

分拣系统优化问题研究

摘要

分拣环节是决定配送中心整体效率的关键环节。如何提高分拣效率以提高配送中心整体性能具有重要的现实价值与研究意义。本文建立了基于 0-1 整数规划的最小分批算法与基于订单相似性的分批算法、最佳货品摆放模型以及最佳指派分拣任务模型，实现分拣效率的提高。

针对任务一，对于设计最小分批算法，本文首先采用模型 0-1 整数规划来对问题求解，为满足在货架数（单批次货品种类数）下以最小批次分拣，本文以所需批次总数最小为目标，基于整数规划建立分批算法。此模型的求解因其数据量的巨大而面临困难，适合采用启发式算法进行求解，在此选用了遗传算法对该模型进行求解，求得的最小批次为 96。为求得更好得结果，本文进一步优化模型，采用根据聚类的算法思路建立了基于订单相似度的分批算法模型求解最小批次，最终求得结果为 66，远小于 96。在此基础之上，考虑订单中的货品种类这一因素，对基于订单相似度的分批算法模型进一步优化，得到的分批结果为 59，进一步减小批次数量。最后，对货架数量进行了灵敏度分析，以验证模型的实用性。

针对任务二，对于以最短拣选距离为目标，同时使得每个货品对应一个货架，本文将其理解为最短路径与旅行商问题的结合，建立最佳货品摆放模型。对于此模型，其求解难点在于订单中货物的分拣顺序不确定，因此本文采用了模拟退火的启发式方法进行计算。本文选择以第 1 批次为例，运用模拟退火进行求解，得到第 1 批次货品的摆放位置以及拣选距离总和。第 1 批次的拣选距离总和为 2193，其他批次的处理方法与第 1 批次类似，最终求得所有批次拣选距离总和为 64959。

针对任务三，对于某一批分拣任务，为任务分配尽量均衡，本文将尽快完成任务的目标转换为最小化任务最多的工人的分拣距离，保证每个分拣工人同一时刻只能分拣一个订单，且一个订单只能由一个分拣工人完成，建立最佳指派分拣任务模型。与任务二相似，以第 1 批次为例，运用模拟退火算法对模型进行求解，得到各个订单的最佳指派结果。

关键词：分批算法、货品摆放、指派问题、订单相似度、模拟退火算法

目录

1、问题背景与重述	1
1.1 问题背景	1
1.2 问题要求	1
2、模型假设及符号说明	1
2.1 模型假设	2
2.2 符号说明	2
3、分批算法的建立与求解	2
3.1 数据预处理	2
3.1.1 数据探索	2
3.1.2 数据筛选	3
3.2 基于 0-1 整数规划问题的分批算法模型	4
3.2.1 模型构建	4
3.2.2 模型求解	5
3.3 基于订单相似度的分批算法	7
3.3.1 模型构建	7
3.3.2 模型求解	7
3.3.3 模型优化	8
3.3.4 模型检验与分析	9
4、优化货品摆放位置模型的建立与求解	10
4.1 优化货品摆放模型的构建	10
4.2 优化货品摆放模型的求解	11
5、指派分拣任务模型的建立与求解	15
5.1 最佳指派分拣任务模型的构建	15
5.2 最佳指派分拣任务模型的求解	17
6、模型的优缺点	17
6.1 模型优点分析	17
6.2 模型缺点分析	17
参考文献	18

1、问题背景与重述

1.1 问题背景

电商公司配送中心的工作流程主要包括商品订单汇总统计、货物转运上架、订单分拣以及核对信息并进行打包等。即首先对当天需要配送处理的所有订单及订单中包含的商品进行汇总统计，其次转运工人将待配送的货物从仓库运往分拣地点并放到货架之上（每个货架进能够放置一种商品）。在这些环节中，分拣环节因其操作复杂耗时很长，其效率高低成为影响整个配送中心效率的关键因素。分拣环节也成为众多学者的研究方向。在提高分拣效率的同时，也面临业务扩大订单数量与商品种类增加，甚至超过货架数量。在不增加货架的前提下，如何规划以解决货架紧张问题并提高分拣效率成为电商公司配送中心提高效率所要解决的问题。

1.2 问题要求

根据以上背景信息，建立数学模型对以下具体问题进行讨论：

1. 将当日订单划分批次，在每个批次所包含的商品种类不超过 N （货架总数）的前提下，建立数学模型以批次数达到最小为目标，以使转运数量达到最小，提高效率。并通过附件 1 信息对所建立的模型算法进行应用，得到在 N 为 200 时的最小批次数以及每个批次的订单数量、商品的种类数量以及分批的方案等。
2. 对货品的拜访位置进行优化，以减小移动距离，提高分拣效率。在问题 1 的基础上，建立数学模型，以同一订单商品最集中为目标即以移动距离最短为目标，设计货品摆放位置。并在问题 1 分批结果的基础上，运用所建立的模型优化问题 1 的货品位置摆放。
3. 设计订单指派算法，在问题 1 和问题 2 的基础上将分拣任务交给 n 位分拣工人，以完成任务的速度位目标，同时尽量使得每位分拣工人的移动距离尽量均衡。通过前两个问题的结果，运用此模型算法，将分拣任务分配给 5 个分拣工人，确定各个批次的订单指派结果以及五位分拣工人的处理订单的顺序。

2、模型假设及符号说明

2.1 模型假设

- (1) 假设附件 1 提供的数据真实可靠；
- (2) 假设货架可以放置的货品数量不受限制（即货架的容积与载重为无限大）；
- (3) 假设每个货架只可以放置一种货物；
- (4) 假设每一个订单内所含货物的种类小于货架数量；
- (5) 假设受场地与成本限制，无法增加货架数量

2.2 符号说明

Symbol	Meaning
K	批次总数
X	商品总数
N	订单总数
M	货架总数
n	订单编号
i	商品编号
y_{nk}	批次 k 是否包含订单 n
u_{ik}	批次 k 是否包含商品 i
x_k	第 k 批次是否存在
a_{in}	订单 n 是否包含商品 i

3、分批算法的建立与求解

研究分批算法问题，假定任何一个批次的任何一个订单数量均不超过货架数量，寻求能够将货物成功分拣的最小批次，以减小运转次数，提高分拣效率。

3.1 数据预处理

3.1.1 数据探索

赛题所提供原始数据为 csv 格式，第一行为标题行，每行以英文逗号分隔，共三列：其中“OrderNo”表示订单编号，“ItemNo”表示货品编号，“Quantity”表示货品数量。

首先对订单及货物信息进行简单统计，统计结果为订单数量为 923，货物种类为 1941 类。其次，对每个订单中的货物种类进行统计，每个订单中的货物种类数量分别求和并排序，其分布如图 3-1，3-2 所示，货物种类最多的订单内共有 143 类货物，货物种类最少的订单内仅有 1 类货物，并且订单内货物种类数量

少于 10 类的订单占据订单总数的 1/4。

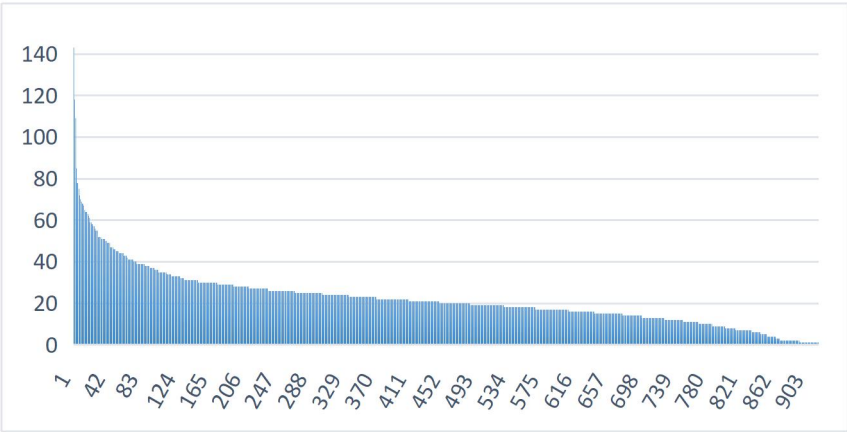


图 3-1 订单内货物种类数量分布柱状图

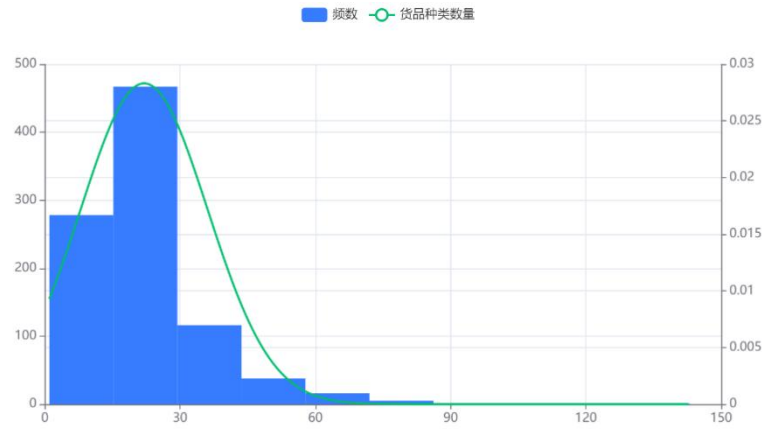


图 3-2 订单内货物种类数量分布直方图

3.1.2 数据筛选

所有订单中货物种类数量少于 2 类的订单数量在共 49 个，经观察发现订单内含有订单种类数量较多的订单对含有订单数量较少的订单之间含有包含关系，即某些含有货物种类较少的订单中的货物种类在其他订单中均同时出现，在此类问题中可以直接将这个货物种类被完全包含的订单与包含其的订单一同处理，降低后续模型的计算复杂度。

使用 Python 对含有包含关系的订单进行筛选，共发现 47 个订单内的货物被其他订单包含如表格 3.1 所示，将这些订单划分至包含其全部货物的订单中一同进行批次划分，即这 47 个订单在进行分批处理时可以忽略，经此步骤操作，降低分批算法 1/20 的数据量，在一定程度上降低了分批算法的求解难度。

表 3.1 被其他订单包含的订单

被包含订单	被包含订单	被包含订单	被包含订单
D0614	D0126	D0026	D0212
D0243	D0318	D0110	D0092
D0091	D0188	D0003	D0002
D0800	D0245	D0205	D0260
D0019	D0043	D0042	D0146
D0119	D0158	D0514	D0221
D0414	D0600	D0140	D0078
D0530	D0162	D0169	D0101
D0215	D0112	D0105	D0045
D0044	D0106	D0009	D0152
D0840	D0442	D0005	D0142
D0166	D0168	D0122	

3.2 基于 0-1 整数规划问题的分批算法模型

3.2.1 模型构建

通过问题解读结合运筹学中整数规划知识，我们参考了其他学者设计构造的分拣模型^[1-5]构建以下模型，其中 K 为目标变量， x_k 、 y_{nk} 、 u_{ik} 为决策变量， a_{in} 为已知变量，均为 0, 1 变量，其取值如下：

$$x_k \begin{cases} 0 & \text{存在第 } k \text{ 批次} \\ 1 & \text{不存在第 } k \text{ 批次} \end{cases}$$

x_k 表示是否存在第 k 个批次；

$$y_{nk} \begin{cases} 0 & \text{批次 } k \text{ 不包含订单 } n \\ 1 & \text{批次 } k \text{ 包含订单 } n \end{cases}$$

y_{nk} 表示在批次 k 中是否存在订单 n ；

$$u_{ik} \begin{cases} 0 & \text{批次 } k \text{ 不包含商品 } i \\ 1 & \text{批次 } k \text{ 包含商品 } i \end{cases}$$

u_{ik} 表示批次 k 中是否包含商品 i ；

$$a_{in} \begin{cases} 0 & \text{订单 } n \text{ 不包含商品 } i \\ 1 & \text{订单 } n \text{ 包含商品 } i \end{cases}$$

a_{in} 表示在订单 n 中是否需要商品 i 。其目标条件为批次最小，即 x_k 的累加和最小，即：

$$\min K = \sum_k x_k$$

达到目标函数所需要满足的约束如下所示：

$$a_{in} * y_{nk} \leq u_{ik}$$

表示在 a_{in} 与 y_{nk} 同时为 1 时， u_{ik} 必当为 1，反之则不成立，即订单 n 中包含商品 i 且批次 k 中包含订单 n ，则在该批次中必定包含商品 i 。

$$y_{nk} \leq x_k$$

表示当不存在第 k 批次时， y_{nk} 也就为 0；

$$\sum_k y_{nk} = 1$$

表示对于订单 n ，有且仅有一个批次对其进行处理；

$$\sum_i u_{ik} \leq M$$

表示每一批次的商品数量不能超过货架的总数。

综上所述，基于 0-1 整数规划问题的分批算法模型为：

$$\begin{aligned} \min K &= \sum_k x_k \\ \text{s. t. } &\begin{cases} a_{in} * y_{nk} \leq u_{ik} \\ y_{nk} \leq x_k \\ \sum_k y_{nk} = 1 \\ \sum_i u_{ik} \leq M \\ 1 \leq n \leq N \\ 1 \leq i \leq I \end{cases} \end{aligned}$$

3.2.2 模型求解

通过对原始数据集进行聚合处理，做成以订单编号为行，以货物种类编号为列的 923*1941 的矩阵，即 N 取值为 923， X 取值为 1941，由题目条件可知货架总数 M 为 200。

表 3.2 部分矩阵效果展示

	P0128	P0129	P0313	P0315	P0398	P0412

D0001	1	1	1	1	1	1
D0002	0	0	0	0	0	0
D0003	0	0	0	0	0	0
D0004	0	0	0	0	0	0
D0005	0	0	0	0	0	0
D0006	0	0	0	0	0	0
D0007	0	0	0	0	0	0
D0008	0	0	0	0	0	0
D0009	0	0	0	0	0	0
D0010	0	0	0	0	0	0

我们参考其他学者^{[3][4]}的方法，使用遗传算法进行求解，首先根据所建模型设置初始参数如图 3-3 所示，设置种群规模为 100，最大进化代数为 100，交叉概率为 0.5，求得最优目标函数值最少批次数为 96。

```

M = 1 # 初始化M (目标维数)
maxormins = [-1] # 初始化目标最小最大化标记列表, 1: min; -1: max
Dim = 3 # 初始化Dim (决策变量维数)
varTypes = [0] * Dim # 初始化决策变量类型, 0: 连续; 1: 离散
lb = [0, 0, 0] # 决策变量下界
ub = [1, 1, 1] # 决策变量上界
lbin = [1, 1, 1] # 决策变量下边界
ubin = [1, 1, 1] # 决策变量上边界

```

图 3-3 初始参数

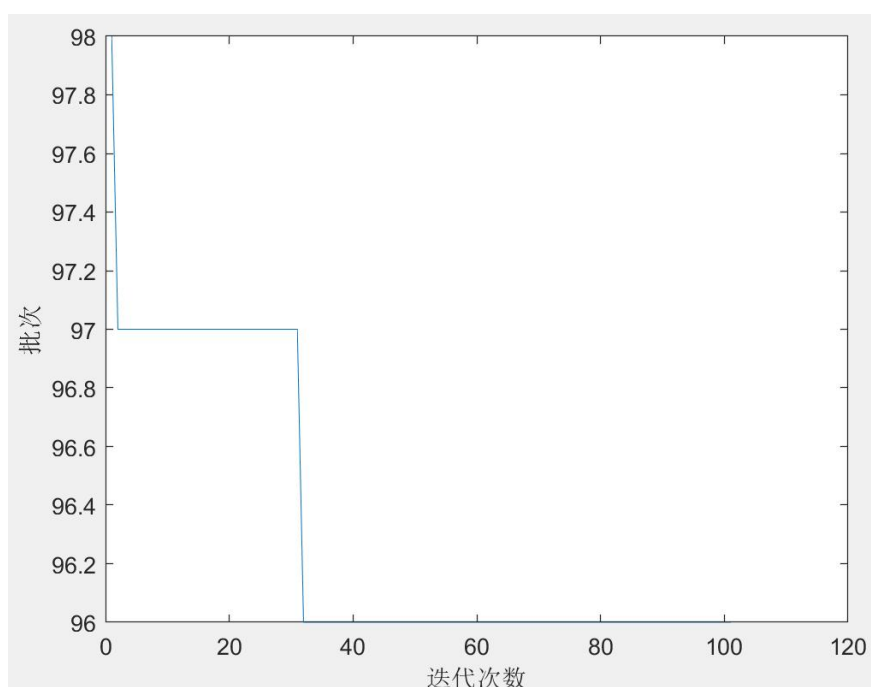


图 3-4 迭代结果

3.3 基于订单相似度的分批算法

为寻找更优的订单分批策略，本文又采用参考聚类思想^[6]的订单相似度分批算法模型。

3.3.1 模型构建

结合题目所提供信息，考虑可是使用聚类思想解决订单分批问题，首先将每个订单中所含物品种类转换成 0,1 矩阵，其中 1 代表该订单含有该类商品，0 表示该订单不含该类商品，初始考虑使用向量间的距离定义订单的相似程度，运用使用最为广泛的欧式距离作为距离度量方法，但实际操作过程中发现以下几点问题：

1. 欧式距离并不能度量订单的相似程度，如在本文 3.1 中提到的订单的包含问题无法使用欧式距离度量，例某订单 D1 的向量表示为：[1,0,1,1],另一订单 D2 向量表示为[1,0,0,0]，明显可以看出 D2 的订单货物种类完全包含于订单 D1 中，可以在不增加货物种类的情况下将 D2 与 D1 划分至同一批次，但是使用欧式距离度量 $d\{D1, D2\} = 2$ ，无法正确表达订单相似程度；

2. 聚类方式较难度量同批订单货物种类数量是否超出阈值；

基于以上两点问题，本文提出自己的订单相似性度量方式，首先选取订单 $D1 = [p1, p2, p3, p4]$ ， $D2 = [p2, p5]$ ，其中 D1 订单货物数量 \geq D2 订单货物数量，此时 D1, D2 的相似性为 $n(D1 \cup D2 - D1)/n(D2)$ ，即使用 D2 订单中与 D1 订单中货物种类不同的货物种类数量占比表示 D2 与 D1 的相似程度。

3.3.2 模型求解

具体求解过程如下：

模型输入：订单及货物信息，货物种类阈值 β

模型输出：分批信息

Step 1：将订单信息按照订单内货物种类数进行降序排列；

Step 2：选取所有订单内所含货物种类最多的订单开始分批，使用上述相似性度量方式依次计算其余所有订单与其相似程度并按照降序排列；

Step 3：依次选取与其相似程度最高的订单，计算将其分至同一批后该批次

内货物种类，若不超过阈值 β ，则分至同一批次，否则放弃，依次遍历其余所有订单；

Step 4：将分至同一批次的订单去掉后从步骤 1 开始循环直至所有订单被分批结束。

在赛题所给数据集中使用该算法所求得所有订单批次为 66，无论是速度还是精度都优于使用遗传算法解目标规划问题。

3.3.3 模型优化

在实际计算过程中发现以上模型中还存在一些问题，即仅考虑本文所提出的订单相似性度量，并未考虑货物种类较少订单的货物种类数量，如一个 $D1 = [p1, p2, p3, p4, p5, p6, p7]$, $D2 = [p1, p8]$, $D3 = [p1, p2, p5, p8, p9, p11]$ ，使用本文所提相似性度量方式计算订单 D2，D3 与 D1 的相似均为 0.5，但其贡献程度是不同的，因此考虑将其物品种类以一定的比例添加至订单相似性度量过程中。但这一比例比较难以确定，因此最终选择使用网格搜索方式选取最优比例，最终订单分批数量为 59，在 3.3.2 的基础上有了一定的进步。每批订单数量、货品种类数如表 3.3 所示，其效果如图 3-5 所示。

表 3.3 每批订单数量及货品种类结果

批次	订单数量	货品数量	批次	订单数量	货品数量
1	28	199	31	12	197
2	27	199	32	9	199
3	25	199	33	16	199
4	9	199	34	15	199
5	5	199	35	21	199
6	17	199	36	17	199
7	17	199	37	8	199
8	15	199	38	23	199
9	15	198	39	13	199
10	15	199	40	13	196
11	10	199	41	15	199
12	9	199	42	13	199
13	23	199	43	22	199
14	18	199	44	16	198
15	15	198	45	17	199
16	17	199	46	15	197
17	15	199	47	15	197
18	22	199	48	20	199

19	13	199	49	16	199
20	17	199	50	29	199
21	22	198	51	13	199
22	13	199	52	12	199
23	15	198	53	14	197
24	16	199	54	12	194
25	11	199	55	19	199
26	11	199	56	12	196
27	13	199	57	12	196
28	23	199	58	12	195
29	11	199	59	11	145
30	14	199			

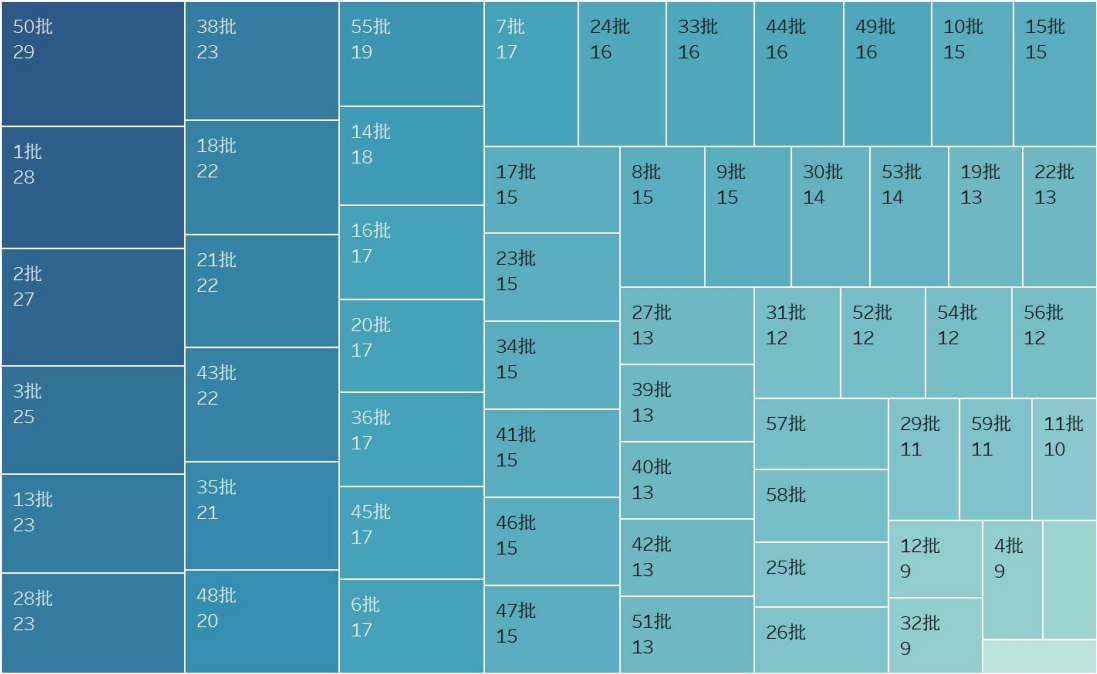


图 3-5 各个批次订单数量示意图

3.3.4 模型检验与分析

为了验证模型的实用性，本文对模型的敏感性进行了分析。在本问题中，货架受到了场地和成本的限制，不能够增加货架。但是在实际生活中，当订单量增加，公司规模变大，也会选择改变场地的规模，即在实际生活之中，货架数量并非固定的，为分析货架数量对本模型的影响，选择调整货架数量为 150、250 以及 300，通过本模型进行计算，得到结果与货架数量为 200 的模型对比如下表。

表 3.4 货架数量灵敏度分析

货架数量	批次
------	----

200	59
150	93
250	52
300	43

根据表 3.4 可得随着货架数量的增加,批次数减小,但批次数的减小速率随着货架数量的增加而减小。

4、优化货品摆放位置模型的建立与求解

针对问题二,寻找货品摆放的最佳位置,以使移动距离最短,提高挑选效率,这一问题可以抽象出类似于旅行商问题的模型。在本问题中,每个商品只能经历一个货架,定义订单 O_k 的挑选距离为该订单中货品所在货架序号的最大值减去最小值,即

$$d(O_k) = \max_{i \in O_k} S(i) - \min_{i \in O_k} S(i)$$

要求对某一分批任务设计货品的最佳摆放位置,其目标为对该批次进行拣选所需距离最小,即为最短路径问题求截。

4.1 优化货品摆放模型的构建

由上文可知,优化货品摆放模型可以抽象为类似于旅行商问题的模型,在此参考学者建立模型^[7-10],以该批次所需总距离最短为目标,同时对于本批次中每个商品对应一个货架。设 n' 为第 k 批次中的订单编号, i' 为第 k 批次中的货品种类编号,与第一问相同, a_{in} 表示第 n 个订单中是否有货品 i 。与最短路径相似,其目标函数为距离最短,在本文中,距离为本批次中所有订单所需挑拣距离的总和,即

$$D_k = \sum_{n'} d(n') = \sum_{O_k} d(O_k) = \sum_{n'} (\max_{i \in O_k} S(i) - \min_{i \in O_k} S(i))$$

其中, $\max_{i \in O_k} S(i)$ 为在订单 O_k 中货物所在货架的最大序号,用角标形式表示即:

$$\max_{i \in O_k} S(i) = \max\{S(i') | a_{i'n'} = 1\}$$

上式表示在当前批次,订单 n' 中包含货品 i' 时,采取货品所在货架的序号的最大值。

其中, $\min_{i \in O_k} S(i)$ 为在订单 O_k 中货物所在货架的最小序号, 用角标形式表示即:

$$\min_{i \in O_k} S(i) = \min\{S(i') | a_{i'n'} = 1\}$$

上式表示在当前批次, 订单 n' 中包含货品 i' 时, 采取货品所在货架的序号的最小值。

寻求上述目标函数的最小值时, 需要满足以下条件:

$$i' = \{i | u_{ki} = 1\}$$

即表示本问题中的 i' 为讨论批次中存在的商品编号。

$$n' = \{n | y_{nk} = 1\}$$

即表示本问题中的 n' 为讨论批次中包含的订单编号。

$$S(i) = 1, 2, 3, \dots, M$$

即表示货品所在货架的编号为 1 到 M 的整数。

$$S(i) \neq S(i + 1)$$

即表示不同货品的货架不同, 确定货品与货架的对应关系。

综上所述, 建立的优化货品摆放模型为:

$$\min D_k = \sum_{n'} d(n') = \sum_{n'} (\max\{S(i') | a_{i'n'} = 1\} - \min\{S(i') | a_{i'n'} = 1\})$$

$$s. t. \begin{cases} i' = \{i | u_{ki} = 1\} \\ n' = \{n | y_{nk} = 1\} \\ S(i) = 1, 2, 3, \dots, M \\ S(i) \neq S(i + 1) \end{cases}$$

4.2 优化货品摆放模型的求解

对于类似于旅行商问题的优化货品摆放模型的求解, 因其变量数量庞大且此算法为非线性, 因此选用启发式算法进行求解。在第一问的基础之上, 以第一批次为例, 通过模拟退火算法确定各个货物放置货架的序号。

由第一问的结果可知, 第 1 批次包含的订单数如表 4.1 所示。通过分析附件一可得这 28 个订单包含的商品数量为 199 具体种类如表 4.2 所示。因此对于批次 1 来说, 其目标就是将这 199 种货品放置到 200 个货架上, 并使得拣选距离最小, 即这 28 个订单拣选所需要的距离之和最小。

表 4.1 批次一包含的订单

订单	订单	订单	订单
D0912	D0726	D0294	D0096
D0601	D0497	D0391	D0148
D0630	D0776	D0539	D0614
D0352	D0337	D0384	D0126
D0613	D0903	D0496	D0026
D0674	D0423	D0312	D0212

表 4.2 批次一种的货品种类

货品种类	货品种类	货品种类	货品种类
P0412	P0414	P0411	P0707
P0817	P0852	P0944	P0945
P0947	P1017	P1018	P1020
P1091	P1685	P0818	P0856
P0909	P0911	P0097	P0167
P0191	P0399	P0981	P1102
P1794	P0209	P0302	P0303
P0306	P0307	P0701	P0853
P0854	P1591	P1596	P1044
P1079	P1588	P0379	P0449
P0879	P0880	P0920	P0922
P1096	P1097	P1098	P1099
P1105	P1628	P0728	P0948
P0949	P1056	P0211	P0550
P0875	P1834	P0415	P0531
P0849	P1571	P0708	P0709
P0710	P0714	P0943	P0063
P0201	P0311	P0757	P0305
P1757	P0065	P0141	P0684
P0881	P1046	P0640	P0878
P1054	P1055	P0769	P1011
P0713	P0404	P1111	P1645
P0842	P0434	P1045	P1112
P1113	P0106	P1100	P0194
P0874	P1213	P1214	P1711
P1264	P0146	P0251	P0252
P0791	P1090	P0053	P0798
P1124	P0165	P1286	P0046
P0046	P0840	P1280	P1115
P0941	P1356	P1154	P1199
P1201	P1281	P1203	P1271
P1288	P1235	P1255	P1257
P0374	P1252	P1253	P0910
P0888	P1006	P1263	P1363
P1242	P1308	P1316	P1251

P1282	P1312	P1254	P1314
P1248	P1189	P0785	P1343
P1344	P1302	P1346	P1362
P0200	P1176	P1351	P1352
P1354	P0312	P0457	P0076
P0077	P0078	P0364	P0365
P0764	P0765	P0787	P0788
P0790	P0857	P0950	P1155
P1157	P1187	P1223	P1224
P1231	P1233	P1266	P1267
P1310	P1313	P1315	P1409
P1699	P1700	P1708	P1709
P1740	P1833	P1317	P1374
P0936	P0938	P1445	P1488
P0044	P0045	P0047	P1535

在计算时，模拟退火的各项参数如下所示：

表 4.3 模拟退火参数

参数	数值
alpha	0.99
t	(1,100)
markovlen	1000

其中，alpha 表示降温系数，t 为温度设置，在本文中，将退火的初始温度设置为 100，终止温度设置为 1，markovlen 表示迭代次数。以最优货品摆放为目标，所得第 1 批次的部分结果如表 4.4 所示：

表 4.4 第 1 批部分结果

货品种类	货架序号	货品种类	货架序号	货品种类	货架序号	货品种类	货架序号
P0412	1	P1154	16	P1288	31	P1591	46
P0874	2	P0911	17	P0191	32	P1286	47
P1356	3	P1099	18	P0798	33	P0948	48
P1100	4	P1176	19	P1045	34	P1224	49
P0251	5	P0849	20	P0713	35	P1409	50
P0167	6	P1091	21	P0457	36	P1351	51
P0411	7	P0922	22	P0941	37	P1280	52
P0765	8	P0414	23	P0201	38	P0728	53
P0044	9	P0791	24	P0141	39	P1266	54
P1267	10	P1344	25	P1111	40	P0707	55
P0211	11	P1264	26	P1113	41	P0399	56
P0981	12	P1834	27	P1374	42	P1214	57

P0449	13	P1700	28	P1203	43	P0076	58
P1488	14	P1199	29	P0910	44	P1699	59
P1054	15	P1046	30	P0550	45	P1317	60

其他批次与批次 1 相同，均采用模拟退火算法进行求解，参数设置如表 4.3 所示。最终所有批次距离总和及各个批次距离总和结果如表 4.5 所示。效果如图 4-1 所示。

表 4.5 所有批次的距离总和及各个批次距离和

批次	批次总距离	批次	批次总距离	批次	批次总距离
1	2193	21	1697	41	1159
2	1842	22	866	42	937
3	1907	23	1147	43	1703
4	496	24	1262	44	1275
5	407	25	860	45	1490
6	1147	26	654	46	1147
7	1063	27	827	47	841
8	882	28	1855	48	1750
9	918	29	682	49	1211
10	906	30	1159	50	1904
11	554	31	734	51	668
12	590	32	600	52	774
13	1977	33	1437	53	944
14	1093	34	766	54	765
15	722	35	1593	55	1665
16	931	36	1325	56	706
17	930	37	614	57	770
18	1739	38	2131	58	631
19	849	39	801	59	287
20	1333	40	843	总批次	64959

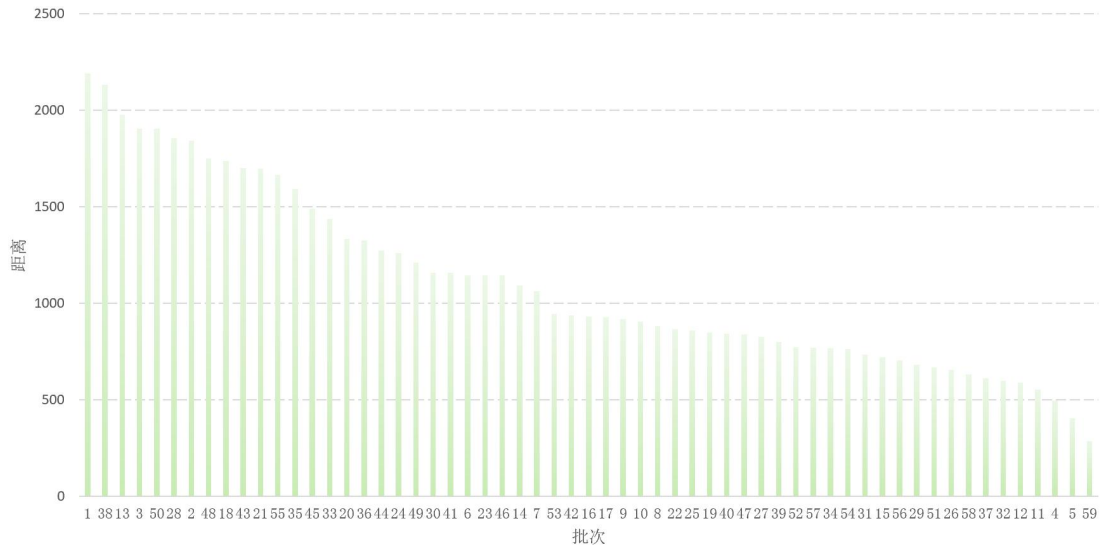


图 4-1 各个批次距离总和示意图

5、指派分拣任务模型的建立与求解

研究各个批次下将订单分配给多个分拣工人完成分拣任务的指派问题。针对此问题，首先由题目的假设条件可知研究的目标为一个批次的订单的指派问题而非所以批次的综合指派问题，即不同批次指派问题的解决是互不影响的。其次，假设各个分拣工的速度为匀速运动且大小相同，同时将包括放置拣货筐时间、取用拣货筐时间、取货时间、切换订单时间以及休息时间等忽略，因此可以将目标从时间最短转变为路径最短。

5.1 最佳指派分拣任务模型的构建

由前文可知，此指派分拣任务不同批次互不影响，可以仅考虑一个批次。因此，在构建模型时，参考其他学者建立的模型^[10-12]，以第 k 批次为例进行构建。设有 R 个分拣工人，第 k 批次的订单数目为 N' 。目标函数为最小总路径以及各个分拣工运动距离均衡，但若以这两个条件作为目标约束，模型求解为 NP 难题，因此，本文将这两个问题转化为求最小化 R 个分拣工运动最大距离，即能保证所需时间最小，又能使得各个分拣工的运动距离尽量均衡。

分拣工人的运动距离可以分为三部分，即完成一个订单时订单内货物简单距离以及到达此订单第一个货物的距离以及最后完成这一批次所有订单后返回 1 号货架的距离。分拣工人的最大运动距离为：

$$d(r)_{max} = \max(g_{n'r} * d_{n'r} + d'_{n'r} + d''_{n'r})$$

其中：

$$g_{n'r} = \begin{cases} 1 & \text{将订单 } n' \text{ 分配给工人 } r \\ 0 & \text{将订单 } n' \text{ 不分配给工人 } r \end{cases}$$

$g_{n'r}$ 表示是否将第 n' 个订单分配给工人 r ，若分配则取值为 1，否则取值为 0。

$$d_{n'r} = \max\{S(i')|a_{i'n'}\} - \min\{S(i')|a_{i'n'}\}$$

$d_{n'r}$ 表示工人 r 完成第 n' 个订单所需要的距离，假设货架尺寸不考虑并且货架距离为 1，因此货架距离为货架序号相减，完成一个订单所需距离也就为订单序号最大减去最小。

$$d'_{n'r} = \begin{cases} |S(i)_{max}^{n'r} - S(i)_{max}^{n'r-1}|, n^r \text{ 为偶数} \\ |S(i)_{min}^{n'r} - S(i)_{min}^{n'r-1}|, n^r \text{ 为奇数} \end{cases}$$

$d'_{n'r}$ 表示工人 r 到达第 n' 个订单第一个货物的距离，即上一个订单的最后一个货物所在货架与本订单第一个货物所在货架的距离。在此选取适用于本题的 S 型策略，即将订单中的货物按照所在货架的序号排列，完成上一个订单后选择从与上一个订单最后一个货物所在货架较近的一端开始拣货。

$$d''_{n'r} = S(i)_r$$

$d''_{n'r}$ 为最后完成这一批次所有订单后返回 1 号货架的距离。 $S(i)_r$ 为第 r 个分拣工人最后分拣货物所在货架的序号。

此目标条件需要满足的约束分别为下面两个公式：

$$\sum_{n'} \sum_r g_{n'r} = N'$$

上述约束条件表示此批次共有 N' 个订单。

$$\sum_r g_{n'r} = 1$$

表示每个订单有且仅有一个分拣工完成。

综上所述，以是否将第 n' 个订单分配给工人 r 为决策变量，建立指派分拣任务模型如下：

$$\min d(r)_{max} = \max(g_{n'r} * d_{n'r} + d'_{n'r} + d''_{n'r})$$

$$s. t. \begin{cases} \sum_{n'} \sum_r g_{n'r} = N' \\ \sum_r g_{n'r} = 1 \end{cases}$$

5.2 最佳指派分拣任务模型的求解

对于标准指派问题的求解，匈牙利解法充分利用了指派问题是 0-1 整数规划问题又是特殊运输问题的特殊性质，减小了此类问题的计算量。但在本模型中，首先其目标函数较为复杂，其次决策变量较多，数据量较大，采用匈牙利解法不再适合，因此本文选用了启发式算法对本模型进行求解。与问题二类似，以第 1 批次为例，运用模拟退火算法对最佳指派分拣任务模型进行求解，模拟退火的参数与问题二相同，如表 4.3 所示。由第二问结果可知，第 1 批次货品所在货架标号，在此基础上对五位分拣工人的分配订单任务进行求解，求解结果如表 5.1 所示。

表 5.1 第一批次工人分配订单

分拣工人	订单
1	D0497,D0246,D0148
2	D0026,D0613,D0126,D0096,D0496,D0776,D0391,D0243
3	D0601,D0352,D0581,D0423
4	D0674,D0539,D0903,D0212,D0912,D0726
5	D0337,D0630,D0384,D0312,D0614,D0294,D0747

其他批次与批次 1 相同，均采用模拟退火算法进行求解，参数设置如表 4.3 所示。

6、模型的优缺点

6.1 模型优点分析

1. 本文首先对附件数据进行预处理，降低了数据量，减小了计算的复杂度。
2. 在本文中，采用了多种模型对比的方法，进一步优化了问题的结果，且针对货架数量进行了灵敏度分析，使得批次数量更小。
3. 任务一的基于订单相似性的聚类分析模型计算速度快，结果较好。

6.2 模型缺点分析

1. 本文任务一的基于订单相似性的聚类分析模型结果虽然较好，但可能为局部最优而非全局最优。
2. 模型较为复杂，数据量大，所需计算时间较长。

参考文献

- [1] Xiang X, Liu C, Miao L. Storage assignment and order batching problem in Kiva mobile fulfilment system[J]. *Engineering Optimization*, 2018, 50(11): 1941-1962.
- [2] 陈方宇. 多区块仓库环境下订单拣选路线规划研究[D]. 华中科技大学, 2014.
- [3] Cergibozan Ç, Tasan A S. Genetic algorithm based approaches to solve the order batching problem and a case study in a distribution center[J]. *Journal of Intelligent Manufacturing*, 2020: 1-13.
- [4] 韩玉芳. 配送中心订单分批问题研究[D]. 山东大学, 2015.
- [5] 黄敏芳, 张源凯, 王颜新, 等. 基于 JIT 装配模式的网上超市订单分拣优化模型[J]. *中国管理科学*, 2020, 28(5): 159-166.
- [6] 秦馨, 赵剑道, 任楠. 基于聚类算法的订单分批策略研究[J]. *制造业自动化*, 2021, 43(01): 134-137.
- [7] Chen M C, Huang C L, Chen K Y, et al. Aggregation of orders in distribution centers using data mining[J]. *Expert systems with applications*, 2005, 28(3): 453-460.
- [8] Henn S, Schmid V. Metaheuristics for order batching and sequencing in manual order picking systems[J]. *Computers & Industrial Engineering*, 2013, 66(2): 338-351.
- [9] Koch S, Wäscher G. A grouping genetic algorithm for the order batching problem in distribution warehouses[J]. *Journal of Business Economics*, 2016, 86(1): 131-153.
- [10] 陈方宇. 多区块仓库环境下订单拣选路线规划研究[D]. 华中科技大学, 2014.
- [11] 王占磊. 配送中心订单分批及拣选路径优化问题研究[D]. 吉林大学, 2013.
- [12] 杜亚江, 王娴. 物流中心高层货架拣选作业的路径优化[J]. *计算机系统应用*, 2013(9): 160-163.