has two forms of expression. The purchase price is: bPrice = x / dY; and the remaining flow is The price of sexual capital is: lPrice=(X0-x)/(Y0+dy). Of course, the two prices are the same (this is the core point), then x / dY = (X0-x) / (Y0+dy), which can be derived

x * (Y0+dy) = (X0-x) * dY

x * (Y0+2 * dy) = X0 * dY

That is (Formula 2) x =(X0* dY)/ (Y0 + 2 * dY),

The formula derived using Uniswap' s constant product formula is:

That is (Formula 1) x =(X0* dY)/ (Y0 + dY)

The formulas are not much different, but the theoretical basis is different and the application is quite different. Using my algorithm (Formula 2), multiple Tokens can be placed in a pool to form multiple pairs. For example, if TokenA, TokenB, and TokenC are all placed in a pool, TokenA=>TokenB, TokenB=> TokenC, TokenC=>TokenA three trading pairs, this is something that Formula 1 cannot do.

My website: https://learnblockchain.cn/article/6709

I hope to discuss it with Uniswap friends.