分拣系统优化问题

摘要

本文主要研究对电商公司配送中心的分拣环节进行优化。首先确定优化目标为最少批次、摆放移动距离最短、分拣用时最短，在此前提下建立基于**贪心算法**的聚类模型与**模拟退火算法模型**。在满足约束的情况下，本文根据每个问题的不同条件，建立对应的最优策略模型，并针对具体数据进行分析。

1. 针对问题一：经过问题分析讨论，首先，本文将货品的订单信息表示为向量，并随之定义了**相似系数**和**距离**，以所含种类最多的订单为中心，计算其和其他订单的距离，基于特定条件选取距离较近的订单，并在保证合并后种类数不超过 200 时，归为同一批，之后取两订单合并作为新的中心订单计算其与其他订单的距离，在此基础上进行多次迭代，在所有订单均分批完毕后结束迭代。尝试使用**贪心算法**的思路去寻找最优的方法，通过以上模型，本文得到其最优参数为：在货架数量为 200 时，一共有 923 需要分类的订单，一共有 1941 货品种类，我们将上述订单最少分为 $J_{best} = 34$ 批。
2. 针对问题二：在问题 1 得到的分批结果基础上，首先将同一批次的各种货物在货架上随机排列，得到初始距离和，之后采用**退火算法**进行迭代优化，得到最终的货品摆放位置模型。应用问题 1 所建立的订单分批模型，对 34 批订单进行计算。对每批进行十次计算取最优值作为最终结果，已通过提交作品渠道提交。
3. 针对问题三：在问题 2 得到的货品摆放顺序结果基础上，首先将订单分成 5 批随机分发给 5 位分拣工人，得到初始距离和，然后采用**退火算法**进行迭代优化，得到最终的货品摆放位置模型。应用问题 2 的结果，对 34 批订单进行计算，对每批进行十次计算取最优值作为最终结果，已通过提交作品渠道提交。

关键字：贪心算法、模拟退火模型、优化算法、数据处理

目录

1 问题重述

1.1 问题背景

随着互联网时代的兴起，人们更青睐于各种网购的形式，这导致了对配送的需求日益提高，配送业务也得到了飞速的发展。随着越来越多的配送产业的出现与配送业务的不断增多，如何有效提升配送效率，将大大影响配送公司的核心竞争力。在这其中，配送环节中的分拣一步，安排操作复杂，重复劳动时间多，容易造成时间，人力，体力的浪费，不恰当的安排布局也会导致分拣过程中的错误，遗漏，影响效率提高成本。因此，合理科学的分拣方法显得尤为重要。

1.2 题目重述

随着公司业务量的增长，可能出现当天待配送订单所含货品种类数大于当前货架数量的情况（假设任何一个订单所含货品种类数均小于货架数量）。已知每份订单所需货品的种类及数量。假设货架数量固定为 N，每一个货架只摆放一种类型的货品，且不考虑货品质量与占地面积，做到尽可能提高分拣效率。

1. 问题一：将当日订单分为不同批次，每批次所含货品种类不超过 N，要求批次达到最少并给出每批订单的分批方案。

2. 问题二：分拣工在挑选货品的过程中，为了保证挑选效率的提高，让移动距离尽可能短，即把同一订单所含货品位置集中。在问题一最优方案的基础上，设计各批次每种货品的摆放位置，给出最佳摆放位置。

3. 问题三：现将所有订单分配给多个分拣工并完成分拣任务。在前两问基础上，当分拣工人为 5 人时，给出可以使工人行走的总路程最短的任务分配方案。

2 问题分析

2.1 问题一分析

对于问题一，首先根据题目所给条件与信息及附件中给出的所有订单，整理出各个订单所包含的种类信息。由此我们可以计算出包含种类最多的订单作为“中心订单”，并且计算其他订单与中心订单的距离，基于特定条件选择距离最小的订单与中心订单尝试合并为新的中心订单，并计算合并后的新订单与其他订单的距离，进行多次迭代。在无法合并时开辟新的批次。调整参数得到最优的分拣方法。

2.2 问题二分析

对于问题二，主体思路是采用退火算法交换货架位置得到最优解。首先针对同一批次随机排列货品摆放位置（显然，如果我们想得到最近的距离那么这些货品一定是紧挨着摆放的），得到初始距离之和。之后随机交换摆放位置得到新的距离，若距离减小，则采用此次交换；若距离增大，则通过公式按一定概率交换。随着尝试交换次数的增加，货品的摆放顺序愈发趋近于一个稳定的排序。多次运行之后，将稳定的货架排序视为最优货品摆放位置。

2.3 问题三分析

对于问题三，主体思路延续上一题的退火算法原理，在货品摆放次序确定之后，我们需要考虑的是 5 个分拣工他们之间应该如何分配分拣订单任务。现在随机给每个工人发放订单。之后随机交换订单的负责分拣工与分拣顺序，若距离减少则采用此次交换，若距离增大，则通过公式按一定概率交换。随着常数交换次数增加，各个工人走的总路程趋向稳定，任务分配方案趋向最优。

3 模型假设

1. 假设货架没有容积限制，且搬运货物不计时间，既忽视同一种类货品的数目信息。
2. 搬运工匀速运动，且互不打扰。
3. 搬运工同时只能处理一个订单，并且搬运能力没有限制，可以同时搬运任意多货品。
4. 搬运工在完成一个订单后可以立即开始下一个订单，无需回到起点。

4 符号约定

表 1: 符号约定

符号	含义
$\boldsymbol{O}_{i,j}$	编号为 i 的，属于第 j 批的订单
$\boldsymbol{M}$	批次的中心订单
$n(\boldsymbol{O})$	订单 $\boldsymbol{O}$ 所包含货品种类数
J_{max}	初始最大批次数
d_{max}	初始极限距离
$wd(\boldsymbol{X},\boldsymbol{Y})$	向量 $\boldsymbol{X}$ 与 $\boldsymbol{Y}$ 的按位相乘
$max\ \boldsymbol{X}, min\ \boldsymbol{X}$	$\boldsymbol{X}$ 中各分量的最大值和最小值
$\bigvee$	累计取或
$\boldsymbol{W}$	工人工作顺序信息
rw	一批订单内的订单数
f	货架上的订单的货架号
r	工人所在位置

5 模型建立

5.1 数据预处理

在题目所给的文件中，我们得到的是每个订单的每一个种类的货品的数目，但考虑我们所面临的三个问题，并没有涉及到有关物品数量的信息，所以我们可以将其整理成货品“订单号——订单内的货品种类”这样的信息。

具体整理方法如下：

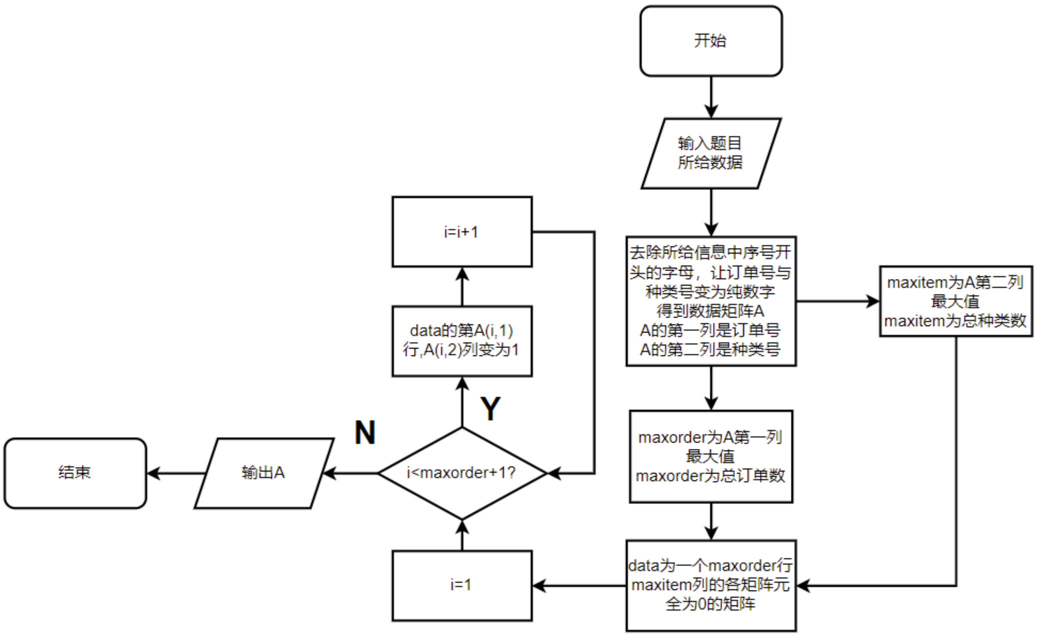

图 1: 数据预处理流程图

上述得到的 A 的第 i 行记作 $\boldsymbol{O}_i$，表示第 i 个订单的种类信息，在后文中使用 $\boldsymbol{O}_i$ 来称呼第 i 个订单。

由此得到的 $\boldsymbol{O}_i$ 是一个所有位置均为 0 或 1 的向量。$\boldsymbol{O}_i(j)=0$ 代表 O_i 中没有第 j 种货品，$\boldsymbol{O}_i(j)=1$ 代表 $\boldsymbol{O}_i$ 中有第 j 种货品。

5.2 问题一的模型建立与求解

5.2.1 问题一的模型建立

对于第一问的问题，基于贪心算法的聚类分析予以尝试解决。

在数据的预处理中，我们将所有订单的种类信息写成了向量的形式，这样，我们就引入相似系数的概念：相似系数指的是两个对象之间的相似度，对于聚类分析的问题来说，我们可以基于相似度较好的定义一个距离。在本题中，我们引入两个订单 i,j 的坐标向量的余弦值作为订单的之间的相似系数。计算方法为：

$$\chi_{i,j} = \frac{\boldsymbol{O}_i \cdot \boldsymbol{O}_j}{|\boldsymbol{O}_i|\,|\boldsymbol{O}_j|} \tag{1}$$

可知越趋向于 1，表示两个订单的相似程度越高。

我们根据相似系数可以定义余弦距离：

$$d_{i,j}^{(cos)} = 1 - \chi_{i,j} \tag{2}$$

那么，距离越短说明，两个订单的相似程度越好。这里值得说明的是：余弦距离满足的正定性与对称性，但是不满足三角不等式，因此余弦距离不是一个严格定义的距离。

在每次迭代的过程中，我们选择距离最“近”的订单尝试加入。在每次成功加入后，都需将目前的批次内的所有订单取或运算作为一个批次的新的中心订单，由这个订单重新与剩余所有订单计算距离。中心订单的第 i 个分量表示这批订单是否有第 i 种货物。我们使用 $\boldsymbol{O}_{i,j}$ 中的 j 来表示 $\boldsymbol{O}_i$ 属于第 j 批订单。

那么第 n 批订单的中心向量 $\boldsymbol{M}$ 由如下等式计算：

$$\boldsymbol{M} = \bigvee_{j=n} \boldsymbol{O}_{i,j} \tag{3}$$

定义 $n(\boldsymbol{O})$ 为 $\boldsymbol{O}$ 中所包含的种类的总数，则实际上 $n(\boldsymbol{O})$ 就等于 $\boldsymbol{O}$ 的模方（在本文中“种类数”运算的自变量均为只有 0 和 1 值的向量，仅表示某一种类在该订单或该批次货物中的有无）。

$$n(\boldsymbol{O}) = |\boldsymbol{O}|^2 \tag{4}$$

为了更好的对订单进行分批，我们设置一个初始最大订单数 J_{max}，和一个初始最远距离限制 d_{max}，我们在 J_{max} 的限制下逐步增加 d_{max} 来放开限制，当无论如何都无法在 J_{max} 批次内完成分配时考虑增加批次来处理剩余订单。

5.2.2 问题一的模型求解

基于贪心算法的聚类分析的思想，我们可以分为以下步骤：

1. 先给定最大允许的批次数为 J_{max}，然后寻找货品种类最多的订单作为中心订单，放入当前批次中。

2. 计算其余订单与中心订单的余弦距离，我们选择在极限距离 d_{max} 之内，且余弦距离最小的订单，尝试将其归为同一批次。

3. 计算合并后种类是否超过 200 种，若没有超过则将它们归置在同一批订单种并计算新的中心订单，重复步骤 2。若超过则跳过这次订单，选择在极限距离内且距离较近的其他订单重复判断，直到穷尽所有满足条件的订单为止，并开始下一批次的分配。

4. 而在新的批次中，我们选择余下订单中货品种类最多的订单作为中心订单并加入当前订单中，重复步骤 2 和步骤 3，直到批次数达到 J_{max}。

5. 之后，我们逐步调整余弦距离的允许范围更大，尝试向之前批次中加入新的订单，加入方法与此前相同，从距离限制内距离最近的订单开始遍历。直到 d_{max} 扩大到大于 1（本文中计算余弦距离的向量的分量均为 0 或 1，所以得到的相似系数均为 0 到 1，余弦距离的范围也均为 0 到 1，因此 d_{max} 大于 1 后将不会构成限制。）。

6. 若此时仍有剩余订单，我们在原有的批次基础上突破批次限制再增加新的一个批次，将剩下的订单中货品种类最多的货物作为中心订单，重复上边的操作，不一样的是此时不再有距离限制，按照余弦距离从小到大尝试加入批次，若成功加入便重新计算中心订单与距离，再次重复寻找订单尝试合并。

7. 重复步骤 6，直至所有订单均已分批完毕。

流程图如下所示：

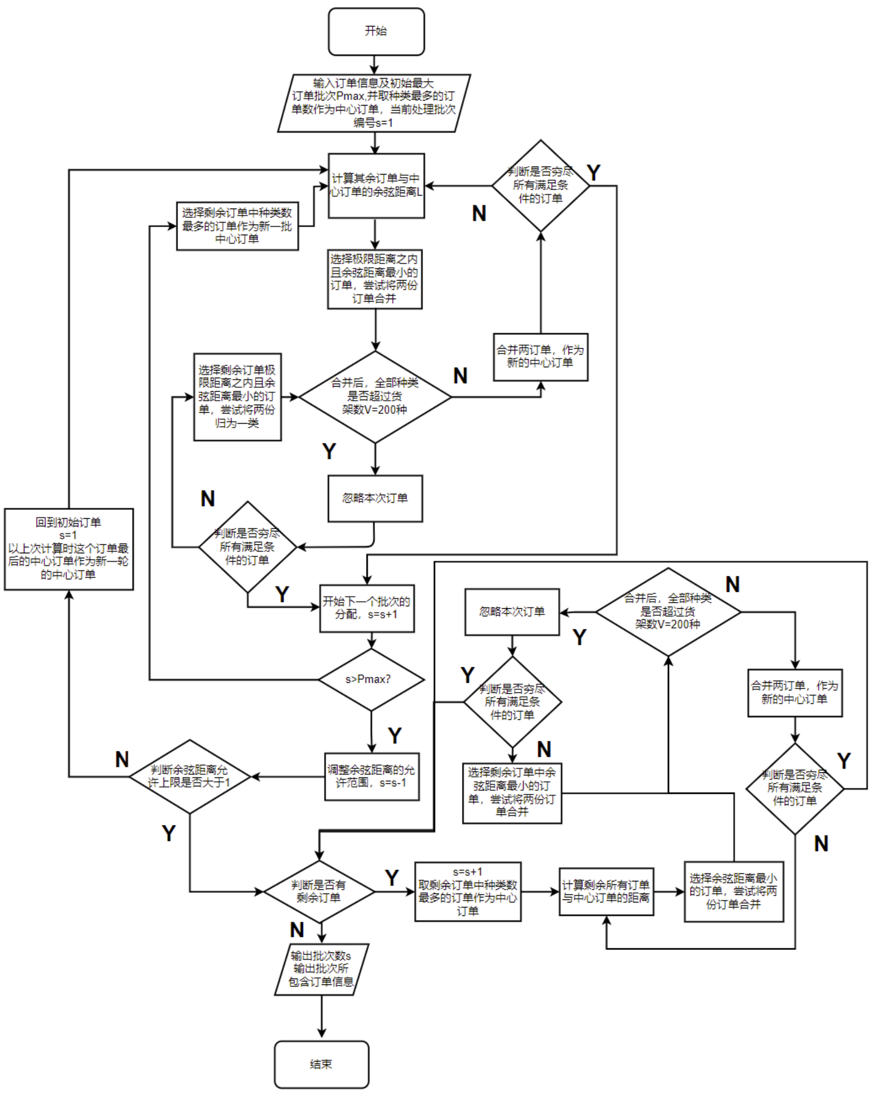

图 2: 问题一流程图

5.3 问题二的模型建立与求解

5.3.1 问题二的模型建立

基于题目要求，我们考虑的问题是如何使距离和函数取到最小值。

在本题中，自变量是不同种类的货品的摆放位置，我们选择的策略是使用模拟退火算法来尝试得到这个最优解。

下面先介绍我们对距离和的算法。

不妨我们先假设一个简单的情况，我们假设存在一批货品，这批货物一共有三个订单，5 种不同的货物，所有订单一共有六种货品。考虑如何计算距离和。

第一个订单有 1 号货品、3 号货品、4 号货品。

第一个订单有 4 号货品、5 号货品。

第一个订单有 1 号货品、6 号货品。

那么有：

$$\begin{aligned} \boldsymbol{O}_1 &= [1,0,1,1,0,0] \\ \boldsymbol{O}_2 &= [0,0,0,1,1,0] \\ \boldsymbol{O}_3 &= [1,0,0,0,0,1] \end{aligned} \tag{5}$$

以及：

$$\boldsymbol{M} = [1,0,1,1,1,1] \tag{6}$$

我们定义这样一个的矩阵 A

矩阵的第一行为中心向量 $\boldsymbol{M}$，他表示在这批货物中有哪几种货品。

对于矩阵的第二行，第一行为 1 的列第二行为当前列数，否则为 0，他记录的是货品的种类序号。

对于矩阵的第三行，第一行为 1 的列第三行为货品摆放着货架上对应的货架序号，否则为 0，上述例子中我们可以假设：

一号货品摆放在三号位置，

三号货品摆放着四号位置，

四号货品摆放在一号位置，

五号货品摆放在五号位置，

六号货品摆放在二号位置。

那么对应 A 有：

$$A = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 3 & 4 & 5 & 6 \\ 3 & 0 & 4 & 1 & 5 & 2 \end{pmatrix} \tag{7}$$

对于任意一个在此批内的订单，将 $\boldsymbol{O}_1$ 与 A 的第三行按位相乘就得到订单对应的在货架上的位置信息。

我们定义 $\boldsymbol{X}_1$，$\boldsymbol{Y}_1$ 两向量按位相乘为：

$$\boldsymbol{C} = wd(\boldsymbol{X}, \boldsymbol{Y}) \tag{8}$$

记矩阵 A 的第三行为：$A(3,:)$

那么就有 $\boldsymbol{O}$ 的位置信息为 $\boldsymbol{R} = wd(\boldsymbol{O}, A(3,:))$

则有题干中的拣选距离 $d(\boldsymbol{O})$:

$$\begin{aligned} d(\boldsymbol{O}) &= max_{\boldsymbol{O}(i)=1} S(i) - min_{\boldsymbol{O}(i)=1} S(i) \\ &= max\ R - min\ R(R(i) \neq 0) \\ &= max\ R + max(-R) \end{aligned} \tag{9}$$

第 n 批拣选距离总和 $D_n = \sum\limits_{\boldsymbol{O}_{i,j}, j=n} d(\boldsymbol{O}_{i,j})$。

回到上述例子，对于第一个订单便有：

$$\begin{aligned} \boldsymbol{R} &= wd(\boldsymbol{O}, A(3,:)) \\ &= [3, 0, 4, 1, 0, 0] \end{aligned} \tag{10}$$

对于第一个订单，有第 1、3、4 号货品，随对应的货架便是 3、4、1。

继续的，我们可以计算：

$$\begin{aligned} d(\boldsymbol{O}_1) &= max_{\boldsymbol{O}_1(i)=1} S(i) - min_{\boldsymbol{O}_1(i)=1} S(i) \\ &= max\ R_1 - min\ R_1(R_1(i) \neq 0) \\ &= max\ R_1 + max(-R_1) \\ &= 3 \end{aligned} \tag{11}$$

同理有：$d(\boldsymbol{O}_2) = 4, d(\boldsymbol{O}_3) = 1$。有 $D = 3 + 4 + 1 = 8$。

基于以上方式，我们可以得到一批订单内拣选距离总和 D 的计算。

显而易见的是，最优解中货品摆放顺序应当是连续的。

这带来的约束条件是 $A(3,:)$ 在从小到大排序后应当为连续的，且除了 0 以外没有重复数字的整数。

在这个约束条件下使 D 达到极小值。

5.3.2 问题二的模型求解

对于该优化过程，我们尝试采取模拟退火算法来进行求解。

在我们利用模拟退火算法来解决问题之前，让我们先简单来介绍一下该优化方法。

通过固体的退火原理我们得到了模拟退火算法，这是一种基于概率的算法：在固体加温至一定温度时，使其逐渐冷却，我们在加热的过程中时，固体内部的粒子随着温度的升高而不断趋于无序化，内能也随之增大，而在逐渐冷却的过程中，粒子又慢慢趋向于有序，并最后在常温时达到基态，内能减为最小。从本质上而言，模拟退火算法是一种双层循环，外层循环我们通过控制温度由高向低变化，初始温度不断乘以一个取值在 [0, 1] 上的温度衰减系数 α，如 0.95；而在内层循环中，我们将温度固定，对旧解随机加入一个扰动得到一个新解，并按已知的规则去接受新解。内层循环的迭代次数称为马尔科夫链长度 M。

对于模拟退火的思想可以用下面的流程图来展示：

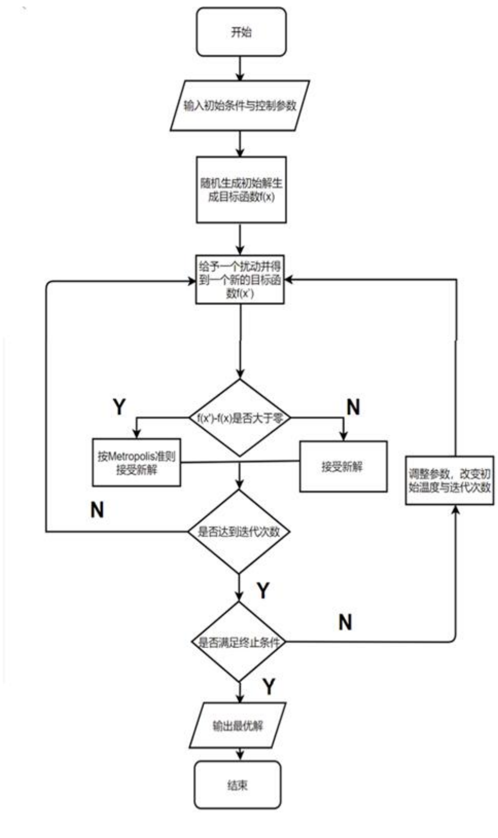

图 3: 退火流程图

在利用模拟退火对问题进行计算时，我们可以将步骤分为如下内容：

1. E 对应于退火问题中的内能项，在此题目中定义为当前批次内所有订单拣选距离总和。首先，我们先对订单中所涉及到的物品进行随机排序（由于最优解一定是紧凑的，我们只需要在 1 到中心订单的种类数 $n(\boldsymbol{M})$ 对应的货架之间进行排序，所以我们只需生成 1 到 $n(\boldsymbol{M})$ 之间随机排列的整数列，来对应到中心订单的各个货品种类作为我们的初始摆放位置）。

2. 设置模拟退火参数：初始温度 20，温度衰减系数 a=0.99，稳定温度 0.1，马尔可夫链长度 M=1000，历史最优内能为当前解所计算的内能，历史最优解为当前解。

3. 对当前解进行随机扰动，得到新解，判断新解随对应内能是否优于当前，若优于，记录新解为当前解并进入第四步，否则进入第五步。

4. 判断新解是否优于历史最优解，若优于，则记录新解为新的历史最优，进入第六步。

5. 生成随机数，判断随机数是否小于临界概率：$P = \dfrac{E_{新} - E_{目前}}{T}$，若小于则记录新解为当前解，否则舍弃新解。

6. 判断迭代次数是否达到马尔科夫链长度，若未达到则返回第三步。

7. 判断温度是否低于稳定温度，若不低于稳定温度则温度衰减（$T' = T \cdot \alpha$），迭代次数清零，返回第三步。

8. 输出最优解。

在实践过程中，我们发现如果全局的马尔科夫链过长会导致程序过慢，而后半段马尔可夫链过短会导致无法达到局域最优解，因此在温度低于 1 时将马尔科夫链延长到 10000。

如图为第一批订单在上述代码运算过程中绘制的内能随温度变化曲线，可以看到内能在最后趋于定值。

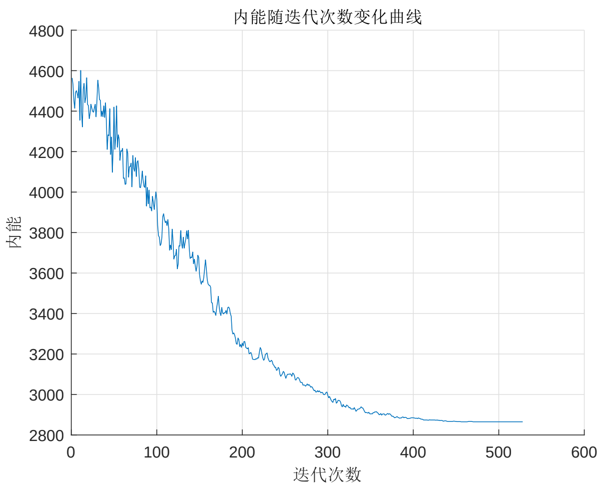

图 4: 内能随迭代次数变化

5.4 问题三的模型建立与求解

5.4.1 问题三的模型建立

题目所求的就是每个工人所分配的订单编号和订单顺序，也就是说我们的自变量是工人的订单分配情况和顺序，为使目标函数——各工人所走路程最大值最短。

在下文讨论中，将货架视作有货架编号视作小编号在左，大编号在右的一排货架，下文中的最左端货架等价于货架编号最小的货架；同理，最右段货架即为货架编号最大的编号。

定义 rw 为该批订单的订单数。

为了能更好描述订单的分配情况，并且避免所有订单分配到一个工人导致内存溢出，在工人数为 5 的前提下，我们定义了一个 $(rw+1)$ 行 5 列的二维矩阵。第 n 行表示工人完成的第 n 个订单，第 m 列表示第 m 个工人去完成。

因为题目中工人的移动速度一定且互不干扰，最佳分配策略对应最短距离 $Dg(\boldsymbol{W})$。$\boldsymbol{W}$ 为一个一维向量，$\boldsymbol{W}$ 的长度为这批订单的订单数量，也就是 rw，$\boldsymbol{W}$ 中的第 p 个坐标对应的是订单号从小到大排列后第 p 个订单的序列号 w_p。序列号 w 对 5 取余加 1 得到分配的工人编号，除以 5 后向上取整得到这个对于这个工人完成这件订单的次序。

例如某一批订单有 6、7、16、17、21、53 六个订单，此时 x=[5,9,2,1,4,3]，那么 7 号订单对应的位置编号为 9。它是第 5 个工人（$((9mod5)+1)=5$），的第二个订单（$\left\lceil \frac{9}{5} \right\rceil$）。

分拣工的最优分拣方式为尽可能不走重复路线。因此每个订单只包含最靠左货品位置 f_{max} 和最靠右货品位置 f_{min} 两个有效信息。当工人在最左和最右位置连线范围之外时，工人选择走直线。当工人在最左和最右位置之间时，工人先去路程较近的边界点，再去另一个路程较远的点。

则每个订单包含的有效信息仅为：$[f_{min}, f_{max}]$

考虑到工人开始第 k 个订单的位置取决于第 k-1 个订单，我们将完成第 k 个订单的情况分为以下三种情况：

在以下讨论中记 r_k 为第 k 次拣择时候的位置，s_k 为第 k 次拣择后所走路程，$s_1 = |r_{k-1} - f_{min,k}|$，$s_2 = |r_{k-1} - f_{max,k}|$，$s_3 = |f_{max,k} - f_{min,k}|$

当 $|r_{k-1} - f_{min,k}| < |r_{k-1} - f_{max,k}|$ 时，$r_k = f_{max,k}$，$d_k = s_1 + s_3$；

当 $|r_{k-1} - f_{min,k}| > |r_{k-1} - f_{max,k}|$ 时，$r_k = f_{min,k}$，$d_k = s_2 + s_3$；

当 $|r_{k-1} - f_{min,k}| = |r_{k-1} - f_{max,k}|$ 时，此时我们需要考虑第 k+1 个订单。在第 k 个订单中，如果我们选择先走向左端点，那么我们最终会出现在右端点；如果我们选择先走向右端点，那么我们最终会出现在左端点。那么我们考虑去下一个订单的左右端点的距离。

在详细讨论之前，我们先看这张图，这张图所绘制的是在两侧距离相同时且当前任务并非最后一个任务时，工人先去右侧端点再去左侧端点这种情况的距离计算方法。先去左侧时与此种情况类似，不再重复论述。

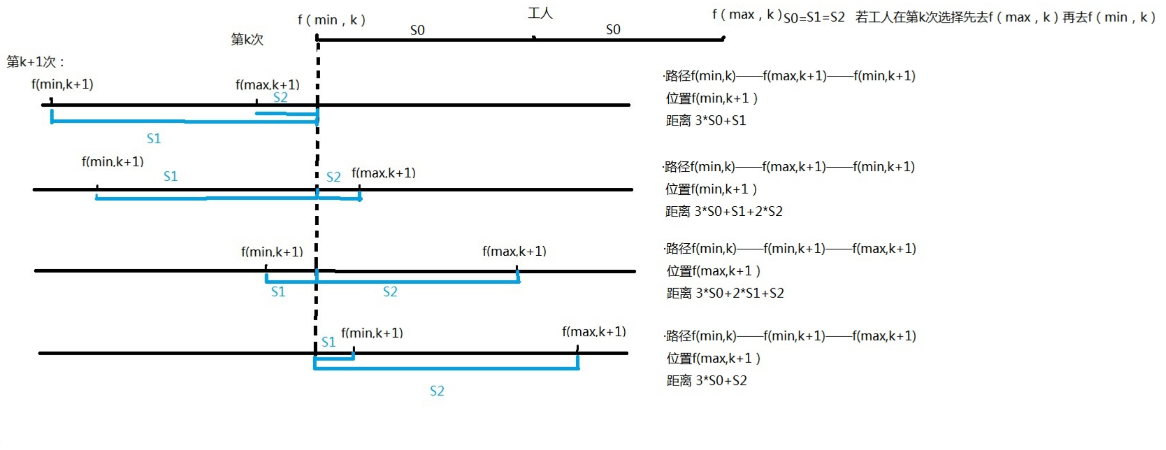

图 5: 工人先去右侧时的距离计算

在第 k 任务并非最后一个任务时，有以下讨论：

我们考虑选择先去往左端点，那么下次拣择时我们会出现在右端点，考虑右端点距离下个订单左右端点的距离。

若 $|f_{max,k}-f_{min,k+1}|>|f_{max,k}-f_{max,k+1}|$，则表示我们“——左——右”这样走会比“——左——左”更近。“——左——左”的距离记作 $sj_1=|f_{max,k}-f_{max,k+1}|$。

若 $|f_{max,k}-f_{min,k+1}|<|f_{max,k}-f_{max,k+1}|$，则表示我们“——左——左”这样走会比“——左——右”更近。“——左——右”的距离记作 $sj_1=|f_{max,k}-f_{min,k+1}|$。

我们考虑选择先去往右端点，那么下次抉择时我们会出现在左端点，考虑左端点距离下个订单左右端点的距离。

若 $|f_{min,k}-f_{min,k+1}|>|f_{min,k}-f_{max,k+1}|$，则表示我们“——右——右”这样走会比“——右——左”更近。“——左——左”的距离记作 $sj_2=|f_{min,k}-f_{max,k+1}|$。

若 $|f_{min,k}-f_{min,k+1}|<|f_{min,k}-f_{max,k+1}|$，则表示我们“——右——左”这样走会比“——右——右”更近。“——左——右”的距离记作 $sj_2=|f_{min,k}-f_{min,k+1}|$。

优于当前距离左右端点距离相等，且当前订单 s_3 和下一订单 s_3 保持恒定不变，所以选择向左和选择向右走的要走的路程的差为 sj_1-sj_2。当 sj_1-sj_2 大于 0 时向右走，否则向左走。

当当前订单为最后一订单时，选择向先向右段走，以保证结束订单时出现在左端点，走回起点的耗时更短，以 r 记为工人完成订单后所在位置。

基于以上讨论我们给出了工人的最佳抉择方式，有了这个我们便可以根据已有任务表算出每个工人总共要走的路程。

$$Dg=\sum_k d_k+r-1 \tag{12}$$

取目标函数为五个工人的移动距离的最大值。这样在目标函数取到最小值的时候各个工人走的路程趋向于平均且最小。

5.4.2 问题三的模型求解

求解方法问题三与问题二的基本求解思路相同，均使用模拟退火算法进行求解。仅距离的算法和交换函数有所不同，顾不再赘述。

6 模型评价

6.1 模型优点

1）基于贪心算法的聚类模型：使用贪心算法，我们可以很大程度上，以一种更简单更快速的方法去得到一个最优解，即在对问题求解时，我们总是做出从当前来看是最好的选择，这是某种意义上的局部最优解。而在本题中，通过贪心算法，我们避免了贪心可能导致的局部性，以更简洁更有效的表现出全局性较优解。2）在考虑第一问中，两个订单的相似系数时，本文利用余弦距离来表示相似程度，不仅提高了两个订单的相似系数的准确度，更简化了问题的求解难度，让问题更加便于计算与表示。3）对于问题二，问题三，我们基于模拟退火模型，通过合理的设置算法参数，避免了局部解而更接近一个全局最优解。

6.2 模型缺点

1，在使用贪心算法的过程中，为了保证不会陷入局部最优解，本文增添了许多限制条件与判断过程，使得程序显得冗长复杂，不够简洁明了。2，在问题二中，为了使模拟退火算法趋向全局最优解，我们设置了两个不同的马尔可夫链长度，分别对应于 1°C 之前的 1000 与 1°C 到 0.1°C 之间的 5000，这样操作的结果虽然使我们得到的数据更加稳定，而带来的后果就是在进行数据处理时，电脑的计算过于复杂缓慢，计算过程较长。

参考文献

[1] 李诗珍　基于聚类分析的订单分批拣货模型及启发式算法 [J] 长江大学 1002- 6487(2008) 12-0053- 02

[2] https://blog.csdn.net/bingokunkun/article/details/118583729 | 小坤兽“退火算法（Annealing）简介与详解”【Online】Available（2021/7/11）

7 附录：代码

7.1 数据预处理

第一部分代码为附件中名为“yuchuli.m”的代码部分，基于提供的信息“附件 1：订单信息.csv”，这部分代码将所给的订单信息进行预处理，输出“fenllei”变量，储存于“fenlei.mat”中。

```
clear;
fid=importdata('附件1：订单信息.csv');
textdata=fid.textdata;
textdata(1,:)=[];
[r,~]=size(textdata);
for i=1:r
    textdata{i,1}(1)=[];
    textdata{i,2}(1)=[];
end
dat=zeros(length(textdata),2);
for i = 1:length(textdata)
    dat(i,1) = str2double(textdata{i,1});
    dat(i,2) = str2double(textdata{i,2});
end
data=fid.data;
data=[dat,data];
[r,~]=size(data);
maxitem=max(data(:,2));
fenllei=zeros(data(r,1),maxitem);
for t=1:r
    fenllei(data(t,1),data(t,2))=1;
end
```

7.2 第一问求解

第二部分代码为附件中名为"f1.m" 的代码部分，这部分代码对数据进行了分批，最终得到 34 批的结果。这部分代码运行依赖“fenlei.mat”文件。文件运行生成“bb”和“pi”两个变量，二者共同存储了订单的批次信息。将其打包存储在“pppp.mat”内。

```
clear;
load("fenlei.mat");
item=sum(fenllei');
[aa,bb]=sort(item);
bb=fliplr(bb);
pi=zeros(2,2);
[r,c]=size(fenllei);
s=0;
re=0;
yangben=zeros(r,c);
jiexian=34;
te=zeros(1,c);
f=ones(r,1);
for t=1:r
    yangben(t,:)=fenllei(bb(t),:);
end
diyici=true;
for linjie=0.85:0.01:1
    if diyici
        diyici=false;
        for y=1:r
            mm=1;
            if f(y)
                f(y)=0;
                s=s+1;sss=1;s
                pi(s,sss)=y;
                ff=zeros(1,r);
                te(s,:)=yangben(y,:);
                for t=1:r
                    ff(t)=1-dot(yangben(y,:),yangben(t,:))/...
                        (norm(yangben(y,:))*norm(yangben(t,:)));
                end
```

```
                while mm
                    mm=0;
                    [~,fff]=sort(ff);
                    for q=1:r
                        if f(fff(q)) && ff(fff(q))<=linjie
                            tes=te(s,:)|yangben(fff(q),:);
                            if sum(tes)<=200
                                sss=sss+1;
                                te(s,:)=tes;
                                f(fff(q))=0;
                                pi(s,sss)=fff(q);
                                re=1;mm=1;
                            end
                        end
                        if re
                            for t=1:r
                                ff(t)=1-dot(te(s,:),yangben(t,:))/...
                                    (norm(te(s,:))*norm(yangben(t,:)));
                            end
                            re=0;
                            break;
                        end
                    end
                end
            end
            if s==jiexian
                break;
            end
        end
    else
        for jk=1:jiexian
            mm=1;abc=pi(jk,:);abc(abc==0)=[];sss=length(abc);
            while mm
                for t=1:r
                    ff(t)=1-dot(te(jk,:),yangben(t,:))/(norm...
                        (te(jk,:))*norm(yangben(t,:)));
                end
```

```
                    mm=0;
                    [~,fff]=sort(ff);
                    for q=1:r
                        if f(fff(q)) && ff(fff(q))<=linjie
                            tes=te(jk,:)|yangben(fff(q),:);
                            if sum(tes)<=200
                                sss=sss+1;
                                te(jk,:)=tes;
                                f(fff(q))=0;
                                pi(jk,sss)=fff(q);
                                re=1;mm=1;
                            end
                        end
                        if re
                            for t=1:r
                                ff(t)=1-dot(te(jk,:),yangben(t,:))/(norm...
                                    (te(jk,:))*norm(yangben(t,:)));
                            end
                            re=0;
                            break;
                        end
                    end
                end
            end
        end
end
for y=1:r
    mm=1;
    if f(y)
        f(y)=0;
        s=s+1;sss=1;
        pi(s,sss)=y;
        ff=zeros(1,r);
        te(s,:)=yangben(y,:);
        for t=1:r
            ff(t)=1-dot(yangben(y,:),yangben(t,:))/(norm...
                (yangben(y,:))*norm(yangben(t,:)));
```

```
        end
        while mm
            mm=0;
            [~,fff]=sort(ff);
            for q=1:r
                if f(fff(q))
                    tes=te(s,:)|yangben(fff(q),:);
                    if sum(tes)<=200
                        sss=sss+1;
                        te(s,:)=tes;
                        f(fff(q))=0;
                        pi(s,sss)=fff(q);
                        re=1;mm=1;
                    end
                end
                if re
                    for t=1:r
                        ff(t)=1-dot(te(s,:),yangben(t,:))/(norm...
                            (te(s,:))*norm(yangben(t,:)));
                    end
                    re=0;
                    break;
                end
            end
        end
    end
end
```

7.3 第一问结果输出

第三部分代码是“result1.m”代码部分，基于“fenlei.mat”和“pppp.mat”生成第一题的解“result1.csv”。

```
clear;
load("fenlei.mat");
load("pppp.mat")
result=zeros(923,2);
result(:,1)=1:923;
[r,c]=size(pi);
for i=1:r
    for j=1:c
        if pi(i,j)
            result(bb(pi(i,j)),2)=i;
        end
    end
end
res=int2str(result(:,1));
res(res==' ')='0';
while length(res(1,:))<4
    res=[int2str(zeros(length(res),1)),res];
end
a=int2str(zeros(length(res),1));
a(a=='0')='D';
res=[a,res];
various={'OrderNo','GroupNo'};
tab=table(res,result(:,2),'VariableNames',various);
writetable(tab, 'result1.csv')
```

7.4 第二问求解

第四部分代码是“q2.m”代码部分，基于“fenlei.mat”和“pppp.mat”给出第二题的答案——货品种类排列方法，为变量“shuju2”，存储在“shuju2.mat”内，由于退火算法的随机性，每次的输出值并不唯一。

```
clear;
load("pppp.mat");
load("fenlei.mat");
[ad,~]=size(pi);
shuju2=zeros(4*ad,200+1);
for i=1:ad
    shuju2(i,1)=ceil(i/4);
end
for ida=1:ad
    E_bbest=inf;
    for iad=1:10
        load("pppp.mat");
        load("fenlei.mat");
        ppi=zeros(size(pi));
        for i=1:length(bb)
            ppi(pi==i)=bb(i);
        end
        pi=ppi;
        pi1=pi(ida,:);
        pi1(pi1==0)=[];
        rs=fenllei(pi1(1),:);
        for i=1:length(pi1)
            rs=double(rs|fenllei(pi1(i),:));
        end
        lll=sum(rs);
        for i=1:length(rs)
            rs(2,i)=i*rs(1,i);
        end
        ll=i;
        rss=rs;
        rs(:,all(rs==0,1))=[];
        rs(3,:)=randperm(length(rs));
```

```
j=1;
for i=1:ll
    if rss(2,i)==rs(2,j)
        rss(3,i)=rs(3,j);
        j=j+1;
        if j>lll
            break;
        end
    end
end
sol_new=rs;
slonew=rss;
sol_current = sol_new; sol_best = sol_new;
T=20;a=0.99;T1=0.1;M=1000;
E_current = inf;E_best = inf;
while T>T1
    for i=1:M
        if (rand < 0.5)
            ind1 = 0; ind2 = 0;
            while (ind1 == ind2)
                ind1 = ceil(rand.*lll);
                ind2 = ceil(rand.*lll);
            end
            tmp1 = sol_new(3,ind1);
            sol_new(3,ind1) = sol_new(3,ind2);
            sol_new(3,ind2) = tmp1;
        else
            ind1 = 0; ind2 = 0; ind3 = 0;
            while (ind1 == ind2) || (ind1 == ind3) ...
                    || (ind2 == ind3) || (abs(ind1-ind2) == 1)
                ind1 = ceil(rand.*lll);
                ind2 = ceil(rand.*lll);
                ind3 = ceil(rand.*lll);
            end
            tmp1 = ind1;tmp2 = ind2;tmp3 = ind3;
            if (ind1 < ind2) && (ind2 < ind3)
            elseif (ind1 < ind3) && (ind3 < ind2)
```

```
            ind2 = tmp3;ind3 = tmp2;
        elseif (ind2 < ind1) && (ind1 < ind3)
            ind1 = tmp2;ind2 = tmp1;
        elseif (ind2 < ind3) && (ind3 < ind1)
            ind1 = tmp2;ind2 = tmp3; ind3 = tmp1;
        elseif (ind3 < ind1) && (ind1 < ind2)
            ind1 = tmp3;ind2 = tmp1; ind3 = tmp2;
        elseif (ind3 < ind2) && (ind2 < ind1)
            ind1 = tmp3;ind2 = tmp2; ind3 = tmp1;
        end
        tmplist1 = sol_new(3,(ind1+1):(ind2-1));
        sol_new(3,(ind1+1):(ind1+ind3-ind2+1)) = sol_new
           (3,(ind2):(ind3));
        sol_new(3,(ind1+ind3-ind2+2):ind3) = tmplist1;
    end
    kk=1;
    for k=1:ll
        if slonew(2,k)==sol_new(2,kk)
            slonew(3,k)=sol_new(3,kk);
            kk=kk+1;
            if kk>lll
                break;
            end
        end
    end
    E_new=0;
    for j=1:length(pi1)
        de=fenllei(pi1(j),:).*slonew(3,:);
        E_new=E_new+max(de)+max(-de);
    end
    if E_new < E_current
        E_current = E_new;
        sol_current = sol_new;
        if E_new < E_best
            E_best = E_new;
            sol_best = sol_new;
        end
```

```
                else
                    if rand < exp(-(E_new-E_current)./T)
                        E_current = E_new;
                        sol_current = sol_new;
                    else
                        sol_new = sol_current;
                    end
                end
            end
            T=T*a;
            if T<1
                M=10000;
            end
        end
        [rrrrrr,cccccc]=size(sol_best);
        sol_best=[sol_best,zeros(3,200-cccccc)];
        if E_best<E_bbest
            E_bbest=E_best;
            shuju2(ida*4-3,2)=E_best;
            shuju2(ida*4-2,2:201)=sol_best(1,:);
            shuju2(ida*4-1,2:201)=sol_best(2,:);
            shuju2(ida*4,2:201)=sol_best(3,:);
        end
    end
end
```

7.5 第二问结果输出

第五部分是“result2.m”代码部分，基于上一部分中生成的“shuju2.mat”，生成要求的“result2.csv”，由于“shuju2.mat”的不唯一性此次输出结果也不唯一。

```
result=zeros(0,2);
for i=1:34
    par1=shuju2(4*i-1,2:201)';
    par2=shuju2(4*i,2:201)';
    par1(par1==0)=[];
    par2(par2==0)=[];
    l=length(par1);
    par0=ones(l,1)*i;
    par=[par1,par0,par2];
    result=[result;par];
end
res=int2str(result(:,1));
res(res==' ')='0';
while length(res(1,:))<4
    res=[int2str(zeros(length(res),1)),res];
end
a=int2str(zeros(length(res),1));
a(a=='0')='P';
res=[a,res];
various={'ItemNo','GroupNo','ShelfNo'};
tab=table(res,result(:,2),result(:,3),'VariableNames',various);
writetable(tab, 'result2.csv')
```

7.6 第三问求解并输出结果

第六部分是“result3.m”代码部分，基于“fenlei.mat”、“pppp.mat”和“shuju2.mat”计算并生成第三问要求给出的分拣工分拣顺序，输出“result3.csv”，由于“shuju2.mat”的不唯一性此次输出结果也不唯一。

```
result=[];
for qwe=1:34
    for shici=1:3
        E_bbest=inf;
        sol_bbest=0;
        wwwww=0;
        N=5;
        www=zeros(1,N);
        wwww=zeros(1,N);
        load('pppp.mat');
        load('fenlei.mat');
        load('shuju2.mat');
        [r,c]=size(fenllei);
        [rr,cc]=size(pi);
        ppi=zeros(rr,cc);
        for zxcv=1:rr
            for zxcvb=1:cc
                if pi(zxcv,zxcvb)
                    ppi(zxcv,zxcvb)=bb(pi(zxcv,zxcvb));
                end
            end
        end
        shu=shuju2(qwe*4-2:qwe*4,2:201);
        [~,huijia]=size(shu);
        xuhao=zeros(3,c);
        j=1;
        piii=ppi(qwe,:);
        piii(piii==0)=[];
        for i=1:c
            if i==shu(2,j)
                xuhao(:,i)=shu(:,j);
                j=j+1;
```

```
            if j>huijia
                break;
            end
        end
    end
    juli=piii';
    ij=length(juli);
    for i=1:ij
        caijian=fenllei(juli(i,1),:).*xuhao(3,:);
        caijian(caijian==0)=[];
        juli(i,2)=min(caijian);
        juli(i,3)=max(caijian);
    end
    fangfa=zeros(ij+1,N);
    while 1
        ind1 = ceil(rand([1,ij]).*(N*(ij+1)));
        if ~(length(ind1)-length(unique(ind1)))
            break
        end
    end
    juli(:,4)=ind1';
    for i=1:ij
        fangfa(ceil(juli(i,4)/N),mod(juli(i,4),N)+1)=juli(i,1);
    end
    fq=fangfa;
    for i=1:N
        faf=fangfa(:,i);
        faf(faf==0)=[];
        faff=[faf;zeros(ij+1-length(faf),1)];
        fangfa(:,i)=faff;
    end
    jili1=zeros(r,4);
    for i=1:ij
        jili1(juli(i,1),:)=juli(i,:);
    end
    snew=fq;
    sol_new=fangfa;
```

```
scur=snew;sbest=snew;
sol_current = sol_new; sol_best = sol_new;
T=10;a=0.98;T1=0.01;M=5000;
E_current = inf;E_best = inf;
while T>T1
    for i=1:M
        ind1=1;ind2=1;
        while ind1==ind2 || ~(snew(ceil(ind1/N),mod(ind1,N)...
                +1) || snew(ceil(ind2/N),mod(ind2,N)+1))
            ind1 = ceil(rand.*(ij+1)*N);
            ind2 = ceil(rand.*(ij+1)*N);
        end
        tem1=snew(ceil(ind1/N),mod(ind1,N)+1);
        snew(ceil(ind1/N),mod(ind1,N)+1)=snew(ceil(ind2/N)...
            ,mod(ind2,N)+1);
        snew(ceil(ind2/N),mod(ind2,N)+1)=tem1;
        for j=1:N
            faf=snew(:,j);
            faf(faf==0)=[];
            faff=[faf;zeros(ij+1-length(faf),1)];
            sol_new(:,j)=faff;
        end
        E_new=0;
        for k=1:N
            ww=0;ste=1;
            for kk=1:ij+1
                if sol_new(kk,k)
                    s1=abs(ste-jili1(sol_new(kk,k),2));
                    s2=abs(ste-jili1(sol_new(kk,k),3));
                    s3=abs(jili1(sol_new(kk,k),3)-jili1...
                        (sol_new(kk,k),2));
                    if s1==s2 && sol_new(kk+1,k)
                        s111=abs(jili1(sol_new(kk+1,k),2)...
                            -jili1(sol_new(kk,k),3));
                        s112=abs(jili1(sol_new(kk+1,k),3)...
                            -jili1(sol_new(kk,k),3));
                        s121=abs(jili1(sol_new(kk+1,k),2)...
```

```
                    -jili1(sol_new(kk,k),2));
                s122=abs(jili1(sol_new(kk+1,k),3)...
                    -jili1(sol_new(kk,k),2));
                s11=min(s111,s112);s12=min(s121,s122);
                if s11<=s12
                    ww=ww+s1+s3;ste=jili1...
                        (sol_new(kk,k),3);
                else
                    ww=ww+s2+s3;ste=jili1...
                        (sol_new(kk,k),2);
                end
            elseif s1<s2
                ww=ww+s1+s3;ste=jili1(sol_new(kk,k),3);
            elseif s2>s1
                ww=ww+s2+s3;ste=jili1(sol_new(kk,k),2);
            else
                ww=ww+s2+s3;ste=jili1(sol_new(kk,k),2);
            end
        else
            ww=ww+abs(ste-1);
            break;
        end
    end
    E_new=max(E_new,ww);
    www(k)=ww;
end
if E_new < E_current
    E_current = E_new;
    sol_current = sol_new;
    scur=snew;
    if E_new < E_best
        E_best = E_new;
        sol_best = sol_new;
        sbest=snew;
        wwww=www;
    end
else
```

```
                    if rand < exp(-(E_new-E_current)./T)
                        E_current = E_new;
                        sol_current = sol_new;scur=snew;
                    else
                        sol_new = sol_current;snew=scur;
                    end
                end
                if sum(wwwww-wwww)
                end
                wwwww=wwww;
            end
            T=T*a;
        end
        if E_best<E_bbest
            E_bbest=E_best;
            sol_bbest=sol_best;
        end
    end
    [rre,cre]=size(result);
    [rso,cso]=size(sol_bbest);
    rmr=max(rre,rso);
    result=[result;zeros(rmr-rre,cre)];
    sol_bbest=[sol_bbest;zeros(rmr-rso,cso)];
    result=[result,sol_bbest];
    qwe
end
zongshu=sum(sum(logical(result)));
result33=zeros(zongshu,4);
for i=1:55
    for j=1:170
        if result(i,j)
            result33(result(i,j),1)=result(i,j);
            result33(result(i,j),2)=ceil(j/5);
            result33(result(i,j),3)=mod(j,5)+1;
            result33(result(i,j),4)=i;
        end
    end
```

```
end
result=result33;
res=int2str(result(:,1));
res(res==' ')='0';
while length(res(1,:))<4
    res=[int2str(zeros(length(res),1)),res];
end
a=int2str(zeros(length(res),1));
a(a=='0')='D';
res=[a,res];
various={'OrderNo','GroupNo','WorkerNo','TaskNo'};
tab=table(res,result(:,2),result(:,3),...
    result(:,4),'VariableNames',various);
writetable(tab, 'result3.csv')
```