**项目进展报告**

# 1. 项目回顾

数字货币种类繁多、兑换汇率随货币当前存量和交易兑换量而变化，选择不同兑换路径（包括中间币种和汇兑渠道），并将货币拆分成若干份分别兑换（根据AMM机制决定，不考虑手续费情况下，拆单更划算），都会对最终收益产生影响。因此，在多币种兑换过程中存在套利机会。具体而言，以一定数量的货币A兑换成货币B，可经不同的渠道（交易所/合约）兑换不同数量的中间货币；采用优化算法可得到一条当前情况下的最优兑换路径，使最终得到的货币B数量最多。为此，**以兑换货币量最大为目标，考虑汇兑路径特征和可行性、手续费等，针对不同问题规模和效率要求，提出一系列精确和近似求解方法**。

我们需要考虑*N*种货币，*K*个渠道（交易所/合约），需要决策从初始货币*o*到最终货币*d*，经过哪些中间币种，以何种渠道进行兑换。

相关假设如下：

（1）每步兑换过程中相关渠道下所有币种的存量，**不受除当前操作以外的交易行为影响**（即兑换过程瞬间完成）——即考虑静态问题。

（2）每个渠道中都存在某些货币对的组合，货币对中所含的两种货币可相互兑换，其汇率由该货币对中两种货币的存量和兑换货币量决定。

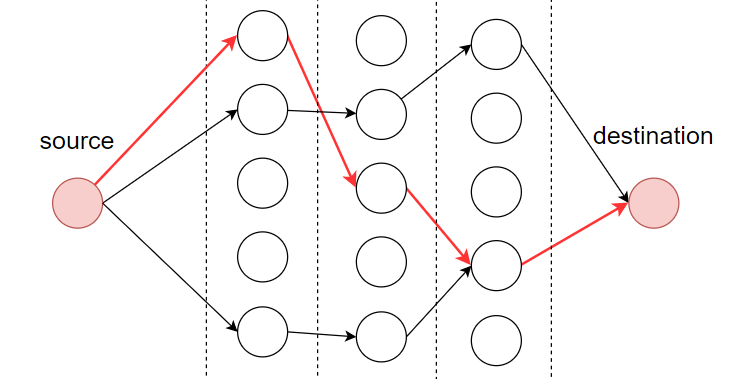
（3）对于某个渠道中某个货币对组合，允许进行拆单，即将一次大数额兑换拆分为多次小数额兑换，分多次完成。

我们建立**两个数学模型**，并提出了如何进行**模型简化**和**算法设计**。

# 2. 数学模型（一）

## 2.1 基本定义

定义**“阶段”**的含义为以源币为起点，交易直到得到目标货币，其最长路径的交易次数。如下图即为一个四阶段的交易场景，其中最长路径用红色加粗线条标出（同一行表示同种货币）。在本模型中，我们限制交易的阶段数S。该模型要求在最后一阶段，所有货币必须转换为目标货币。



(1) 变量

|  |  |
| --- | --- |
| 符号 | 含 义 |
|  | 在阶段*s*，为经渠道*k*兑换成货币*j*，使用货币*i*的数量，非负连续变量。时表示在该阶段未交易的货币量。 |
| *Zij* | 是否存在从币种*i*到币种*j*的流量，0-1变量。 |
|  | 是否在阶段*s*从渠道*k*将货币*i*兑换成货币*j*，0-1变量 |
|  | 货币*i*的访问序号，非负连续变量 |
| *G* | 总的手续费 |

(2) 参数

|  |  |
| --- | --- |
| 符号 | 含义 |
|  | 交易的阶段数，正整数 |
|  | 渠道*k*中 (i,j)对中货币*i*的存量，非负数 |
|  | 待兑换的货币*o*的总量，非负数 |
| *G*1 | 单次交易的固定手续费（实为$15，需提前转换成目标货币） |
| *G2* | 单位数量的手续费比例（2/1000） |
| *Rij* | 币种*i*到币种*j*的兑换比例，用于统一手续费的度量 |
| 𝛼 | 目标函数中目标币种数量的系数 |
| 𝛽 | 目标函数中手续费的系数 |

## 2.2 目标和约束

最大化最后一个阶段从各个渠道的所有币种兑换成币种*d*的量与手续费的加权和，即为目标：



将初始货币和最终货币分别看作网络图的起点和终点，则可以得到对于网络中各个节点的“流限制”约束，以表示各个节点的性质和节点之间的关系。

**（1）起始和结束阶段要求**

**起始阶段**需将持有的所有货币*o*兑换出去，其他货币转出值为0。在**结束阶段**，除了目标货币，其他货币转入值为0。







**（2）中间阶段要求**

对于任意中间**阶段的币种**，兑换得到的量一定等于从此种货币兑换出的量，也即流平衡约束。同时，同种货币相互兑换表明该部分货币在该阶段未进行交易，所以数量保持不变。



**（3）子环路约束**

为避免最优解中出现子环，添加子环路消除约束。







**（4） 流量上限约束**

从某一币种换出的货币量，不能超过所有渠道该币种总量。



**(5) 手续费计算**

注意，手续费包含gas部分和交易费两部分。







## 2.3 模型汇总

























# 3. 简化模型

在第2章的基础上，可以增加其他方面的限制，以缩小可行域。

1. 限制使用的渠道总数，即限制最优解中使用的渠道数量不能超过若干条。
2. 限制所用的币种总数，即限制最优解中使用的币种数量不能超过若干种。
3. 使用分段线性函数对原模型中分式部分近似，但这部分效果有待完善。

**3.1 相关定义**

1. 参数

|  |  |
| --- | --- |
| 符号 | 含义 |
| *K* | 使用的最大渠道数 |
| *I* | 所用的最大币种数 |

1. 变量

|  |  |
| --- | --- |
| 符号 | 含义 |
| *Pk* | 是否使用渠道*k*，0-1变量 |
| *Qi* | 是否使用币种*i*，0-1变量 |

**3.2 实现方式**

（1）限制使用的渠道总数







（2）限制使用的币种总数



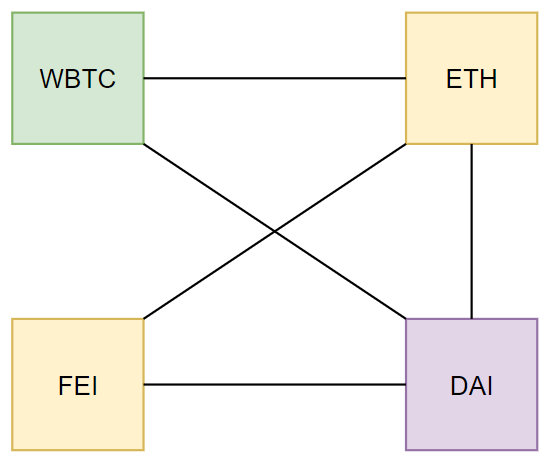




# 4. 数学模型（一）的实际案例

以Uniswap为例，读取各个pair的货币存量；通过1inch网站的最小手续费模式来近似得到各币种之间的折算汇率。在此基础上，进行实际案例测试，具体结果及分析如下。

问题：在uniswap交易所中，有WBTC、ETH、FEI和DAI四种货币，现要将一定量的WBTC兑换成DAI，求最多能兑换多少？



考虑兑换过程中的手续费。

手续费分为两部分，首先每笔交易的固定手续费为15美元（最终要表示为目标货币的数量）；同时每笔交易还产生2/1000交易量的手续费，用前序币种的数量表示（最终要表示为目标货币的数量）。

假设兑换出的目标币种数量与手续费权重相等（均为0.5）。

根据[Uniswap Info](https://info.uniswap.org/home)网站的信息，各个Pair的货币量如下：

('FEI', 'ETH'): (1300000000, 452381),

('DAI', 'ETH'): (76785498, 29104),

('WBTC', 'ETH'): (3111, 64324),

('FEI', 'DAI'): (5531286, 5060441),

('WBTC', 'DAI'): (6.3499, 344547)

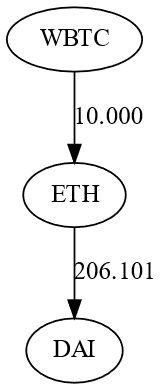
根据[1inch - No.1 Ethereum DeFi aggregator and high efficient AMM](https://app.1inch.io/#/1/swap/ETH/WBTC)网站，可得到币种之间的转换汇率，用于将不同币种表示的手续费表示为目标币种。

('FEI', 'DAI'): 0.91678,  
('ETH', 'DAI'): 2633.94,  
('WBTC', 'DAI'): 54411.1,  
('DAI', 'DAI'): 1  
('USDT', 'DAI'): 0.99806

**4.1 案例1**

首先假设起点货币量为**10**单位，用分阶段模型求解，结果如下：

1. 阶段数=2

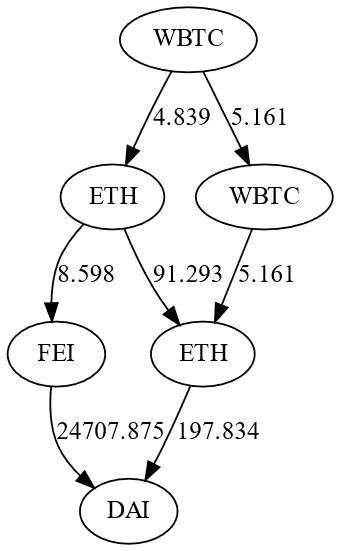


总手续费=2203.877074975969

目标货币数量（不扣手续费）=539934.6148432087

目标函数: 268865.36888411635

1. 阶段数=3

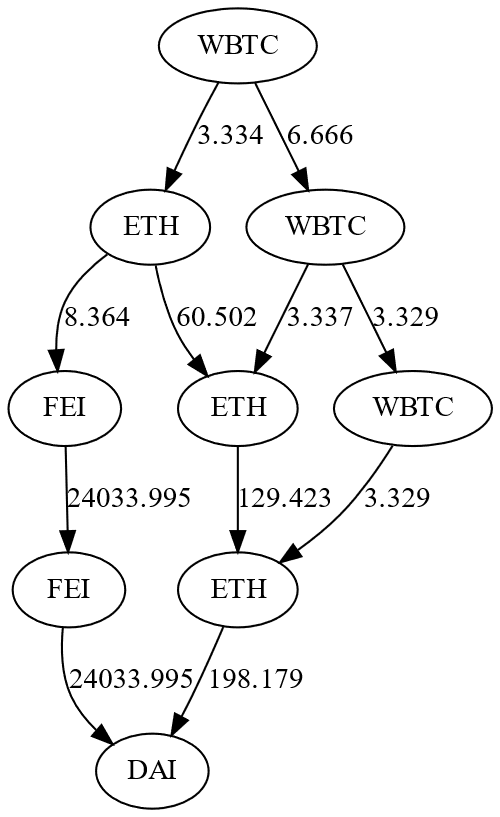


总手续费=2295.8400774523952

目标货币数量（不扣手续费）=540928.6208359122

目标函数: 269316.3903792299

1. 阶段数=4



总手续费=2310.1589610449414

目标货币数量（不扣手续费）=541216.2165952543

目标函数: 269453.02881710464

**和1inch网站结果对比如下**：

目标货币量（去掉2/1000手续费）

|  |  |  |  |
| --- | --- | --- | --- |
| Stage | 2 | 3 | 4 |
| 我们的 | 537760 | 538707 | 538995 |
| 1inch | 545318 | 545318 | 545318 |

Gas费

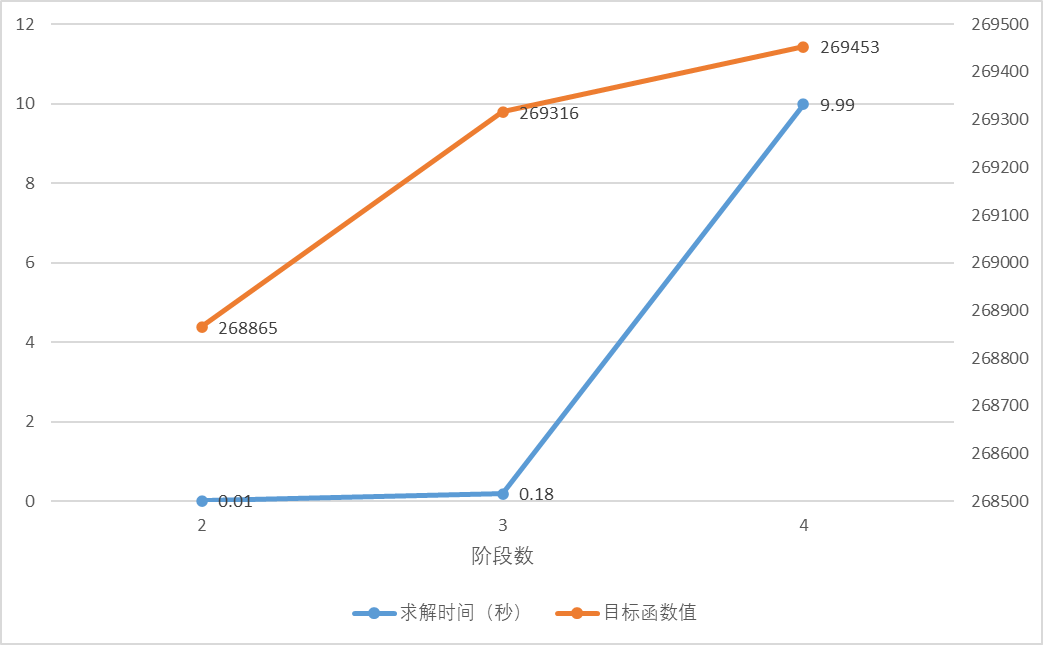
|  |  |  |  |
| --- | --- | --- | --- |
| Stage | 2 | 3 | 4 |
| 我们的 | 29.94 | 74.85 | 89.82 |
| 1inch | 50.65 | 50.65 | 50.65 |

我们兑换出的货币量略低于1inch给出的货币量，原因如下：

* 仅使用了一个交易所
* 仅使用了四种货币
* 数据和1inch后台的有差别

但是需要知道，1inch的结果也只是静态方案的评价，对实际兑换结果的衡量也不完全准确。而且我们和1inch的对比本身就不是完全公平的（数据不同，也无法完全限制他使用哪些货币）。

而后进行阶段数的灵敏度分析。

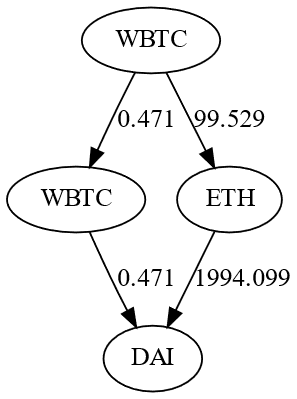


容易看出，随着阶段数增加，求解时间和目标函数值都有所增加，但求解时间增长快，目标函数增长慢，因此需要合理控制阶段数，在求解时间和目标函数之间取得平衡。

**4.2 案例2**

现增加起点货币量至**100**单位，用分阶段模型求解，结果如下：

1. 阶段数=2

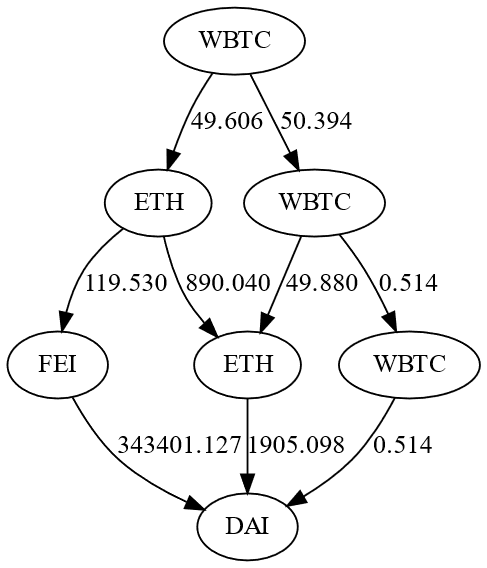


总手续费=21431.80567628811

目标货币数量（不扣手续费）=4947490.585328097

目标函数: 2463029.3898259047

1. 阶段数=3

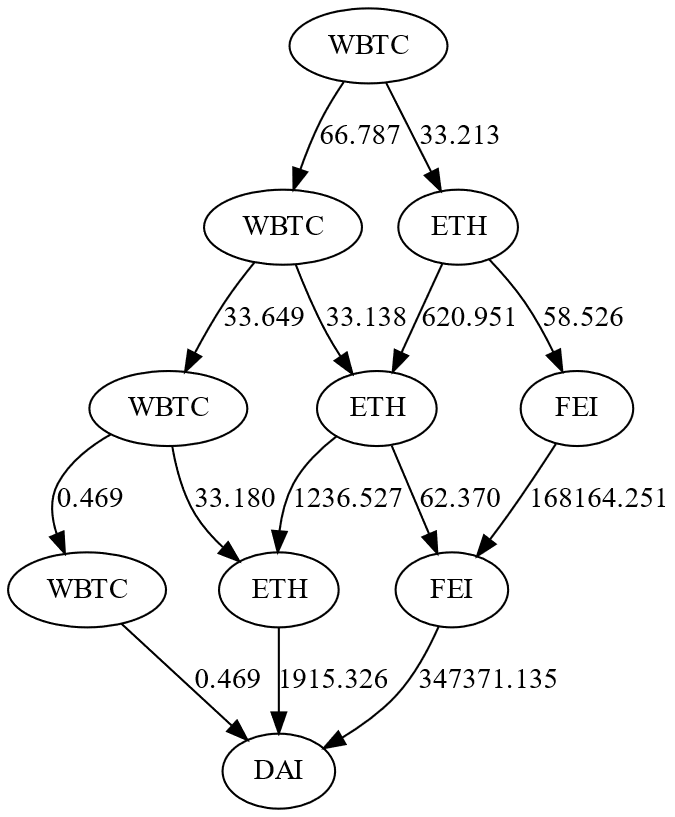


总手续费=22267.18866434745

目标货币数量（不扣手续费）=5039063.67022648

目标函数: 2508398.2407810665

1. 阶段数=4



总手续费=22365.489147212615

目标货币数量（不扣手续费）=5063928.828287168

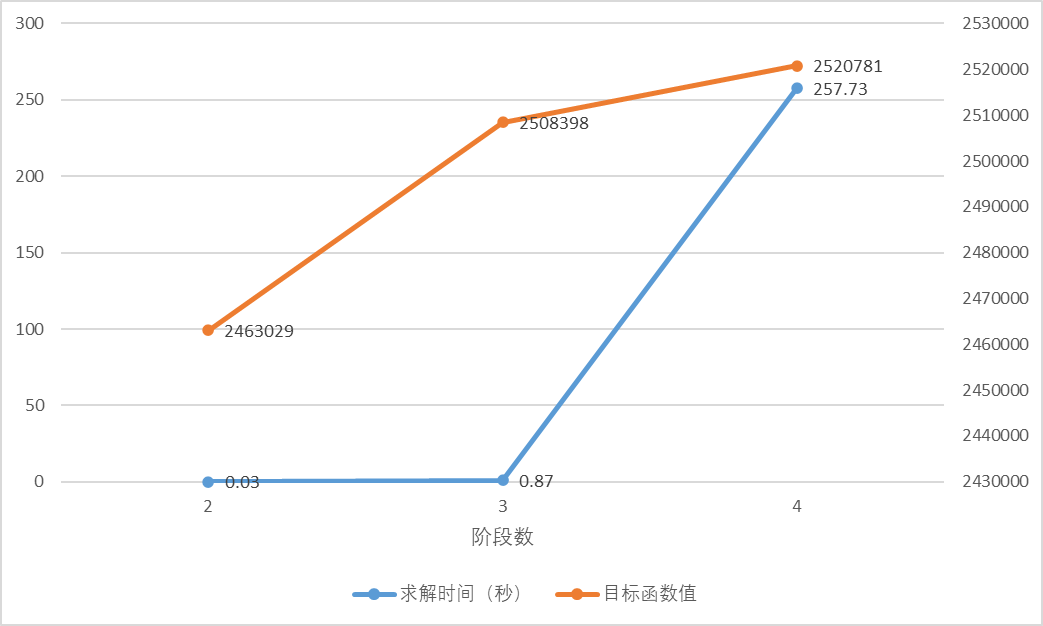
目标函数: 2520781.669569978

目标货币量（去掉手续费）

|  |  |  |  |
| --- | --- | --- | --- |
| Stage | 2 | 3 | 4 |
| 我们的 | 4926103 | 5016886 | 5041683 |
| 1inch | 5239238 | 5239238 | 5239238 |

Gas费

|  |  |  |  |
| --- | --- | --- | --- |
| Stage | 2 | 3 | 4 |
| 我们的 | 44.91 | 89.82 | 119.76 |
| 1inch | 57.65 | 57.65 | 57.65 |



# 5. 数学模型（二）

**5.1 定义**

（1）角标

|  |  |
| --- | --- |
| 符号 | 含义 |
| *i*, *j* | 某种货币，取值属于[1,*N*]之间，整数 |
| *k* | 某个渠道，取值属于[1,*K*]之间，整数 |
| *o* | 初始货币 |
| *d* | 最终货币 |
| *p* | 分拆订单序号，取值属于[1,*P*]之间，整数 |

（2）变量

|  |  |
| --- | --- |
| 符号 | 含义 |
|  | 为经渠道*k*兑换成货币*j*的第*p*单货币*i*的数量，非负连续变量 |
|  | 是否经渠道*k*将货币*i*的第*p*单兑换成货币*j*，0-1变量 |
|  | 是否将货币*i*兑换成货币*j*，0-1变量 |
|  | 货币*i*的访问序号，非负连续变量 |
|  | 兑换过程中产生的手续费总量 |

（3）参数

|  |  |
| --- | --- |
| 符号 | 含义 |
|  | 渠道*k*中货币对中货币*i*的数量 |
|  | 待兑换的货币*o*的总量，非负数 |
| *M* | 建模所需，一个足够大的正数 |
|  | *i*货币对*j*货币的近似汇率 |
|  | 每次兑换的手续费，与兑换次数有关，非负数 |
|  | 每次兑换的手续费，与交易量有关，非负数 |
|  | 目标货币量的权重 |
|  | 手续费在目标函数中的权重 |

**5.2 目标**

由于计算货币兑换量需要汇率的表达式，故此处先对汇率形成机理进行分析。在某个渠道*k*下的两种不同货币*i*和*j*形成货币对，其当前存量分别为**和**，根据Uniswap协议，二者的乘积应保持恒定值。假设使用的货币*i*能兑换出的货币*j*，则有如下关系成立，进一步推导可得：



在此基础上，最大化从各个渠道的所有币种兑换成币种*d*的量并减去兑换过程中产生的所有手续费即为目标函数：



**5.3 约束**

将初始货币和最终货币分别看做网络图的起点和终点，则可以得到对于网络中各个节点的“流限制”约束，以表示各个节点的性质和节点之间的关系。

**（1）起点和终点要求**

起点不能有“流入”，即不存在某个货币兑换成货币*o*。同理，终点不能有“流出”，即一旦换成货币*d*就不能继续兑换了。此外，起点需将持有的所有货币*o*兑换出去。







**（2）中间货币要求**

对于任意中间货币，兑换得到的量一定等于从此种货币继续兑换出的量，也即流平衡约束。同时，不允许存在同种货币的自循环式兑换。





**（3）手续费计算**

手续费由与兑换与否相关和与交易量相关的两部分组成。用以支付手续费的货币种类为当次交易中所得货币的类型，为方便计算，使用近似汇率将所有支付的手续费以目标货币的数量结算。**手续费和交易次数以及交易量有关**，因此需借助相应的0-1变量表达限制。同时，给出和的关系。

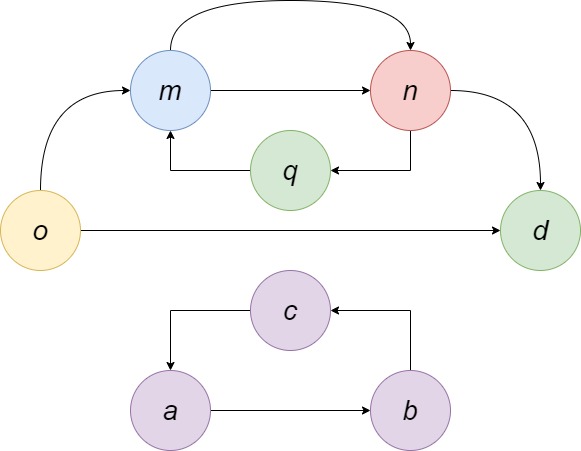






**（4）兑换路径子环路消除**

子环路即独立形成闭环的环路，如下图中的m-n-q-m或a-b-c-a。币种之间的子环路最好在建模过程中去除，以减少求解时候的搜索空间，提升求解效率。



上图中的子环路**一定**对目标函数无正向贡献（如m-n-q-m），**可能**与初始持有货币量无关（如a-b-c-a）。因此，在不超过手续费约束的情况下，将可能不受到限制。但这种兑换操作显然不合理，故应借助0-1变量引入相关约束，去掉此种情况。同时引入和的关系：







注意，如果考虑到汇兑过程中货币之间汇率的时变性，则子环路可能对目标函数产生正向贡献，即使用给定数量的某种货币开始兑换，最终兑换回本币得到的数量多于初始持有量。当然，孤立于正常兑换路径之外的子环路仍然需要避免。

**（5）变量上限约束**

为保证精确求解时候的数值稳定性，拟增加针对部分决策变量的上限约束。对于初始货币，每次向外兑换的量都不超过总持有货币量 。对于其他货币，每次兑换所消耗的货币量应小于所有货币池中该货币在所有货币对中存量的总和。



**5.4 模型汇总**

由此我们得到一个**带有分式项的非线性混合整数规划模型**，如下所示：



数学模型（二）采用了和数学模型（一）不同的方式建模，**都对原问题等价**。两种模型的精度和效率比较后续还将继续进行。

此外，第3章中的简化方法仍然适用。

# 6. 数值优化方法初探

**6.1 理论思路**

对企业来说，购买商业求解器是一笔较大开销，故项目团队也探索了基于开源求解器的数值优化方法。

由于本问题为具有整数变量和双线性约束的非凸优化问题，因此首先应进行松弛处理，而后直接使用基于梯度的方法（如序列二次规划方法，SQP）求得局部最优解。该局部最优解未必可行，后续需要设计邻域搜索机制进行修复，最终得到一个可行解（但不保证是最优解）。

（1）对于整数变量处理

1）线性松弛

2）去掉整数变量所在的约束

（2）对于双线性项处理

直接调用Scipy等开源求解工具，使用SQP算法，有限次迭代后收敛得到局部最优解。为了提升局部最优解的质量：

1）采用Multi-Start方法，从多个初始解出发进行迭代下降，并对结果优中选优。

2）当陷入局部最优时，可给一个较大幅度的随机扰动，使其脱离局部最优点，并继续开始迭代。

（3）修复调整为可行解的方法

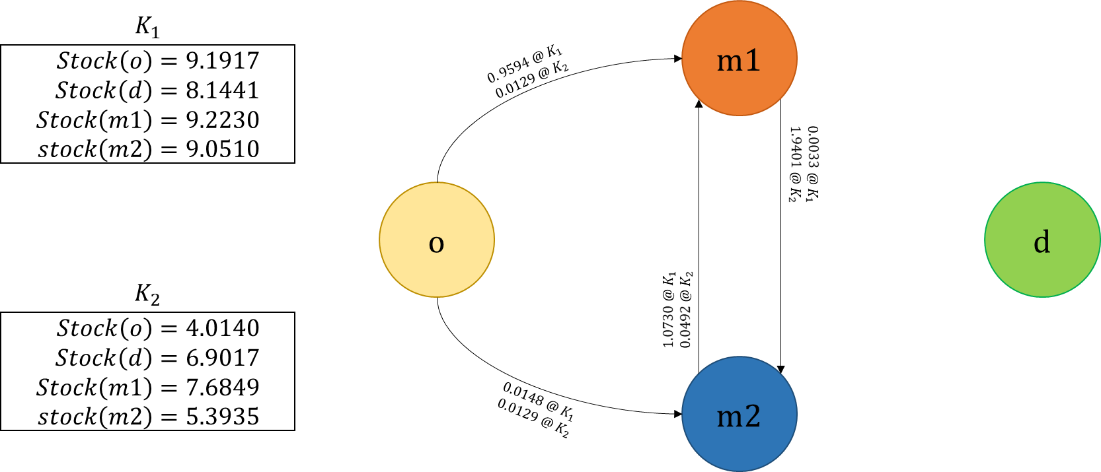
进行路径的局部调整来满足流平衡。例如，环路中去掉一段，该路段的起点终点重新找：起点直接通向d，终点连接o，相应的流量做局部调整。当然，也可以采用其他local search方法。

**6.2 测试案例**

我们使用上一次汇报中的模型（本次汇报的模型还未来得及进行实验）开展实验，即不考虑拆单，也不进行任何其他限制。

考虑两个渠道，四种货币，其中：与的货币存量分别如下：

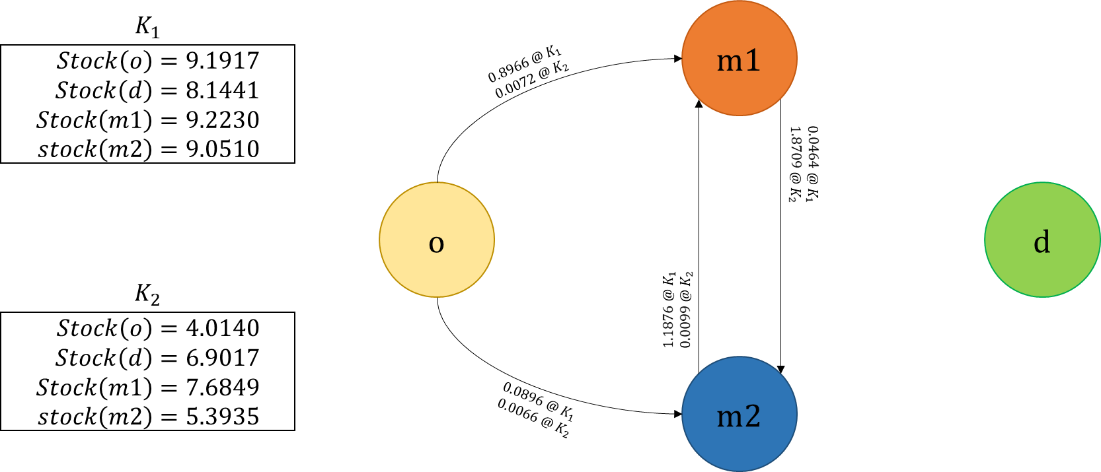
在不考虑消除子环路时，使用SQP算法求解的结果为（仅列出非0变量）：



所得目标值为4e-17单位的货币，即**算法并没有使得兑换的目标货币量最大**。因为子环的存在得出与预期相差较大的局部最优解。

下面加入子环约束条件并对变量作**线性松弛**处理：

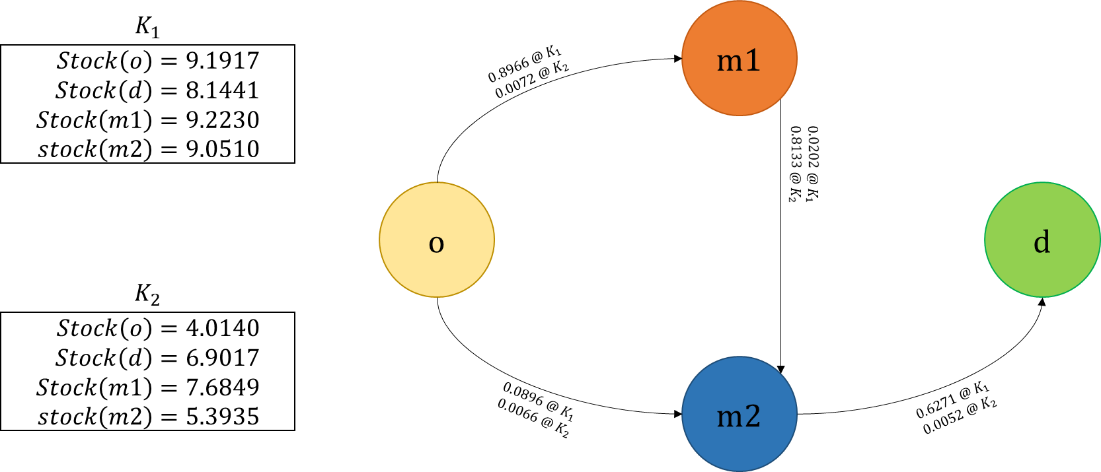
再次使用SQP算法求解，其结果（仅列出非0变量）为：



所得目标值为8e-09单位的货币，**可见线性松弛仍未能有效优化解的质量**。

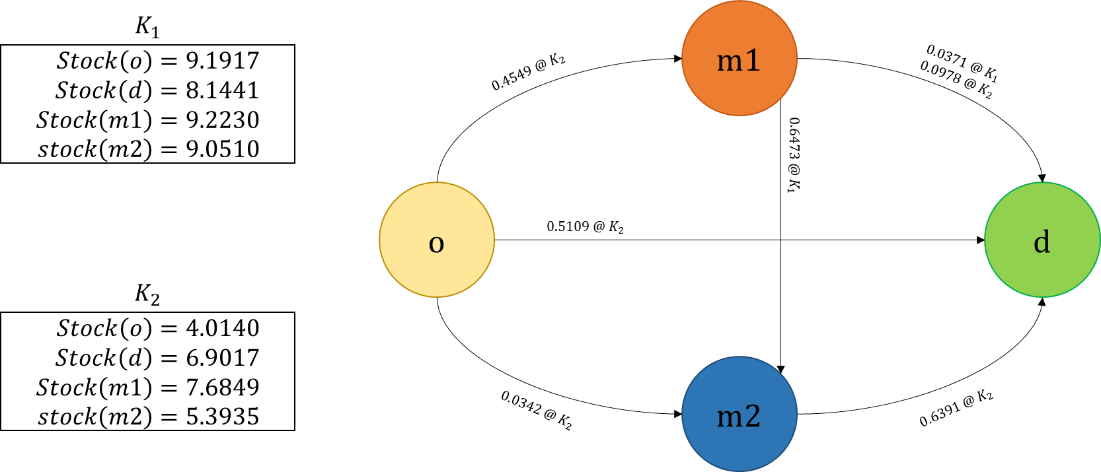
对当线性松弛所得解做**局部搜索**，破除子环。具体来说，将原本流入的直接兑换为目标货币，则调整后的货币流动如下：

此处需要注意的是，因为不再允许货币从流入（红色标识），所关联的流平衡约束也需要重新计算，即蓝色所标识的部分。经计算此时获得总计0.63228单位。根据先前流入时，流经和的比例（和），令新货币流以相同比例流入。



最终所得目标货币共计0.5268单位，**显著低于Gurobi所得的全局最优目标值**1.6299单位。

附：以下为Gurobi所求得的全局最优解：



与做线性松弛和局部搜索后的SQP结果相比，Gurobi额外利用了和两条路径来提高目标值。

这也提示我们，需要**进一步调整邻域搜索方法**，解还有较大提升空间。

需要说明的是，由于数值优化需要自己设计路径搜索算法完成修复工作，不仅效果无法保证而且耗费人工， 故暂时不作进一步推进。

# 7. 后续工作

|  |  |
| --- | --- |
| 时段 | 任务 |
| 2021.05.06-2021.05.16 | 1. 数学模型（一）和（二）之间的比较 2. 使用更多的实盘数据测试，完善之前简单的数值实验 |
| 2021.05.17-2021.05.27 | 各个简化模型和精确模型之间的比较——衡量精度与效率之间的权衡 |
| 2021.05.28-2021.05.31 | 阶段性总结汇报，探讨下一步方向 |