

队伍编号	205218
题号	C

基于模拟退火遗传算法求解

仓内拣货问题的混合 0-1 整数规划模型

摘 要

本文针对仓内拣货任务的规划问题，需要通过规划货格访问顺序、人员执行任务以及任务执行顺序，提高系统的整体效率。

针对问题一要求在考虑实际障碍物的前提下，计算 3000 个货格与 13 复核台共 3013 个元素之间的距离。本文首先对仓库进行分区，基于曼哈顿距离并考虑纵、横向的偏移距离，分别讨论处于相同区和不同区货格之间的计算方法，在此基础上提出了各元素之间的距离计算方法，最终得到了相应的距离。

针对问题二要求规划出拣货员负责一个任务时的最优拣货路线，本文将任务单中的每个货格视为访问点，建立了旅行商模型，并采用模拟退火算法求解。得到了最理想的货格访问顺序，并且计算得到完成出库的时间为 466.67 秒返回的复核台为FH10。

针对问题三要求得到单拣货员多拣货台多任务的最优拣货路线。为缩小解空间，本文对问题条件下不存在囤货现象进行了证明，并以指派问题的模型为基础，建立了 0-1 整数规划模型。在求解过程中，首先基于问题二中的方法计算出各任务单各情况下的最优距离，之后通过 LINGO 求解。计算得到了最理想的任务执行顺序与货格访问顺序，最后得到完全出库时间为 2245.06 秒，复核台FH03和FH11的利用率分别为 2.67%和 4.01%。

针对问题四要求规划多拣货员多拣货台多任务的最优拣货路线，本文考虑到复核台货物囤积、每个拣货员每次只能执行一个任务等现实约束，并将其中的分段函数转化为线性表达式，建立了混合 0-1 整数规划模型。接下来充分结合解的形式，设计了模拟退火遗传算法对模型进行求解，最终得到了每位员工的理想任务执行顺序和货格访问顺序。货物完全出库的时间为 2579.7 秒，复核台FH01,FH03,FH10和FH12的利用率分别为 15.14%,16.24%,10.61%以及 15.02%。

针对问题五要求评估增加复核台对完全出库时间带来的影响，本文采用模拟退火遗传算法对每一种增添复核台的情况的进行计算。最后的计算结果显示增添的复核台与最初的四个复核台距离越远，完全出库时间减少的就越明显。

针对问题六要求权衡畅销品放置于离拣货台更近所带来的距离收益和堵塞损耗，本文提议将热销商品放置于面向外围和靠近横向通道的位置，以减少拣货员在通道堵塞的可能，并根据储位指派法则给出了商品摆放的相关建议。

本文主要通过启发式算法对模型进行求解，在面对规模较大的问题时，启发式算法能够大幅度提升求解速度，提高问题处理的效率，更符合柔性物流系统的要求。此外，还通过数据可视化技术对计算的过程与结果进行了呈现，更为直观。

关键词：混合 0-1 整数规划模型；模拟退火遗传算法；模拟退火算法；

目录

1 问题重述	1
2 模型假设	2
3 符号说明	2
4 模型建立与求解	3
4.1 问题一	3
4.1.1 问题分析	3
4.1.2 数据分析与处理	3
4.2 问题二	5
4.2.1 问题分析	5
4.2.2 模型构建	5
4.2.3 模型求解与结果分析	6
4.3 问题三	7
4.3.1 问题分析	7
4.3.2 模型构建	8
4.3.3 模型求解与结果分析	9
4.4 问题四	10
4.4.1 问题分析	10
4.4.2 模型构建	10
4.4.3 模型求解	11
4.4.4 结果分析	12
4.5 问题五	13
4.5.1 问题分析	13
4.5.2 模型求解和结果分析	13
4.6 问题六	14
5 模型评价	14
5.1 模型优点	14
5.2 模型缺点	14
参考文献	15

1 问题重述

本问题是针对仓内拣货的一个规划问题，需要对拣货过程中的各个要素进行分析，提高物流拣选系统的整体效率。

问题一需要根据问题中给出的仓库布局，考虑到货架对道路的阻挡，给出实际仓内货格与货格、货格与复核台以及复核台与复核台之间距离的计算方法。

问题二需要对一个任务单内的货格访问顺序进行排序，以得到最短的完全出库时间。

问题三在问题二的基础上增添至两个复核台和五个任务，并要求计算得到最短完全出库的时间为目标下任务的完成顺序、货格访问顺序以及各复核台的利用率。

问题四在问题三的基础上增加了拣货员的数量，此时需要充分考虑到复核台货物囤积造成时间的浪费等实际约束，提出任务的分配方式、货格的访问顺序等方案对最理想的安排方式进行描述，并计算得到完全出库的时间和复核台的利用率。

问题五则在问题四的基础上，要求评估当增添一个复核台时，对完全出库时间带来的影响。本问题需要关注的是不同位置复核台的增添对完全出库的时间影响不同，因此需要分别进行讨论。

问题六则是需要权衡在畅销品放置于邻近复核台所带来的距离收益和拣货员堵塞造成时间上的浪费，并提出通过改良商品摆放位置减少堵塞的方案。

2 模型假设

1. 拣货员下架商品时，处于货格的面向通道一侧的正中位置。
2. 在问题二、三、四中不考虑拣货员及相关搬运设备的体积。
3. 假设拣货员的行走速度不因搬运货物的重量而降低。
4. 假设仓库内不含有其它可能会作为障碍物的设备。

3 符号说明

符号	意义
x	横轴坐标
y	纵轴坐标
d_{ij}	元素 i 到元素 j 的距离
Z	总距离
x_{ij}	是否从 i 元素到 j 元素
l_c	货格 c 到所属货架上下边界的最小值
l_d	货格 d 到所属货架上下边界的最小值
T	出库时间
μ_l	可用复核台中顺序为 l 的复核台的利用率
$e_{pijk_1k_2}$	拣货员 p 完成的第 i 个任务为订单 j 且从复核台 k_1 出发返回复核台 k_2 后，物品等待复核台 k_2 恢复空闲的时间

4 模型建立与求解

4.1 问题一

4.1.1 问题分析

由于仓库内货架的阻挡，在不同巷道间穿梭时，往往需要绕行。因此，两个货格之间的距离用坐标直接计算欧几里得距离是不合适的。为此，需要采用合适的距离计算方法。需要注意到，仓内货架的摆放方式比较特殊，距离的计算方法应尽量利用这一特点。

4.1.2 数据分析与处理

根据仓内货架摆放的方式，观察到通道仅有纵横两个方向。因此尝试使用曼哈顿距离进行计算。同时，将仓库以排为单位进行分区[1]，以此将货格之间的位置关系分为同区与异区，分别提出不同的距离计算方法。另外，还需要考虑复核台与货格、复核台与复核台之间的距离。

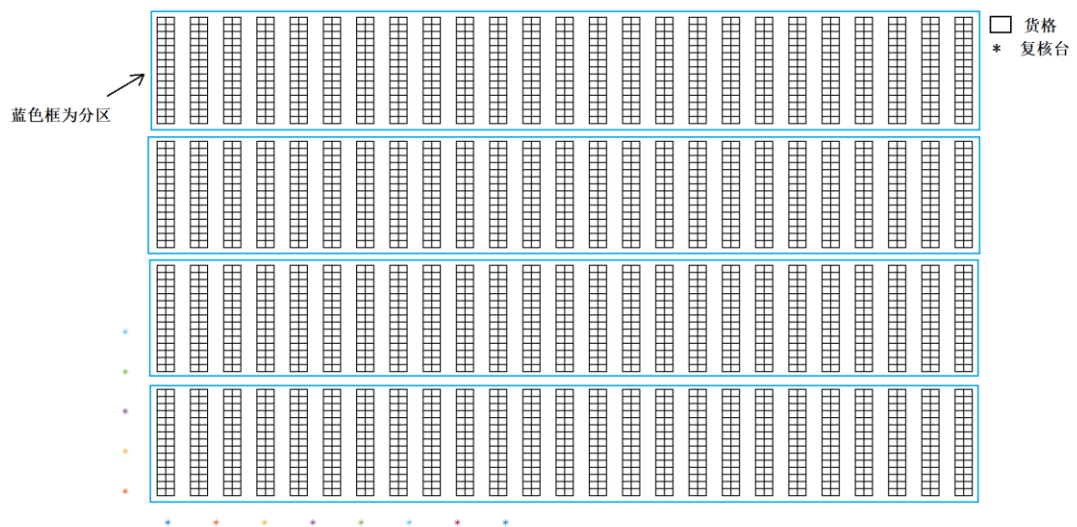


图1 仓库分区

在具体计算距离之前，首先要对货格的坐标进行处理。根据题目中的标示，两货格之间的距离是指由某个货格面向通道一侧边上的中点，在保持纵向、横向偏移距离都为750mm的条件下，避开障碍物，到达另一个货格面向通道一侧边上的中点。因此，对于下架通道在左边的货格，其坐标修正公式为：

$$x = x_0 - 750 \quad (1)$$

$$y = y_0 + 400 \quad (2)$$

对于下架通道在右边的货格，其坐标修正公式为：

$$x = x_0 + 1150 \quad (3)$$

$$y = y_0 + 400 \quad (4)$$

其中， x_0 和 y_0 分别为数据集中提供的货格坐标， x 和 y 分别为修正之后的货格坐标。

(1) 位于不同区的两个货格

对于不同区的货格 a 和货格 b ，现计算两者之间的距离。首先需要起点和中点分别向通道方向偏移 750mm，之后按照曼哈顿距离计算偏移后起点和终点之间的距离，具体公式如下：

$$d = 750 * 2 + |x_a - x_b| + |y_a - y_b| \quad (5)$$

其中， x_a 、 y_a 分别为货格 a 的横纵坐标， x_b 和 y_b 分别为货格 b 的横纵坐标。

(2) 位于相同区的两个货格

对于相同区但不在同一个纵向巷道两侧货格 c 和货格 d ，除了需要先向通道横向偏移 750mm 外，还需要额外加上绕行的距离。在额外绕行距离的计算时，需要对不同的路径进行比较并取最小值。同时，在转弯的时候仍需保持偏移，所以还需要额外计算纵向的偏移距离 750mm，具体公式如下：

$$d = 750 * 4 + |x_c - x_d| + |y_c - y_d| + 2 * \min(l_d, l_c) \quad (6)$$

$$l_c = \min(|y_{hc} - y_c|, |y_{hc} + 12000 - y_c|) \quad (7)$$

$$l_d = \min(|y_{hd} - y_d|, |y_{hd} + 12000 - y_d|) \quad (8)$$

其中各变量的实际意义如下图所示：

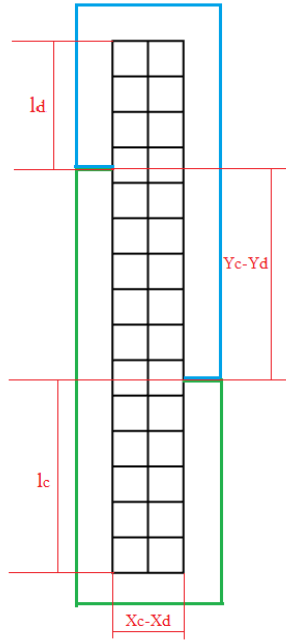


图 2 变量含义示意图

当两货格位于相同区但在同一个纵向巷道两侧或同侧，两者之间的距离则按如下方法计算：

$$d = 750 * 2 + |y_c - y_d| \quad (9)$$

(3) 复核台与货格之间的距离

根据题目的说明，复核台与货格之间的距离简化为货格中点到复核台最近一条边中点的距离。在具体计算时，首先需要复核台向货架方向移至偏移距离 750mm 处，之后计算此点与货格之间的实际距离。因此，仍需要考虑此点与目

标货格是否处于同一区。对于复核台 e 与货格 f ，具体计算公式如下：

$$d = 750 + |x_e - x_f| + |y_e - y_f| + l \quad (10)$$

若处于相同区且复核台位于货架区域下方时：

$$l = 2 * \min(l_e, l_f) \quad (11)$$

$$l_e = \min(|y_{he} - y_e|, |y_{he} + 12000 - y_e|) \quad (12)$$

$$l_f = \min(|y_{hf} - y_f|, |y_{hf} + 12000 - y_f|) \quad (13)$$

若处于相同区且复核台位于货架区域左侧时：

$$l = 2 * \min(l_e, l_f) + 2 * 750 \quad (14)$$

$$l_e = \min(|y_{he} - y_e|, |y_{he} + 12000 - y_e|) \quad (15)$$

$$l_f = \min(|y_{hf} - y_f|, |y_{hf} + 12000 - y_f|) \quad (16)$$

若处于不同区：

$$l = 0 \quad (17)$$

至此，计算得到 3000 个货格与 13 个复核台共 3013 个元素之间的距离。

表 4-1 各元素间距离矩阵

	S00101	S00102	...	FH12	FH13
S00101	0	2300	...	23800	19300
S00102	2300	0	...	24600	20100
...
FH12	16100	15300	...	0	4500
FH13	20600	19800	...	4500	0

完整的计算结果已填入附件表中。

4.2 问题二

4.2.1 问题分析

该问题需要规划出拣货员P在复核台FH10处获取任务T0001后，完成该任务并回到复核台的最理想路线。观察到拣货员P需要经过每一个任务T0001中所包含的货格，最终回到的复核台由最后访问的货格决定。由此可以把本问题认为是旅行商问题（TSP，Traveling Salesman Problem）变体。

4.2.2 模型构建

因为复核台复核打包的时间，以及各货格取货的时间已经固定，所以只需要获取最短路线便可达成目标，依此建立寻求最短路程的目标函数：

$$\min Z = \sum_{i=1}^N \sum_{j=1}^N d_{ij} x_{ij} \quad (18)$$

其中， d_{ij} 为从元素 i 到元素 j 之间的距离， N 为任务T0001中要求经过的货格总数。而 x_{ij} 表示是否经过了由元素 i 到元素 j 的路程的 0-1 变量，即：

$$x_{ij} = \begin{cases} 1, & \text{经过从} i \text{到} j \text{的道路} \\ 0, & \text{未经过从} i \text{到} j \text{的道路} \end{cases} \quad (19)$$

约束条件为：

① 每个T0001任务所要求的货格仅到达一次

$$\sum_{j=1}^N x_{ij} = 1, \quad i = 1, 2, \dots, N \quad (20)$$

② 每个T0001任务所要求的货格仅离开一次

$$\sum_{i=1}^N x_{ij} = 1, \quad j = 1, 2, \dots, N \quad (21)$$

③ 破解多个子环的情况，使其称为一个环

$$u_i - u_j + (n - 1) * n_{ij} \leq n - 2, \quad i, j = 2, 3, \dots, n \quad (22)$$

其中， $u_i \geq 0, u_j \geq 0$.

计算得到最短路径之后，将其转化为花费的总时间：

$$t_q = \begin{cases} 5 * y_q, & y_q < 3 \\ 4 * y_q, & y_q \geq 3 \end{cases}, \quad q = 1, 2, \dots, N \quad (23)$$

$$T = \sum_{q=1}^N t_q + Z / (1.5 * 1000) \quad (24)$$

其中， y_q 表示拣货员在 i 货格下架的货物数量。

4.2.3 模型求解与结果分析

由于问题规模较大，本文采用启发式算法中的模拟退火算法对模型进行求解。模拟退火算法一种被广泛应用的启发式算法，其在运算速度上有着较好的表现，同时还能够在一定程度上防止陷入局部最优解。其算法思想简述如下：对于某一个初始可行解，生成一个随机的邻解不断寻优，同时根据 Metropolis 准则，有限度的接受恶化解。

Metropolis 准则可以表示如下：

$$p = \begin{cases} 1, & \text{if } E(x_{new}) < E(x_{old}) \\ \exp\left(-\frac{E(x_{new}) - E(x_{old})}{E(x_{old})}\right), & \text{if } E(x_{new}) \geq E(x_{old}) \end{cases} \quad (25)$$

式中 p 表示接受新状态的概率， $E(x)$ 表示某一状态下的能量。本文使用了 MATLAB 对模拟退火算法进行了实现，并根据模型求解得出了结果。

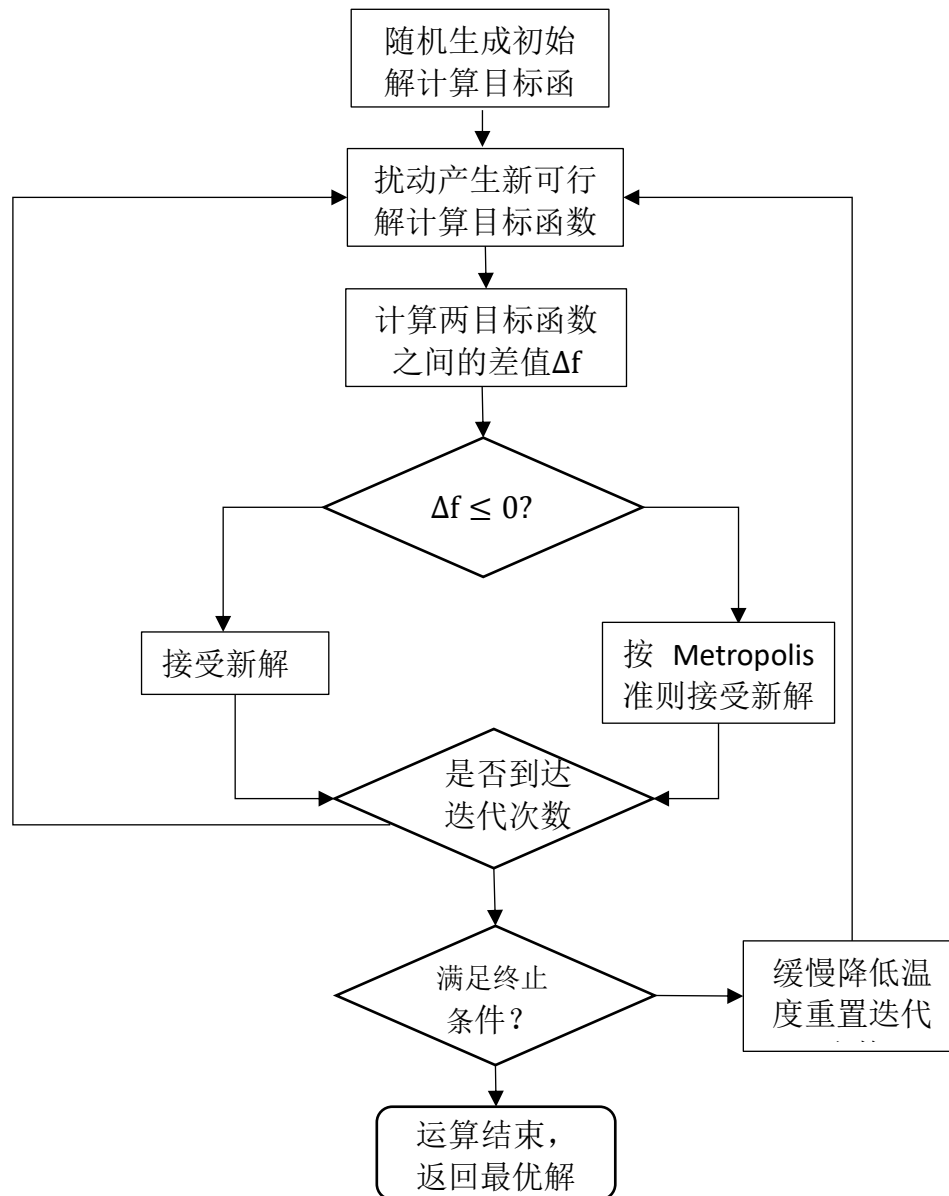


图 3 模拟退火算法流程图

经过上述计算，得到了完全出库时间 466.67 秒，最后回到的复核台为FH10，并得到了最理想路线：

FH10->S07305->S10508->S10501->S12103->S13004->S13809->S13812->S14510->S14908->S14401->S13509->S15911->S12608->S11205->S11106->S10115->S07212->S06213->S07515->S08502->S01308->S01713->S00107->FH10

4.3 问题三

4.3.1 问题分析

该问题在上一题的基础上，将复核台增加至 2 个（FH03,FH11），并且任务增加至 5 个（T0002 – T0006），拣货员数量不变，目标仍为求出最理想的拣货

路线，本质上是一个指派问题的变体。

由于增加了复核台，每个任务完成的时间需要分别考虑以FH03和FH11作为该任务起点和终点并自由组合的4种情况，另外还需要考虑到每次只能领取一个任务以及每个任务只能完成一次的约束。

4.3.2 模型构建

为缩小解空间，首先证明不存在货物囤积在复核台的情况：

由于仅存在一个拣货员，如果搬运任务时间的下界仍大于复核台复合货物的30秒，则可以认为不存在货物囤积的时间。通过遍历所有的任务搬运距离得到下界为 $\underline{c} = 366000\text{mm}$ ，其对应时间 $244\text{s} > 30\text{s}$ ，所以后续在建模时可以不考虑货物在复核台囤积。

已知货物复核打包的时间和各货格取货时间的综合为固定值，所以仍然只需要找到最短的搬运距离即可，由此给出目标函数：

$$\min Z = \sum_{i=1}^5 \sum_{j=2}^6 \sum_{k_1=1}^2 \sum_{k_2=1}^2 d_{jk_1k_2} x_{ijk_1k_2} \quad (26)$$

其中， $d_{jk_1k_2}$ 表示从从复核台 k_1 到 k_2 完成任务 j 所需要的事件，即每个任务在不同起点和终点情况下对应的第二问中的最短路程 Z 。 $x_{ijk_1k_2}$ 表示拣货员是否将任务 j 作为其第 i 个任务，并且从 k_1 复核台出发，将货物送至到 k_2 。

$$x_{ijk_1k_2} = \begin{cases} 1, & \text{该事件发生} \\ 0, & \text{该事件未发生} \end{cases} \quad (27)$$

约束如下：

① 拣货员每次只能进行一个任务

$$\sum_{j=1}^6 \sum_{k_1=1}^2 \sum_{k_2=1}^2 x_{ijk_1k_2} = 1, \quad i = 1, 2, \dots, 5 \quad (28)$$

其中，记 $k_1 = 1$ 表示起点为复核台FH03， $k_1 = 2$ 表示终点为复核台FH11。

② 每个任务只能完成一次

$$\sum_{i=1}^6 \sum_{k_1=1}^2 \sum_{k_2=1}^2 x_{ijk_1k_2} = 1, \quad j = 2, 3, \dots, 6 \quad (29)$$

③ 每个任务的起点与上个任务的终点相同

显然，若需要先从其中一个复核台移至另一个复核台再开始任务才能产生最优解，则在计算任务单最优访问顺序的时候已经考虑到，因此为了减少运算量，在此增添上一个任务的终点与下一个任务的起点相同的约束。

$$\sum_{j=1}^6 \sum_{k_1=1}^2 x_{i+1jk_2} = \sum_{j=1}^N \sum_{k_2=2}^2 x_{ijk_1p}, \quad i = 1, 2, \dots, 5, \quad k_1, k_2, p = 1, 2 \quad (30)$$

④ 第一个任务的起点必为复核台FH03

$$\sum_{j=1}^6 \sum_{k_2=1}^2 x_{1j1k_2} = 1 \quad (31)$$

至此，得到了最优的安排路线，将其换算为出库事件以及复核台利用率：

$$t_q = \begin{cases} 5 * y_q, & y_q < 3 \\ 4 * y_q, & y_q \geq 3 \end{cases}, \quad q = 1, 2, \dots, N \quad (32)$$

$$T_j = \sum_{q=1}^N t_q + Z_j / (1.5 * 1000) \quad (33)$$

由于不存在囤货情况，最终出库时间 T_f 为：

$$T_f = \sum_{j=2}^6 T_j + 30 \quad (34)$$

最后计算复核台利用率：

$$\mu_l = \frac{(\sum_{i=1}^5 \sum_{j=2}^6 \sum_{k_1=1}^2 x_{ijk_1l} * 30)}{T_f}, \quad l = 1, 2 \quad (35)$$

4.3.3 模型求解与结果分析

首先，根据第二问的方式，将任务每一种起点和终点的组合都进行计算，得到反映所有任务搬运距离的矩阵：

表 4-1 距离矩阵

起点-终点 任务号	3-3	3-11	11-3	11-11
T0002	464.7	382	427.7	366.2
T0003	451.5	375.6	424.4	366
T0004	445.1	375.6	424.4	369.2
T0005	446.6	395.3	439.3	375.4
T0006	464.7	382	427.7	366.2

然后将上述数据以及约束输入 LINGO 求解，得到了最优的安排方式：

T0003->T0006->T0004->T0002->T0005

复核台FH03和FH11的利用率分别为 2.67%和 4.01%。

详细的安排方式已经写入附件表中。

4.4 问题四

4.4.1 问题分析

本问题在问题三的基础上，增加了复核台、任务单的数量，同时拣货员由一个变为多个，目标仍为求解最理想路线。由于人员数量变多，出现了多人到达同一个复核台导致货物囤积的现象。而且在复核台增多后，意味着每个任务都有4*4种起点复核台和终点复核台的组合，这些都使得问题的规模急剧增加，在求解时需要注意效率。

4.4.2 模型构建

首先对模型中的下标和集合进行描述：

P – 拣货员的集合，下标 p

S – 复核台的集合，下标 k_1, k_2

I – 任务单号的集合，下标为 i

J – 拣货顺序的集合，即指示第1-49个任务，下标为 j

考虑到不合理的安排可能使货物在复核台囤积，目标函数应为全部货物出货完成的时间。

$$\min R = \max(C_{pijk_1k_2} + e_{pijk_1k_2} + 30) \quad (36)$$

其中 $e_{pijk_1k_2}$ 为该任务对应的等待时间。为保证模型的结构，并提高求解速度，将其转化为等价的线性表达式：

$$\min R = F \quad (37)$$

$$F \geq (C_{pijk_1k_2} + e_{pijk_1k_2} + 30), \quad j \in J, p \in P, i \in I, k_1, k_2 \in S \quad (38)$$

其余约束为：

① 前一个任务的终点复核台和后一个任务的起点复核台应相同

$$\sum_{j \in J} \sum_{k_1 \in S} x_{pi+1jk_3k_2} = \sum_{j \in J} \sum_{k_2 \in S} x_{pijk_1k_2}, \quad j \in J, i \in I, k_1, k_2, k_3 \in S, p \in P \quad (39)$$

② 每人每次只能执行一个任务

$$\sum_{j \in J} \sum_{k_1 \in S} \sum_{k_2 \in S} x_{pijk_3k_2} = 1, \quad i \in I, p \in P \quad (40)$$

③ 每个任务只能完成一次

$$\sum_{p \in P} \sum_{i \in I} \sum_{k_1 \in S} \sum_{k_2 \in S} x_{pijk_3k_2} = 1, \quad j \in J \quad (41)$$

④ 拣货员上一个任务完成后，经过 $t_{j_2k_1k_2}$ ，下一个任务 j_2 才能完成

$$\sum_{j \in J} \sum_{k_1 \in S} \sum_{k_2 \in S} C_{pi+1jk_3k_2} \geq \sum_{j \in J} \sum_{k_1 \in S} \sum_{k_2 \in S} (C_{pijk_3k_2} + t_{jk_1k_2}) x_{pijk_3k_2}, \quad i \in I, p \in P \quad (42)$$

其中，因为拣货员的第 i 个任务未必是任务 j ，这使得必须由 $x_{pijk_1k_3}$ 来控制

$C_{pijk_3k_2} + t_{jk_1k_3}$ 的约束是否发挥作用。 $t_{jk_1k_2}$ 表示从 k_1 号复检台出发 j 任务再到 k_2 复检台的时间。

考虑到拣货员 p 一共只有 i_p 个任务 ($i_p \leq 49$) , 所以对于未发生的所有任务, 可以认为其结束时间是一个极大的数 M 。

$$M(1 - x_{pijk_1k_2}) \leq C_{pijk_3k_2}, \quad i \in I, p \in P, j \in J, k_1, k_2 \in S \quad (43)$$

⑤ 复核台货物囤积的情况

对货物两两之间进行讨论, 若 $C_{p'i'j'k'_3k'_2} - (C_{pijk_3k_2} + 30 + e_{pijk_1k_2}) \geq 0$, 则前一个任务单已经出库, 后一个任务单还没到复核台, 此时不产生约束。

若 $C_{pijk_3k_2} - C_{p'i'j'k'_3k'_2} \geq 0$, 则认为前一个任务还没到, 后一个任务已经出库, 显然不可能发生这样的情况, 故也不产生约束。

若 $(C_{pijk_3k_2} + 30 + e_{pijk_1k_2}) > C_{p'i'j'k'_3k'_2} > C_{pijk_3k_2}$, 则任务到达复核台后还需要等待上一个任务出库才可以进行复核打包, 所以此时产生约束:

$$C_{p'i'j'k'_3k'_2} + e_{p'i'j'k'_3k'_2} \leq L_{pijk_3k_2}, \quad j \in J, i \in I, k_1, k_2 \in S, p \in P \quad (44)$$

将其转化为含有判断功能的表达式:

$$M\omega_{pijk_1k_2} \geq C_{p'i'j'k'_3k'_2} - C_{pijk_3k_2}, \quad i \in I, p \in P, j \in J, k_1, k_2 \in S \quad (45)$$

$$M(1 - \omega_{pijk_1k_2}) \geq C_{pijk_3k_2} - C_{p'i'j'k'_3k'_2}, \quad i \in I, p \in P, j \in J, k_1, k_2 \in S \quad (46)$$

$$M\omega'_{pijk_1k_2} \geq C_{p'i'j'k'_3k'_2} - L_{pijk_3k_2}, \quad i \in I, p \in P, j \in J, k_1, k_2 \in S \quad (47)$$

$$M(1 - \omega'_{pijk_1k_2}) \geq L_{pijk_3k_2} - C_{p'i'j'k'_3k'_2}, \quad i \in I, p \in P, j \in J, k_1, k_2 \in S \quad (48)$$

$$e_{p'i'j'k'_3k'_2} \geq W(L_{pijk_3k_2} - C_{p'i'j'k'_3k'_2}), j \in J, i \in I, k_1, k_2 \in S, p \in P \quad (49)$$

其中, $L_{pijk_3k_2} = C_{pijk_3k_2} + 30 + e_{pijk_1k_2}$, $W = \omega'_{pijk_1k_2}\omega_{pijk_1k_2}$

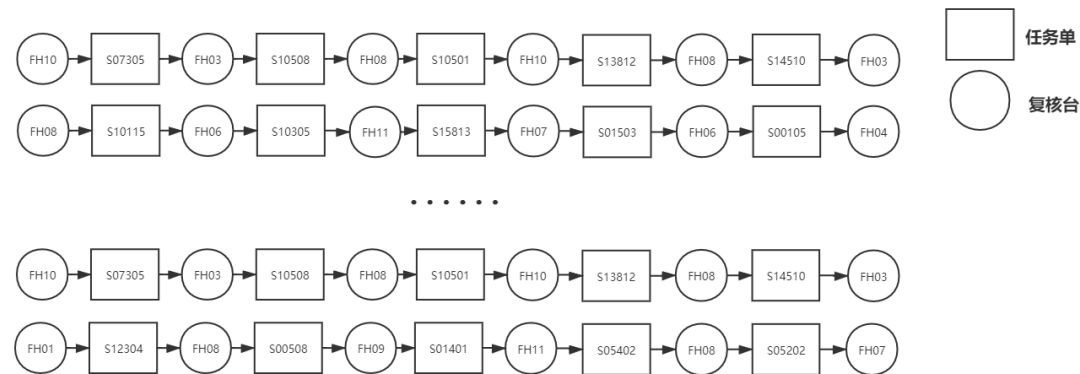
得到了最后一个任务完成复核出库的时间后, 求复核台的利用率。

$$\mu_{k_2} = \frac{\sum_{p \in P} \sum_{i \in I} \sum_{j \in J} \sum_{k_1 \in S} x_{pijk_1k_2} * 30}{R} * 100\%, \quad k_2 \in S \quad (50)$$

4.4.3 模型求解

由于问题规模较大, 不适合通过 LINGO 直接求解。在此根据模型的基本思路, 采用模拟退火遗传算法对其进行求解。模拟退火的流程已经在问题二中详细说明, 在此仅对模拟退火遗传算法做出的改变进行阐述。

在模拟退火遗传算法的求解过程中，与模拟退火算法的区别在于产生新解的方式。不同于传统的遗传算法将数据进行二进制编码，本问题中解序列的特殊结构本身就可以视为一个染色体。现以某种情况为例展示：



将染色体上的每一个基因看作一个任务单号或一个复核台号，一条染色体视为一个拣货员的任务单号序列，那么对于题目中的 9 个拣货员，利用 9 条染色体来进行表示，那么这样即可以表示 9 个拣货员的任务分配以及任务单的处理顺序。

那么利用模拟退火遗传算法，一开始随机生成一个初始解，再不断产生新解。产生新解的方式包括复核台号的随机突变与相邻染色体间不定长任务单序列的交叉。

交叉：对于每条染色体，都可随机与一条非自身的染色体进行某一段序列的交换，即改变两个拣货员处理订单的数量，订单集合以及处理顺序。交叉操作可以保证各种订单的随机组合都可以出现。

突变：对于每一条染色体上的复核台，可以随机突变为另一个复核台，若不是染色体两端的复核台，则突变后会改变某个订单的出发复核台和结束复核台，若是染色体两端的复核台，则突变后会改变某个染色体的出发复核台或者结束复核台。突变操作可以保证在染色体的订单顺序确定之后各种复核台的访问顺序都可以出现。

利用以上两种方式产生新的可行解之后，通过计算该可行解下货物出库时间，进而用模拟退火算法的原理对解的最优性进行判别，最后选择输出结果或继续迭代。

4.4.4 结果分析

通过 MATLAB 计算后，得到完成出库的时间 $R = 2579.7s$ ，FH01复核台的利用率 $\mu_1 = 15.14\%$ ，FH03复核台的利用率 $\mu_2 = 16.24\%$ ，FH10复核台的利用率 $\mu_3 = 10.61\%$ ，FH12复核台的利用率 $\mu_4 = 15.02\%$ 。

详细的任务单分配、路线规划方案已经写入附件表格中。

4.5 问题五

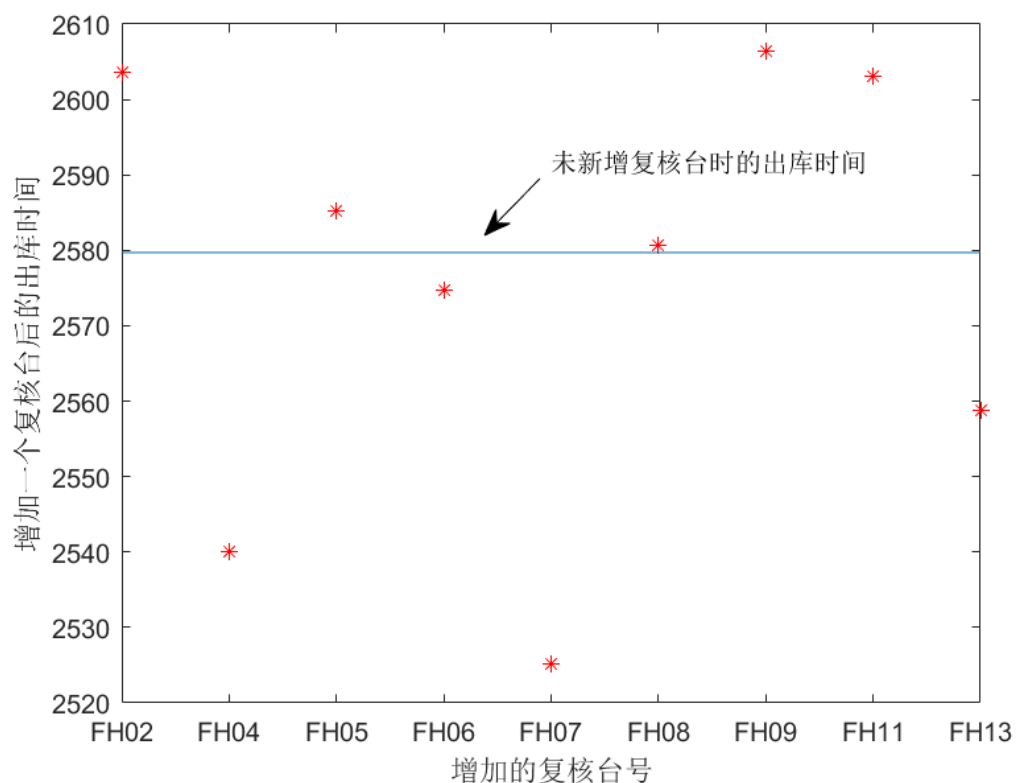
4.5.1 问题分析

本问题要求在评估在上一问的基础上增加一个复核台后对出库时间产生的影响。根据仓库的示意图可以发现，除去上一个问题中的FH01, FH03, FH10, FH12复核台，还有 9 个复核台分布于仓库的不同位置，这会导致不同复核台的增加会对最终出库时间的影响不同，理论上，根据问题四中模型的形式，可以在线性化后，通过灵敏度分析判断当前解是否仍然为最优解进而采取相应的分析处理手段。

然而由于问题规模极大，难以根据线性规划的方法求解分析。同时由于启发式算法求解速度较快，能够处理本问题的规模，所以本文使用模拟退火算法对各复核台增加的情况分别计算分析。

4.5.2 模型求解和结果分析

由于仅仅是资源数量变化，所以沿用问题四中的模型和求解方法，通过 MATLAB 实现。



观察到在增加了复核台后，大部分出库时间变化不明显，甚至某些情况下出库时间反而增加，目前估计是因为启发式算法不稳定导致的。但仍然能够说明当增添的复核台离最初的四个复核台距离越远，完全出库时间减少的就越明显。

4.6 问题六

仓内商品摆放建议

考虑实际的物流过程，需要权衡通过将畅销品放置于邻近复核台处等方式减少运输时间所带来的收益以及由于巷道宽度有限，大量拣货员同时出现在一条巷道中引起的堵塞。固然，通过专业的物流设施规划以及选用合适的物流设备能够对这一情况进行缓解，但根据问题的要求，本文在此着重讨论如何改进仓内商品的摆放方式提升拣货效率。

首先对仓库的基本情况进行分析。巷道由于具有一定的宽度限制，并且出于安全考虑，还需要与货架之间保持一定的距离，所以当多个拣货员在同个巷道内进行运输作业以及拣货作业时，必然会产生拥堵，进而由于等待产生了浪费。另外，注意到纵向的巷道宽度为 1500mm，横向的巷道宽度为 2000mm 而仓库外周宽度并无具体的宽度限制，因此可以认为通过调整商品摆放的方式，在保证畅销品尽可能靠近复核台的前提下，尽可能避免拣货员频繁穿梭于狭窄的纵向巷道即可使得拣货效率提升。

然后需要事先进行商品销售情况的分析。一方面，通过销量对产品的畅销程度进行评估，将畅销的产品放置于靠近复核台的货格，减少运输距离；另一方面，将还需要对同一个订单内产品的相关性进行评价，基于“产品相关性法则”，将相关性较高的产品放置于在相近的货格内，能够减少拣货员的运输距离，减少运输时间。

除此之外，考虑到实际过程中是人工分拣，因此还需要遵循“重量特性法则”，将较轻的货物放在高处，较重的货物放在下处，减少员工的疲劳程度，提高取货的速度，也能间接的缓解巷道拥堵的情况。

5 模型评价

5.1 模型优点

1. 考虑到包括复核台货物囤积在内的多种现实约束，真实全面的反映实际情况。
2. 通过启发式算法求解，面对复杂的大型问题仍能快速得到结果，减少了计算难度，缩短了求解时间，更符合柔性物流系统的要求。
3. 在问题三中通过相关的证明缩小了解空间，增加了求解速度。

5.2 模型缺点

1. 由于模型的规模过大，需要使用专业的计算平台进行求解，使得模型的计算成本较高。
2. 启发式算法的求解结果不稳定，无法保证结果的优度。

参考文献

- [1]宁浪,张宏斌,张斌.面向 JIT 制造的零部件配送中心货位优化研究[J].管理科学学报,2014,17(11):10-19.
- [2]王晟旭,皇甫遥遥.基于猫群算法的人工拣选作业路径优化研究[J].中国储运,2020(03):116-119.
- [3]王宏,符卓,左武.基于遗传算法的双区型仓库拣货路径优化研究[J].计算机工程与应用,2009,45(06):224-228.
- [4]李哲. 物流中心拣选单处理及拣选路径优化研究[D].大连海事大学,2011.
- [5]史朋涛. 基于智能优化算法的人工拣货仓库的仓储问题研究[D].长春工业大学,2017.