定义第*i*个流向的总包裹数为N*i*，原平均价格为P*i*，最大的折扣数为MAX\_DISCOUNT*i*，有

新的平均价格为

， 

收益的包裹数：





收益变化量：



第*i*个流向投入*x*折扣，包裹数增量为

newDataAnalysis.m假定，得到

和，

其中

分别存入矩阵SLOPE和INCREMENT

判定方案：

向量**DISCOUNT**和**NUMPKG**分别用于存放第*i*个流向投入的折扣和相应的包裹数增量。

第一步：

每次增加step元折扣，找到使最大的i，有

**DISCOUNT**（*i*）= **DISCOUNT**（*i*）+step

**NUMPKG**（*i*）= INCREMENT（**DISCOUNT**（*i*）， *i*）

第二步：

用j替代上面得到的i

取 ，其中k满足MAX\_DISCOUNT*k*>= **DISCOUNT**（*k*）且k≠j，找到使*Ik*最大且*Ik*>0的k ，将*k*和*j*对应的DISCOUNT交换，即

**DISCOUNT**（*j*）= **DISCOUNT**（*k*）

**NUMPKG**（*j*）= INCREMENT（**DISCOUNT**（*j*）， *j*）

**DISCOUNT**（*k*）= **DISCOUNT**（*j*）

**NUMPKG**（*k*）= INCREMENT（**DISCOUNT**（*k*）， *k*）

第三步：

找到DISCOUNT>0的所有流向q

n = length(q);

如果n大于1，取所有cnm为的所有组合





找到所有满足INCREMENT(*discount\_amount\_cnm*， *i*)>*numpkg\_amount\_cnm*且INCREMENT(*discount\_amount\_cnm*， *i*)最大的i

然后有

**DISCOUNT**（*i*）= *discount\_amount\_cnm*

**NUMPKG**（*i*）= INCREMENT(*discount\_amount\_cnm*， *i*)

对应的cmn流向的**DISCOUNT**和**NUMPKG**重置为0

第四步：

下次循环使用的step：

取raw\_step，3e6-SUM(**DISCOUNT**)和min(MAX\_DISCOUNT*i* > SUM (**DISCOUNT**)) - SUM(**DISCOUNT**)中的最小值。