

队伍编号	202856
题号	C

仓内拣货优化问题

摘要

针对问题一：要求设计一种计算 3000 个货格和 13 个复核台之间距离的方法，本题的困难之处在于货架之间互相形成障碍，不能用欧氏距离进行描述，所以本文采用了曼哈顿距离进行计算。首先计算货格与货格之间的距离，将距离分为横向，纵向和转角距离。横向距离采用曼哈顿距离进行计算，纵向距离采用改进 $h(n)$ 函数的 A-star 算法，转角距离由两个货格之间的所在行数，列数以及是否同侧给出。接下来计算货格与复核台的距离时，将横向和纵向复核台进行了分类，同样采用改进的 A-star 算法进行计算。再通过 MATLAB 编程即可求出在附件四中的 3013×3013 的矩阵。

针对问题二：在给 P 规划理想的拣货路线时，由于第一问已经将任意两个货格和复核台的距离计算出，因此将货架造成的障碍可以忽略，转化成已知各点间距离的 TSP 问题。对 P 拣货路线的 TSP 问题的求解方法，本文采用了连续 Hopfield 神经网络进行求解，得到了优化前后的路径图。在计算出库时间时，同样本文转化为三部分：拣货时间，位移时间与复核打包时间进行计算。

针对问题三：相比第二问，任务单数量增为 5 个。因为一个任务单中，货格之间的路径距离是确定的，要使出货距离最短，即要每个任务单中的起止货格与复核台距离和相邻两个单的衔接复核台距离最短。首先通过 MATLAB 计算距离得出每个任务单的起止复核台，接着根据每个任务单的起止复核台进行首尾相接使相邻两单衔接复核台距离最短，即为任务领取最佳顺序。然后根据问题二得到的最优路线出库时间公式分别计算任务单 T0002-T0006 的出库时间，最后计算每个复核台的利用率。

针对问题四：同问题三，首先判断起始复核台和终止复核台，唯一不同的是拣货员数量增加。本文针对 9 个拣货员将任务单进行均分，采用 ETE 模型判断任务单衔接顺序，同样采用第二问的 Hopfield 对每个任务单内的路径进行求解。

针对问题五：首先利用 Markov Chain 的状态空间的转化关系来构建未增加复核台时与增加复核台后两种情况的每个复核台之间的关系，通过不断的转化迭代使最后的系统趋向于一个稳定状态。再利用排队论多服务台模型计算排队时间，通过比较未增加复核台和增加复核台后的排队时间进而得到增加 FH004 复核台比未增加复核台时的排队时间长，延长了出库时间。

最后，本文首先通过图表对畅销品货架的位置进行定性分析，接着比较了移动一个货架减少的位移时间和增加的排队时间，得到当复核台为 FH01 时，移动编号为 135 的货架到 1 货架的位置会节省 340.89s。

关键词：曼哈顿距离；A-star 算法；连续 Hopfield 神经网络；Markov Chain；M/M/S 模型

目录

一、问题的重述	1
二、问题分析	1
三、模型假设	2
四、符号说明	3
五、问题一模型的建立与求解	3
5.1 模型的准备	3
5.2 基于曼哈顿距离和改进的 A-star 算法构建距离模型	3
六、问题二模型的建立与求解	6
6.1 连续 Hopfield 神经网络的优化模型	6
6.2 CHNN 模型的优化求解	8
6.3 计算并分析出库时长	9
七、问题三模型的建立与求解	11
7.1 规划任务领取的最佳顺序	11
7.2 计算出库时长	12
7.3 计算每个复核台的利用率	13
八、问题四模型的建立与求解	14
8.1 任务单起始复核台的筛选	14
8.2 ETE 模型的建立和分配原则	14
8.3 分析并计算相关数据	15
九、问题五模型的建立与求解	17
9.1 基于马尔可夫 Markov Chain 的预测模型	17
9.2 基于 (M/M/s/∞) 排队论的估算模型	20
十、问题六模型的建立与求解	21
十一、模型的评价	23
11.1 模型的优点	23
11.2 模型的缺点	24
参考文献	24
附录 1: MATLAB 代码	25
1.1 问题一	25
1.2 问题二, 三, 四	26
附录 2: 问题三的相关图片	29

一、问题的重述

某电商公司客户订单下达仓库后，商品开始下架出库，为了使商品尽快出库，本文需要搜索最短路径，规划理想的拣货路线，省时省力，提高复核台的利用率。

问题一：当拣货员在仓库中拣货时，需要在货格之间、货格与复核台之间、复核台与复核台之间行走。由于这些行走通常要绕过障碍物，设计一种计算 3000 个货格和 13 个复核台总共 3013 个元素之间距离的方法。

问题二：假设所有复核台正常工作，任务单 T0001 等待拣货，拣货员 P 在复核台 FH10 领取了任务单 T0001。给出 P 规划理想的拣货路线，计算完成出库花费的时间。

问题三：假设 2 个复核台 (FH03, FH11) 正常工作，5 个任务单 (T0002-T0006) 等待拣货，继续由拣货员 P 负责拣货，P 初始位置为 FH03。通过建模和优化，请给 P 指定任务领取顺序，规划理想的拣货路线。计算完成尽快出库需要花费的时间和每个复核台利用率。

问题四：假设 4 个复核台 (FH01, FH03, FH10, FH12) 正常工作，49 个任务单 (T0001-T0049) 等待拣货，9 个拣货员 (P1-P9) 负责拣货，给出每个拣货员分配任务单、起始拣货复核台，并分别规划理想的拣货路线，并计算完成 49 个任务单尽快完成出库需要花费的时间和每个复核台利用率。

问题五：在问题 4 中，有 4 个复核台 (FH01, FH03, FH10, FH12) 正常工作，评估增加一个正常工作的复核台对出库时间的影响。

问题六：商品在货架中的摆放位置，会影响拣货效率。若将畅销品放置在离复核台较近的位置，拣货员行走距离相应减少，但畅销品所在货架可能拥挤，反而降低拣货效率。对于仓内商品摆放问题，提出意见。

二、问题分析

某电商公司客户订单下达仓库后，商品开始下架出库，为了使商品尽快出库，本文需要搜索最短路径，规划理想的拣货路线，省时省力，提高复核台的利用率。

针对问题一：首先根据题目建立合理坐标系。再按照图中距离标识，根据曼哈顿距离公式，将货格与货格，复核台与复核台，货格与复核台之间的距离分别分类计算，货格与货格基于横坐标，纵坐标和转角个数三个量子数分类，货格与复核台基于复核台的位置分类，复核台与复核台的分类题中给出，由此建立基于曼哈顿距离函数模型。然后采用以曼哈顿距离函数为启发函数的改进 A-star 算法求解模型，利用 MATLAB 软件计算出 3013×3013 的最短路径矩阵。

针对问题二：因为货架障碍可以忽略，本文从 TSP 问题入手对路径排序问题进行解答。在选取求解方法时，本文选取了 Hopfield 神经网络进行求解。要求出库时间，本文将其分为三部分：位移时间，拣货时间，复核时间。

针对问题三：首先需要判断每个任务单的最佳起始复核台与终止复核台，根据每个任务单的最短路径均为一条闭合曲线，将每个任务单的最佳起始复核台与终止复核台进行首尾相接即为任务领取最佳顺序。根据问题二得到的最优路线出库时间公式分别计算

任务单 T0002-T0006 的出库时间，最后通过计算空闲时间，计算每个复核台的利用率。

针对问题四：同问题三，首先判断每个任务单的起始复核台和终止复核台，根据每个任务单的起始复核台和终止复核台进行首尾相接

针对问题五：首先利用 Markov Chain 的状态空间的转化关系来构建未增加复核台时与增加复核台后两种情况的每个复核台之间的关系，通过不断的转化迭代使最后的系统趋向于一个稳定状态。再利用排队论多服务台模型计算排队时间，通过比较未增加复核台和增加复核台后的排队时间进而评估增加一个正常工作的复核台对出库时间的影响。

针对问题六：商品在货架中的摆放位置会影响拣货效率，也可能造成拣货员在畅销商品前拥堵，由此，本文对于仓内商品摆放问题，提出一些具有参考价值的建议

三、模型假设

- ◇ 假设货格与货格，货格与复核台，复核台与复核台的水平距离只与横坐标有关；
- ◇ 假设在状态空间中从初始状态到状态 n 的实际代价 $g(n)=1$ ；
- ◇ 假设 $x_i=0$ 的复核台的左边长离货格最近， $y_i=0$ 的复核台的上边长离货格最近；
- ◇ 假设每个货格到复核台时都至少走三次直角；
- ◇ 假设平均到达率 $\lambda_0=0.02$ 人/s；

四、符号说明

符号	说明
d_{corner}	转角个数
e	每个货架的货格数
k	自下而上的某一排
l_0	第一排货架与复核台之间的中心线与横坐标轴的距离
l_1	相邻两排的的中心线与相邻的相邻两排中心线的距离
d	当障碍物折线行走时横向与纵向偏移
d_2	货格长宽
d_3	竖直方向相邻两排货架纵向距离
V	系统输出状态变量的集合
d_{xy}	货格 x 到货格 y 的距离
δ_{xi}	(-1,+1) 区间的随机值
A_{mount}	从商品货格中取出的商品个数
λ	平均到达率
μ	平均服务率
W_q	平均等待时间
p	每个复核台的利用率

五、问题一模型的建立与求解

5.1 模型的准备

曼哈顿距离: 两点在南北方向上的距离加上在东西方向上的距离, 即

$$d(i, j) = |x_i - x_j| + |y_i - y_j| \quad (1)$$

对于一个具有正南正北、正东正西方向规则布局的城镇街道, 从一点到达另一点的距离正是在南北方向上旅行的距离加上在东西方向上旅行的距离, 因此曼哈顿距离又称为出租车距离, 曼哈顿距离不是距离不变量, 当坐标轴变动时, 点间的距离就会不同.

5.2 基于曼哈顿距离和改进的 A-star 算法构建距离模型

由于曼哈顿距离是对于一个具有正南正北、正东正西方向规则布局的城镇街道的公式, 与本题的竖直与水平方向相符, 故使用曼哈顿距离来刻画货格之间、货格与复核台

之间、复核台与复核台之间的距离。当拣货员在仓库中拣货时，需要在货格之间、货格与复核台之间、复核台与复核台之间行走。需要分三种情况讨论：

1、货格与货格之间：

借鉴量子力学中描述原子的状态有四个量子数，即用四个变量来表示状态。所以这里本文选用横坐标 x ，纵坐标 y ，转角数 d_{corner} ，这三个量子数来表示。则有

$$S_1 = x + y + d_{corner} \quad (2)$$

其中 x 表示为横坐标的变化， y 表示纵坐标的变化， d_{corner} 表示拐角数的变化。

通过曼哈顿距离公式，我们得到如下三个量子数的规律：

(1) 横坐标 x 的变化规律，均为

$$x = |x_j - x_i| \quad (3)$$

(2) 纵坐标 y 的变化规律分为两种情况：

这里本文采用改进的 A-star 算法计算最短路径即距离最短。A-star 算法是一种静态路网中求解最短路径最有效的直接搜索方法，也是解决许多搜索问题的有效算法。算法中的距离估算值与实际值越接近，最终搜索速度越快。公式表示为：

$$f(n) = g(n) + h(n) \quad (4)$$

其中， $f(n)$ 是从初始状态经由状态 n 到目标状态的代价估计， $g(n)$ 是在状态空间中从初始状态到状态 n 的实际代价， $h(n)$ 是从状态 n 到目标状态的最佳路径的估计代价。

当 $t_m = t_n$ 时，即为同一排时，可以走上，下两个方向的路线，如下图所示：
分别为

$$f_1 = \begin{cases} |y_j + y_i - (l_0 + 2l_1(k-1))|, & y_j + y_i < ed_2 + d_3 + (l_0 + 2l_1(k-1)) \\ 2ed_2 + 2d_3 - |y_j + y_i - (l_0 + 2l_1(k-1))|, & y_j + y_i > ed_2 + d_3 + (l_0 + 2l_1(k-1)) \end{cases} \quad (5)$$

其中 k 表示为自下而上的某一排， l_0 表示为第一排与复核台之间的中心线与横坐标轴的距离， l_1 表示为相邻两排的的中心线与相邻的相邻两排中心线的距离， d_2 表示为货格长宽， d_3 表示为竖直方向相邻两排货架纵向距离， e 为每个货架的货格数。

将采用改进 A-Star 算法，其主要原因为：A-star 算法具有较小的空间复杂度、时间复杂度，能在短时间内获得最优路径；A-Star 算法作为启发式算法具有较强的适应能力，可以根据实际的约束条件来调整启发函数；

在本文所取得启发函数为曼哈顿距离函数，即 $h(n) = f_1$ ，并取 $g(n)=1$ (因为无论路径，

拣货员每次移动的距离均相同), 则

$$f(n) = g(n) + h(n)$$

$$= 1 + \begin{cases} |y_j + y_i - (l_0 + 2l_1(k-1))|, y_j + y_i < ed_2 + d_3 + (l_0 + 2l_1(k-1)) \\ 2ed_2 + 2d_3 - |y_j + y_i - (l_0 + 2l_1(k-1))|, y_j + y_i > ed_2 + d_3 + (l_0 + 2l_1(k-1)) \end{cases} \quad (6)$$

其中, $f(n)$ 是从初始状态经由节点 n 到目标节点的距离估计, $g(n)$ 是在状态空间中从初始节点到节点 n 的实际距离, $h(n)$ 是从节点 n 到目标节点的最佳路径的估计距离.

当 $t_m \neq t_n$ 时, 即不是同一排时, 均为

$$f_2 = |y_j - y_i| \quad (7)$$

即纵坐标 y 的变化规律为

$$y = \begin{cases} \begin{cases} |y_j + y_i - (l_0 + 2l_1(k-1))|, y_j + y_i < ed_2 + d_3 + (l_0 + 2l_1(k-1)) \\ 2ed_2 + 2d_3 - |y_j + y_i - (l_0 + 2l_1(k-1))|, y_j + y_i > ed_2 + d_3 + (l_0 + 2l_1(k-1)) \end{cases}, t_m = t_n \\ |y_j - y_i|, t_m \neq t_n \end{cases} \quad (8)$$

(3) 转角的变化规律分为两种情况:

是否同列: 若 $r = |x_j - x_i| = 0$ 或 $2d_2$, 则两货格处于同列. 定义 $\delta(r)$: 若二者同列, 返回值为 0; 不然, 则为 1.

是否同侧: 若 $\omega = \frac{|x_j - x_i|}{2d_2 + d_3} = Z^+$ (Z^+ 表示为正整数), 则两货格处于同侧. 定义 $\delta(\omega)$: 若二者同侧, 则返回值为 0; 不然, 为 1.

由于货格处于不同列和处于不同侧都会使转角个数增加 2 个, 而不处于同一排不会对拐角有任何影响, 故有如下公式:

$$d_{\text{corner}} = d(2 + 2\delta(\omega) + 2\delta(r)) \quad (9)$$

其中 d_1 表示为水平方向每组货架之间的距离, d 表示为当绕障碍物折线行走时横向和竖向偏移.

本文赋予以下说明: 当 $t_m = t_n, |x_j - x_i| = 0$ 或 $2d_2, x_j = x_i$ 时, 即同排同列同侧时相邻两个货格的距离会有两个转角的增加. 在附件四: 计算结果中, 同排同列同侧的相邻两货格之间的距离 $d_0 = 800\text{mm} + 2 \times d = 800\text{mm} + 2 \times 750\text{mm} = 2300\text{mm}$, 而不仅仅两个货格间的垂直距离是 800mm

2、复核台与复核台之间:

根据题中所描述的复核台之间距离简化为两复核台坐标差的绝对值之和, 即

$$S_2 = |x_j - x_i| + |y_j - y_i| \quad (10)$$

3、货格与复核台

若复核台在左边，我们假设复核台的右边长距离货格最近，且 $\delta(l)$ 返回值为 1；若复核台在下边，我们假设复核台的上边长距离货格最近，且 $\delta(l)$ 返回值为 0。即

$$S_3 = |x_j - x_i| + |y_j - y_i| + d(3 + \delta(1)) \quad (11)$$

最后通过 MATLAB 编程实现最短距离的计算，得到 3013*3013 的矩阵

六、问题二模型的建立与求解

当拣货员在某个复核台领拣货车及任务单，根据推荐顺序依次访问任务单中商品所在货格并下架商品，将商品放在拣货车上，将拣货车送往某个复核台。同一任务单，货格访问顺序不同，行走距离也有差异。合理规划拣货路线，将会在很大程度上节约时间，提高电商公司的经济效益。

6.1 连续 Hopfield 神经网络的优化模型

在给 P 规划理想的拣货路线时，由于第一问已经将任意两个货格和复核台的距离计算出，因此将货架造成的障碍可以忽略，转化成已知各点间距离的 TSP 问题。经典 TSP 问题可以描述为：一个商品推销员要去若干个城市推销商品，该推销员从一个城市出发，需要经过所有城市后，回到出发地。应如何选择行进路线，以使总的行程最短。拣货员到不同商品货格取货物可类比为商品推销员的行为，因此，本文将拣货路线问题转为 TSP 问题进行优化计算处理。

旅行商问题属于典型的组合优化问题，目标是从组合问题的可行解集中求出最优解。利用神经网络解决组合优化问题是神经网络应用的一个重要方面。运用 Hopfield 神经网络求解 TSP 问题，把拣货路线问题的目标函数转化为神经网络的能量函数，把变量（经过的货格）与网络神经元状态相对应。这样，由于连续型 Hopfield 神经网络的稳定性理论可知，当网络的能量函数收敛于极小值时，问题的最优解也随之求出，即规划出最短时长的理想拣货路线。

通过对神经元和神经网络进行分析建立模型，用计算机模拟生物神经网络的学习和思考过程，这样的网络叫做人工神经网络（ANN）。ANN 按照信息流可以分为前馈式和后馈式，本文运用的连续 Hopfield 神经网络 (CHNN) 属于反馈式神经网络，适用于处理组合优化问题。下图为反馈式神经网络结构图。

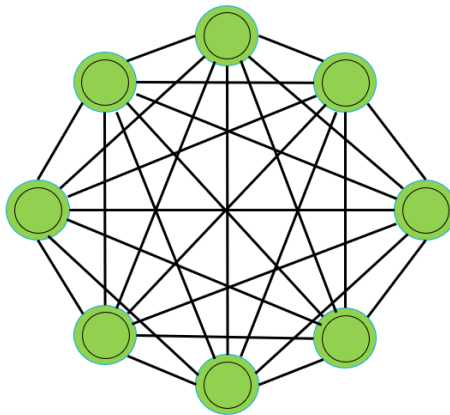


图 1 反馈式神经网络的结构图

使用 Hopfield 神经网络的 CHNN 模型，将网络抽象等效为放大电子电路。利用模拟电路 (电阻、电容以及运算放大器) 实现了对网络的神经元的表述。CHNN 电路形式如下：依据设计思路，将 TSP 问题映射为一个连续型 Hopfield 神经网络主要分为以下

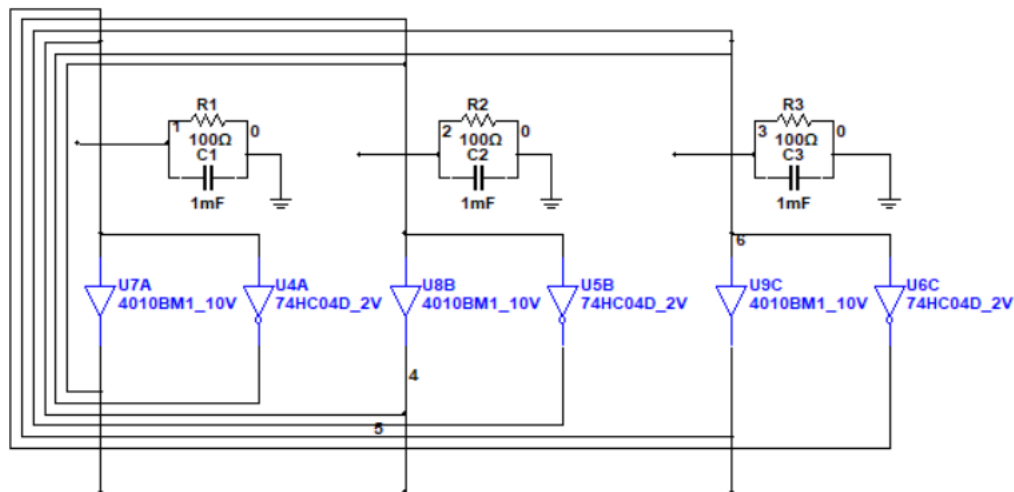


图 2 CHNN 电路形式

几个步骤：1.CHNN 模型下换位矩阵的设计

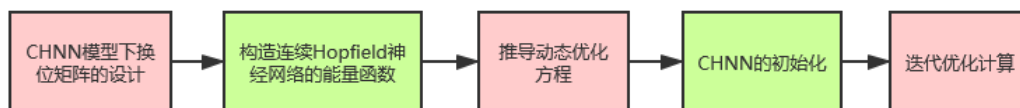


图 3 映射步骤图

为了将 TSP 问题映射为一个神经网络的动态过程，构造 NN 模型 (换位矩阵), 将每一个商品货格对应于神经网络的每一个神经元。每行每列只有一个 1，其余为 0。矩阵中 1 的总和为 N，则称该矩阵为换位矩阵。本章中 T0001 任务单共有 23 个商品，用 23×23

的矩阵表示拣货员访问的 23 个商品所在货格位置，即任务单 T0001 理想拣货路线，需要用 23×23 个神经元来实现。

2. 构造网络能量函数和动态优化方程

对于 TSP 问题，我们在一般性 CHNN 能量函数的基础上，需要考虑以下两点问题来设计规划理想拣货路线的能量函数：

(1) 规划理想拣货路线的能量函数需要量化的翻译置换矩阵的规则。

(2) 在拣货路线问题中的多种合理路线中，能量函数要有利于量化表示最短路线的解。神经网络的能量函数包含目标项和约束项两部分，其中目标项为与能量函数相对应的目标函数，约束项则为换位矩阵。目标函数为任务单 T0001 的拣货最短路径。由于能量函数过于复杂，汲取前人们对能量函数优化的经验，能量函数改进为：

$$E = \frac{A}{2} \sum_{x=1}^n \left(\sum_{i=1}^n v_{xi} - 1 \right)^2 + \frac{A}{2} \sum_{i=1}^n \left(\sum_{x=1}^n v_{xi} - 1 \right)^2 + \frac{D}{2} \sum_{x=1}^n \sum_{y=1}^n \sum_{i=1}^n v_{xi} d_{xy} v_{y,i+1} \quad (12)$$

式中，前两项为问题约束项，第三项为目标项。 x, y 为每一个商品货格数， n 为任务单 T0001 商品货格总数， d_{xy} 表示货格 x 到货格 y 的距离。 V 为系统输出状态变量的集合：

$$V = \{v_i(t) | i = 1, 2, \dots, n\} \quad (13)$$

由此可以推导出 CHNN 动态优化方程为：

$$\begin{aligned} \frac{du_{xi}}{dt} &= -\frac{\partial E}{\partial v_{xi}} \\ &= -A \left(\sum_{i=1}^n v_{xi} - 1 \right) - A \left(\sum_{y=1}^n v_{yi} - 1 \right) - D \sum_{y=1}^n d_{xy} v_{y,i+1} \end{aligned} \quad (14)$$

3. CHNN 的初始化和迭代优化 Hopfield 神经网络迭代过程对能量函数和动态方程的系数非常敏感。所以，系数的选取将影响着拣货路线理想规划的准确性和严谨性。经查找分析，网络输入初始化选取如下：

$$U_{xi}(t) = \frac{1}{2} U_0 \ln(n-1) + \delta_{xi} \quad (15)$$

式中 $U_0=1$ ， N 为商品货格数， x_i 为 $(-1, +1)$ 区间的随机值，并取 () 中 $A=200, D=100$ 。将网络初始化后，进行迭代优化计算。寻优迭代过程中包括动态方程的计算，输入输出神经元状态更新和能量函数的相关计算，采样时间设置为 $\text{step}=0.0001$ ，迭代次数为 10000。

6.2 CHNN 模型的优化求解

当 CHNN 的结构和参数设计完成后，迭代优化计算过程将会变得容易。具体步骤如下。第一步：基于第一问的模型求解，导入 T0001 任务单的商品货格之间的距离。完成导入后，进行网络的初始化操作；

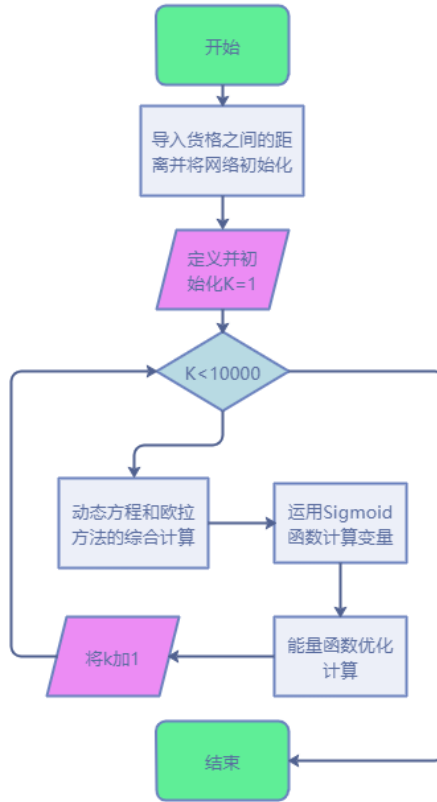


图 4 CHNN 模型迭代优化计算过程

第二步：利用动态方程计算

$$\frac{du_{xi}}{dt} \quad (16)$$

由一阶欧拉方法计算神经网络下一个时刻的输入状态，公式如下：

$$U_{xi}(t+1) = U_{xi}(t) + \frac{dU_{xi}}{dt} \Delta t \quad (17)$$

式中， Δt 为两个时刻的时间间隔；

第三步：Sigmoid 函数常被用作神经网络的激活函数，将变量映射到 0,1 之间。因此，根据 sigmoid 函数更新神经网络下个时刻的输出状态，即计算出 $V_{xi}(t)$ 。公式如下：

$$V_{xi}(t) = \frac{1}{2} \left[1 + \tanh \left(\frac{U_{xi}(t)}{U_0} \right) \right] \quad (18)$$

式中， \tanh 激活函数是 sigmoid 函数的改进版；

第四步：利用能量函数优化计算，迭代到达 10000 次后，结束优化计算。

6.3 计算并分析出库时长

任务单 T0001 中共有 23 个订单，随之对应 23 个商品货格。下图为随机产生的初始路径，当前任务单 T0001 的拣货路线长为 888.38m。经连续型 Hopfield 神经网络优化后，

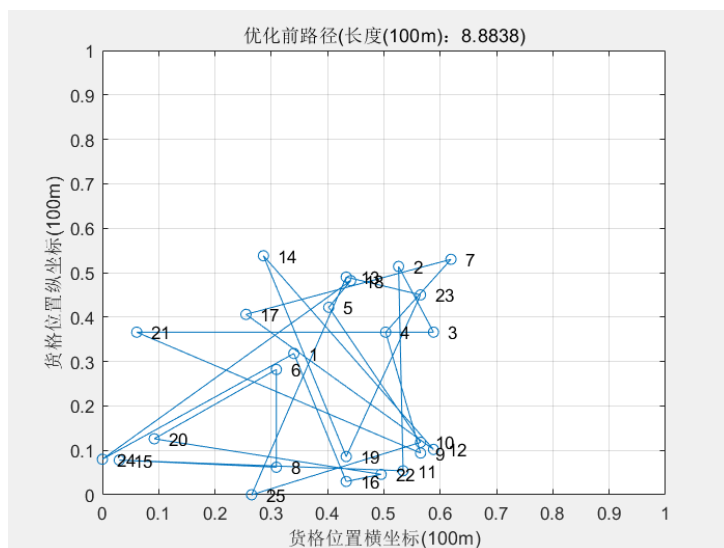


图 5 随机产生的初始路径

产生的最优途径如下图所示。未进行优化前的路径为 888.38m，优化后的最优拣货路线

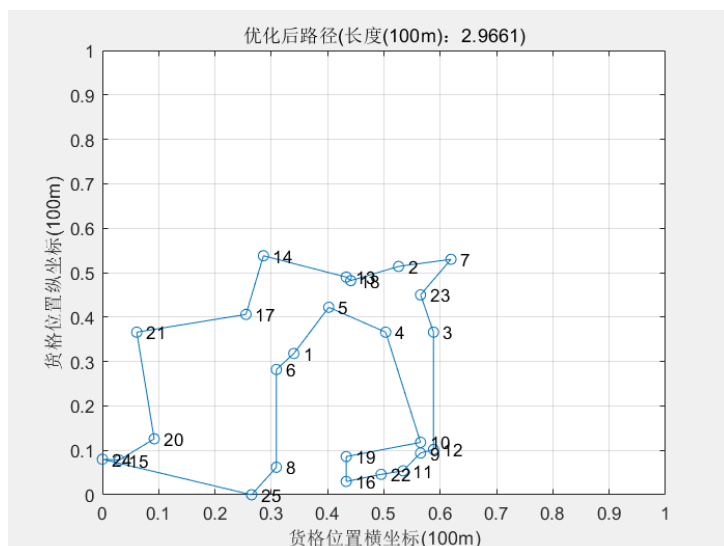


图 6 连续 Hopfield 网络优化后的路径

长为 296.61m，路线长度降低了 66.61 能量函数随迭代次数变化的曲线如下，从图中可以看出，神经网络的能量随着迭代次数增加不断减少。当能量变化值趋于 0 时，神经元状态也趋于平衡。此时，所求得的路径为任务单 T0001 理想的拣货路线。本着距离优先原则，综合上图分析，计算出 T0001 任务单中货格与复核台 FH10 距离最近的一个货格作为起始货格。终止复核台为最近的 FH02。按照上图，理想拣货路线的路线为：

FH10→ 24 → 20 → 21 → 17 → 14 → 13 → 18 → 2 → 7 → 23 → 3 → 12 → 9 → 11 → 22 → 16 → 19 → 10 → 4 → 5 → 1 → 6 → 8 → 25 → 15 → FH02(数字为上图中任务单 T0001 商品按大小排序的序号)

与拣货路线对应的货格访问路线为：

FH10→ S01308 → S07515 → S08502 → S06213 → S07212 → S10115 → S11106 →

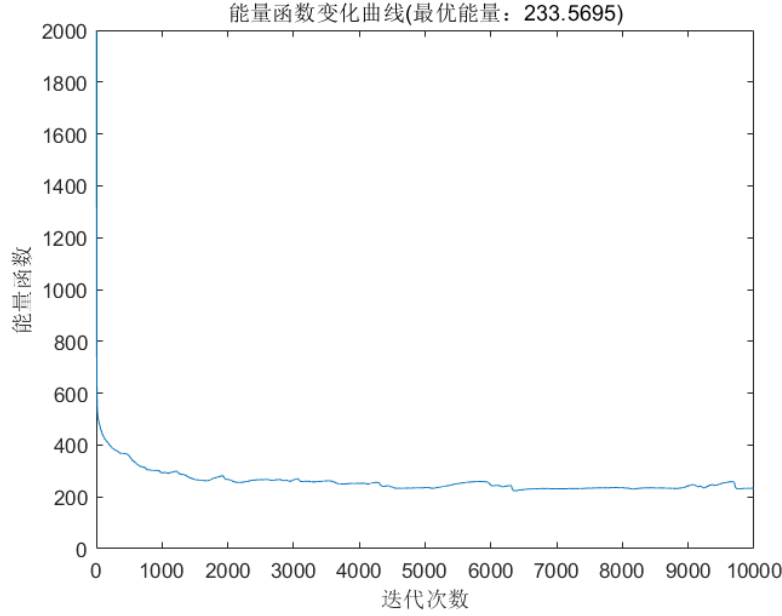


图 7 能量函数变化曲线图

$S11205 \rightarrow SS15911 \rightarrow S13509 \rightarrow S14401 \rightarrow S12609 \rightarrow 14909 \rightarrow S13004 \rightarrow S14510 \rightarrow S13812 \rightarrow S11205 \rightarrow S10508 \rightarrow S10501 \rightarrow S07305 \rightarrow S01713 \rightarrow S00107 \rightarrow FH02$

由题中所给条件可知，拣货员的行走速度 $v=1.5\text{m/s}$ ，一个订单的复核和打包需要 30 秒。若 $A_{\text{mount}} < 3$ 时，定义 $\delta(q) = 1, \delta(m) = 0$ ；若 $A_{\text{mount}} \geq 3$ 时，定义 $\delta(q) = 0, \delta(m) = 1$ 。其中 A_{mount} 表示为从商品货格中取出的商品个数。拣货员按照最优路线出库所花费的时间为

$$t = \sum \frac{(S_1 + S_3) \times 10^{-3}}{v} + \sum (5\delta(q)A_{\text{mount}} + 4\delta(m)A_{\text{mount}}) + 30n \quad (19)$$

其中 n 为订单个数。最终计算得出，任务单 T0001 拣货员按照最优路线出库所花费的时间为

$$t_1 = \frac{36.848}{1.5} + 21 \times 5 + 18 \times 4 + 30 \times 10 \approx 501.57s \quad (20)$$

七、问题三模型的建立与求解

7.1 规划任务领取的最佳顺序

当每位拣货员负责对多个任务单时，任务的领取顺序成为了规划拣货路线的主导因素。选取最优的领取顺序，使得拣货员的任务尽快出库，提高经济效益。

本文将任务领取顺序的求解进行分步处理。

第一步，判断每一个任务单的最佳起始复核台。计算该任务单中每个货格与两个复核台之间的距离，距离最小的一组规定为起始复核台和拣货员第一个经过的货格。

第二步，规划选取终止复核台的位置。分析得到，因为生成的最短路径是一条闭合曲线，所以当在与第一个经过的货格相邻的两个货格中选择一个，将其规定为最后一个经过的货格时，会降低选取终止复核台的难易程度。分别计算出这两个货格与两个复核

台对应的距离，两两距离中最近的一组规定为拣货员最后一个经过的货格和对应的终止复核台。

第三步，根据每个任务单的起始复核台和终止复核台进行首尾相接即为任务领取顺序。经过上述三个步骤，可以得到任务领取的最佳顺序，同时为下文计算出库时长打下基础。

7.2 计算出库时长

经过上述分析计算，本文得到：任务单 T0002 应从复核台 FH11 出发先到 S01115 即标号 15 经过路径最终到达 S05715 即标号 11，最后到达复核台 FH03。则任务单 T0002

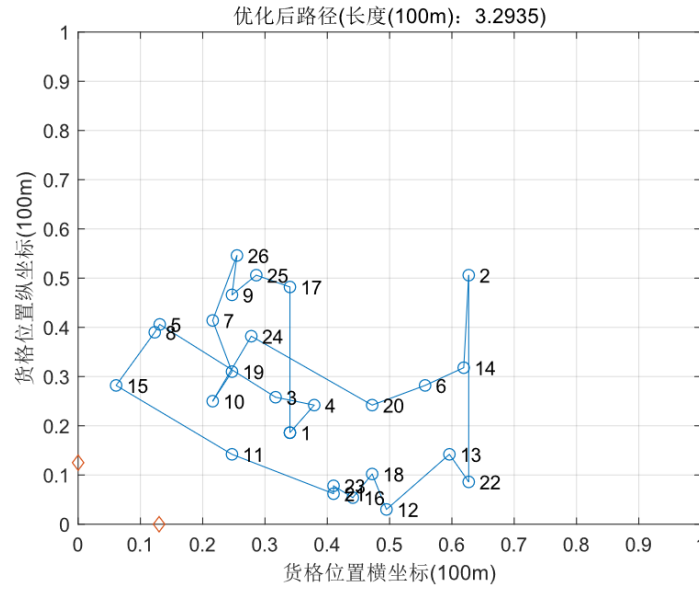


图 8 T0002 的优化路径

完成需要花费的时间为 (图中两菱形方格为 FH03 和 FH10)

$$\begin{aligned}
 t_2 &= \sum \frac{(S_1 + S_3) \times 10^{-3}}{v} + \sum (5\delta(q)A_{\text{mount}} + 4\delta(m)A_{\text{mount}}) + 30n \\
 &= \frac{329.35 + 30.14}{1.5} + 22 \times 5 + 18 \times 4 + 30 \times 15 \\
 &\approx 871.66s
 \end{aligned} \tag{21}$$

任务单 T0003 应从复核台 FH03 出发先到 S07301 即标号 10 经过路径最终到达 S08106 即标号 2，最后到达复核台 FH03。则任务单 T0003 完成需要花费的时间为

$$t_3 = \frac{231.4 + 38.36}{1.5} + 15 \times 5 + 24 \times 4 + 30 \times 11 \approx 680.84s \tag{22}$$

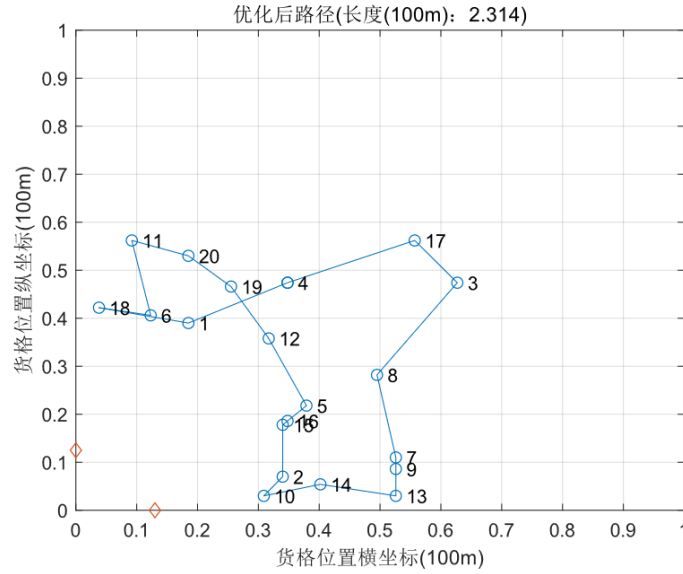


图 9 T0003 的优化路径

同理，任务单 T0004 应从复核台 FH03 出发先到 S02604 即标号 18 经过路径最终到达 S01815 即标号 11，最后到达复核台 FH11。则任务单 T0004 完成需要花费的时间为

$$t_4 = \frac{321.63 + 14.67}{1.5} + 30 \times 5 + 9 \times 4 + 30 \times 13 \approx 800.2s \quad (23)$$

任务单 T0005 应从复核台 FH03 出发先到 S02505 即标号 20 经过路径最终到达 S03405 即标号 1，最后到达复核台 FH03。则任务单 T0005 完成需要花费的时间为

$$t_5 = \frac{237.19 + 11.83}{1.5} + 21 \times 5 + 15 \times 4 + 30 \times 13 \approx 721.01s \quad (24)$$

任务单 T0006 应从复核台 FH11 出发先到 S00414 即标号 7 经过路径最终到达 S03601 即标号 16，最后到达复核台 FH03。则任务单 T0006 完成需要花费的时间为

$$t_6 = \frac{248.59 + 25.5}{1.5} + 24 \times 5 + 12 \times 4 + 30 \times 13 \approx 740.73s \quad (25)$$

至此，本文按照依次 T0003-T0005-T0004-T0002-T0006 领取任务单，通过最短路径使 T0002-T0006 这五个任务单尽快出库。且出库花费时间为

$$TOTAL_{TIME} = t_2 + t_3 + t_4 + t_5 + t_6 = 3814.44s \quad (26)$$

7.3 计算每个复核台的利用率

$IDLE_{TIME}$ 指的是空闲时间，即没有复核任务时都是在空闲时间。到达复核台行复核的任务单，需要使完成出库需要花费的时间减去复核订单的时间，即

$$IDLE_{TIME} = t - 30n \quad (27)$$

对于复核台 FH03 来说

$$IDLE_{TIME} = t_2 + t_3 + t_4 + t_5 + t_6 - 30 \times 52 = 2254.44s \quad (28)$$

$$P(FH03) = 1 - \frac{IDLE_{TIME}}{TOTAL_{TIME}} = 1 - \frac{2254.44}{3814.44} = 0.41 \quad (29)$$

对于复核台 FH11 来说

$$IDLE_{TIME} = t_2 + t_3 + t_4 + t_5 + t_6 - 30 \times 13 = 3424.44s \quad (30)$$

$$P(FH11) = 1 - \frac{IDLE_{TIME}}{TOTAL_{TIME}} = 1 - \frac{3424.44}{3814.44} = 0.10 \quad (31)$$

八、问题四模型的建立与求解

在实际电商公司运作时，将会有多名拣货员负责拣货工作。同时，他们对应多个任务单。应结合实际订单和复核台情况，给每名拣货员分配合适平均的任务量。进而，规划出所有拣货员的理想拣货工作路线。既可以降低电商公司的运营成本，同时可以提高公司工作效益。

8.1 任务单起始复核台的筛选

当任务单数量增长时，每位拣货员将对应更多的任务单。基于问题三的研究分析，从正常运作的复核台中筛选出距离最优的起始终止复核台，成为规划理想拣货路线的首要任务。

基于第三问的求解方法，得到每个任务单的多个商品货格的拣货顺序和路线。本着距离优先原则，综合路线分析，计算出每个任务单中货格与 4 个复核台距离最近的两种组合。因此，这两个复核台对应起始和终止复核台 (可能为同一个，无影响)。

8.2 ETE 模型的建立和分配原则

共有 9 名拣货员 (P1-P9) 负责 49 份任务单的拣货工作。本章中 9 名拣货员并未指定相应的工作，可以将 P1-P9 的关系类比于粒子物理学中的一组完全相同的费米子体系，体系中的粒子没有差别。由此，可以遵循平均分配 Equal division 原则，随机给拣货员们分配任务。将 9 人平均分配，其中 5 人平均分配 5 个任务单，剩余 4 人平均分配 6 个任务单。

上文已经筛选出每个任务单的起始复核台和终止复核台，本文建立自定义 ETE(end to end) 模型进行路线顺序规划处理。使得每名拣货员当前订单的终止复核台和下一个订单的起始复核台为同一个，可以有效地减少工作人员拣货路线的距离。本着 ETE 模型的原理，将 49 订单平均分配如下表：

至此，9 个拣货员的订单分配工作已经完成。每位拣货员的订单都是首尾相接的，第一个订单的起始复核台和最后一个订单的终止复核台则为该拣货员的总起始和终止

表 1 9 个拣货员的订单分配

拣货员		订单				
P1	T0006	T0015	T0036	T0010	T0012	
P2	T0018	T0029	T0007	T0022	T0024	
P3	T0008	T0039	T0040	T0044	T0047	
P4	T0016	T0026	T0032	T0049	T0045	
P5	T0043	T0031	T0009	T0042	T0021	
P6	T0001	T0004	T0013	T0014	T0020	T0023
P7	T0002	T0003	T0011	T0034	T0038	T0041
P8	T0017	T0019	T0027	T0030	T0048	T0046
P9	T0005	T0025	T0028	T0033	T0035	T0037

复核台。在上文得到的 49 个任务单的起始终止复核台进行查找，得到下表：

表 2 每个拣货员的起始复核台

拣货员	起始复核台
P1	FH12
P2	FH10
P3	FH01
P4	FH01
P5	FH10
P6	FH10
P7	FH12
P8	FH10
P9	FH03

8.3 分析并计算相关数据

1、出库时长

基于问题二的公式分析，已经得到最优路线的出库时间公式。公式如下：

$$t = \sum \frac{(S_1 + S_3) \times 10^{-3}}{v} + \sum (5\delta(q)A_{\text{mount}} + 4\delta(m)A_{\text{mount}}) + 30n \tag{32}$$

可以根据公式，计算出每个拣货员完成所有任务的时间：

$$t_{P1} = \frac{1568.69}{1.5} + 663 + 30 \times 66 = 3688.79s \tag{33}$$

$$t_{P2} = \frac{1703.41}{1.5} + 585 + 30 \times 71 = 3850.61s \tag{34}$$

$$t_{P3} = \frac{1628.86}{1.5} + 654 + 30 \times 78 = 4079.91s \quad (35)$$

$$t_{P4} = \frac{1623.94}{1.5} + 642 + 30 \times 71 = 3854.63s \quad (36)$$

$$t_{P5} = \frac{1485.13}{1.5} + 684 + 30 \times 68 = 3714.09s \quad (37)$$

$$t_{P6} = \frac{1921.22}{1.5} + 810 + 30 \times 83 = 4580.81s \quad (38)$$

$$t_{P7} = \frac{1863.46}{1.5} + 792 + 30 \times 82 = 4494.31s \quad (39)$$

$$t_{P8} = \frac{2008.61}{1.5} + 804 + 30 \times 78 = 4483.07s \quad (40)$$

$$t_{P9} = \frac{1688.91}{1.5} + 582 + 30 \times 82 = 4167.94s \quad (41)$$

故完成出库所花费的时间为

$$t' = t_{P1} + t_{P2} + t_{P3} + t_{P4} + t_{P5} + t_{P6} + t_{P7} + t_{P8} + t_{P9} = 36914.16s \quad (42)$$

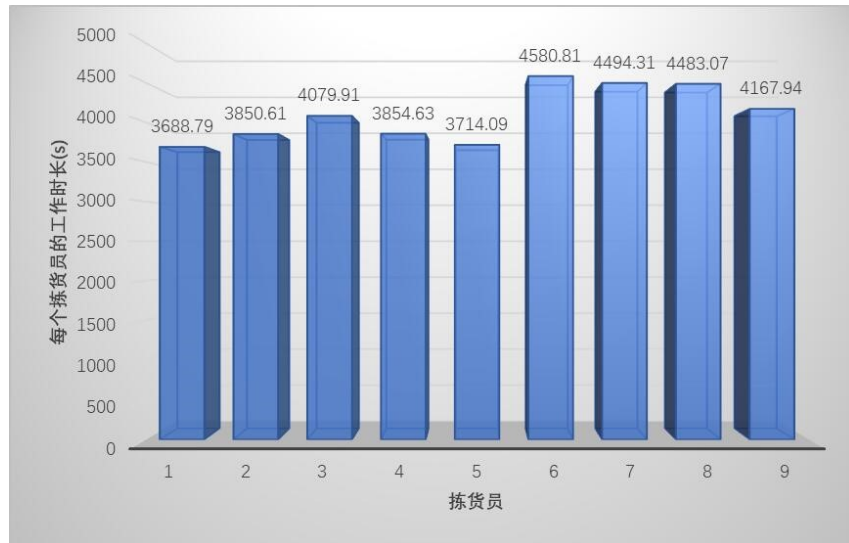


图 10 每个拣货员的工作时长

为了使 49 个任务单尽快出库，应尽量使每个拣货员的工作时长相同，但也无法不受外界因素的影响。从图中可以看出，每个拣货员的工作时长存在差值，计算出的时间标准差位 329.485s。由此看出，本文建立的公式和模型较为准确。

2. 复核台利用率基于问题二的公式分析，利用复核台利用率公式计算 4 个复核台 (FH01,FH03,FH10,FH12) 的对应数值，公式如下：

$$P = 1 - \frac{IDLE_{TIME}}{TOTAL_{TIME}} \quad (43)$$

$$P(\text{FH01}) = 1 - \frac{36914.16 - 30 \times 33}{36914.16} = 0.03 \quad (44)$$

$$P(\text{FH03}) = 1 - \frac{36914.16 - 30 \times 388}{36914.16} = 0.32 \quad (45)$$

$$P(\text{FH10}) = 1 - \frac{36914.16 - 30 \times 88}{36914.16} = 0.07 \quad (46)$$

$$P(\text{FH12}) = 1 - \frac{36914.16 - 30 \times 170}{36914.16} = 0.14 \quad (47)$$

九、问题五模型的建立与求解

通过对问题四的研究分析可以看出，当任务单数量增加时，拣货员的工作非常繁重，开设的 4 个复核台周围也是摩肩接踵。因此，本文欲研究：增加一个正常工作的复核台，是否可以缓解因任务单数量增多，而使复核台周围发生拥挤的情况。

9.1 基于马尔可夫 Markov Chain 的预测模型

Markov Chain 体现的是状态空间的转化关系。最简单的马尔科夫过程是一阶模型，下一个状态只决定与当前的状态。马尔可夫链中随机变量的状态随时间步的变化被称为演变或转移。马尔可夫链结构的两种途径为转移矩阵和状态图。状态图的转换关系可以用一个转换矩阵 T 来表示，当前状态的转换矩阵不断转换迭代，最后系统趋向于一个稳定状态。Markov Chain 的转移矩阵的定义式如下：

$$P_{n,n+1} = (P_{i_n,i_{n+1}}) = \begin{bmatrix} P_{0,0} & P_{0,1} & P_{0,2} & \cdots \\ P_{1,0} & P_{1,1} & P_{1,2} & \cdots \\ P_{2,0} & P_{2,1} & P_{2,2} & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix} \quad (48)$$

式中，下标 i_n 表示第 n 步的转移， p 为转移概率。

本文通过利用 Markov Chain 状态空间的转化关系，来构建增加复核台前后两种情况下复核台之间的关系。从而，可以直观的看出增加复核台前后的对比状况。同时为后面评估增加复核台对出库时间影响作准备。

1. 未增加复核台时基于问题 4 得到的数据，进行统计得知，当开设 4 个复核台 (FH01,FH03,FH10,FH13) 时：从复核台 FH03 出发拣货的拣货员，结束后，依旧回到 FH03 复核台复检的概率为 0.7895；到 FH10 复检的概率为 0.1053；到 FH12 复检的概率为 0.1053...具体转化关系以及概率大小如图所示：

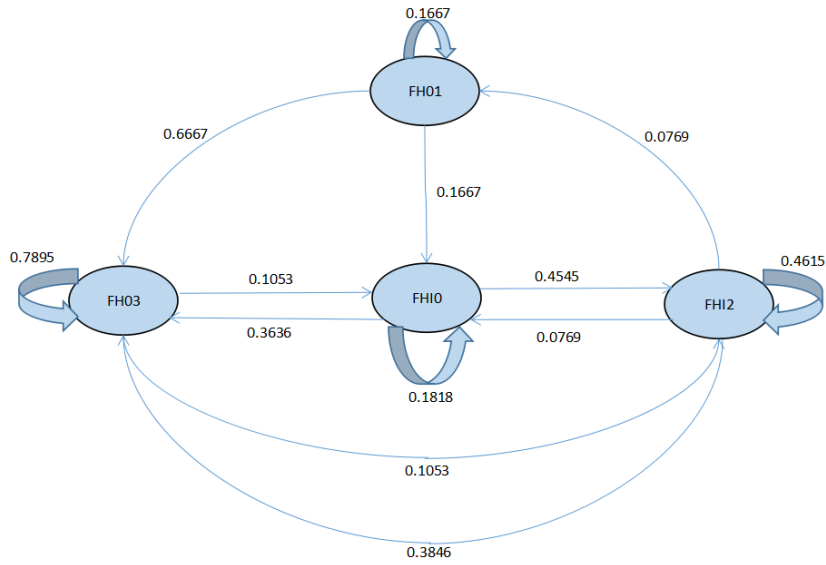


图 11 未增加复核台时的转换图

参照式子 (48) 的转换矩阵，将得到的概率值分别按行和列依次填入实际数值转换矩阵，从而得到如下式子：

$$T = \begin{bmatrix} 0.1667 & 0.6667 & 0.1667 & 0 \\ 0 & 0.7895 & 0.1053 & 0.1053 \\ 0 & 0.3636 & 0.1818 & 0.4545 \\ 0.0769 & 0.3846 & 0.0769 & 0.4615 \end{bmatrix} \quad (49)$$

经上述计算，得知当前状态为 $u(x) = (0.1224, 0.3878, 0.2245, 0.2653)$ 。4 个复核台的占用率可以直观看出复核台的工作现状，将迭代次数作为自变量，4 个复核台的占用率作为因变量。假设按照这个转换矩阵一直不断进行转换，通过迭代计算可得到 4 个复核台的占用率。如下图所示：

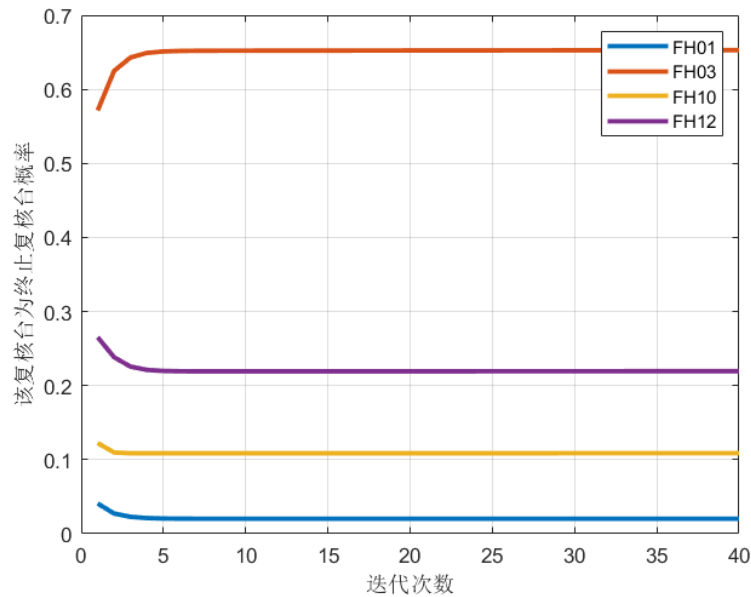


图 12 四个复核台的占用率

从图中可以看出：FH01,FH03,FH10,FH12 四个复核台的占用率最终都趋向于稳定，且复核台 FH03 的占用率高达 65%。

2. 增加一个复核台时：由上述分析，复核台 FH03 的占用率高达 65%。由此可知，FH03 复核台占用率较高，可能出现拥挤的状况。本着距离最优原则对选取 FH04 复核台为新增复核台。统计得知，当开设 5 个复核台（FH01,FH03,FH10,FH13，FH04）时：从复核台 FH03 出发拣货的拣货员，结束后，依旧回到 FH03 复核台复检的概率为 0.2727，到 FH12 复检的概率为 0.0909...具体转化关系以及概率大小如图所示：

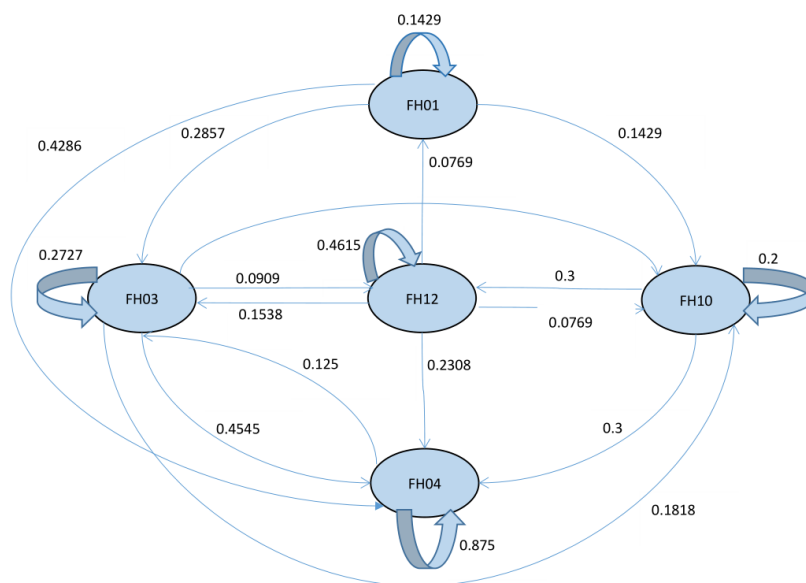


图 13 增加复核台后的 Markov Chain

将得到的数值分别按行和列依次填入实际数值转换矩阵，从而得到如下式子

$$T = \begin{bmatrix} 0.1429 & 0.2857 & 0.4286 & 0.1429 & 0 \\ 0 & 0.2727 & 0.4545 & 0.1818 & 0.0909 \\ 0 & 0.125 & 0.875 & 0 & 0 \\ 0 & 0.2 & 0.3 & 0.2 & 0.3 \\ 0.0769 & 0.1538 & 0.2308 & 0.0769 & 0.4615 \end{bmatrix} \quad (50)$$

经上述计算，得知当前状态为 $v(x) = (0.1224, 0.2245, 0.2041, 0.2245, 0.2653)$ 。5 个复核台的占用率可以直观看出新增复核台后占有率的整体变化。假设按照这个转换矩阵一直不断进行转换，通过迭代计算可得到 5 个复核台的占用率。如下图所示：

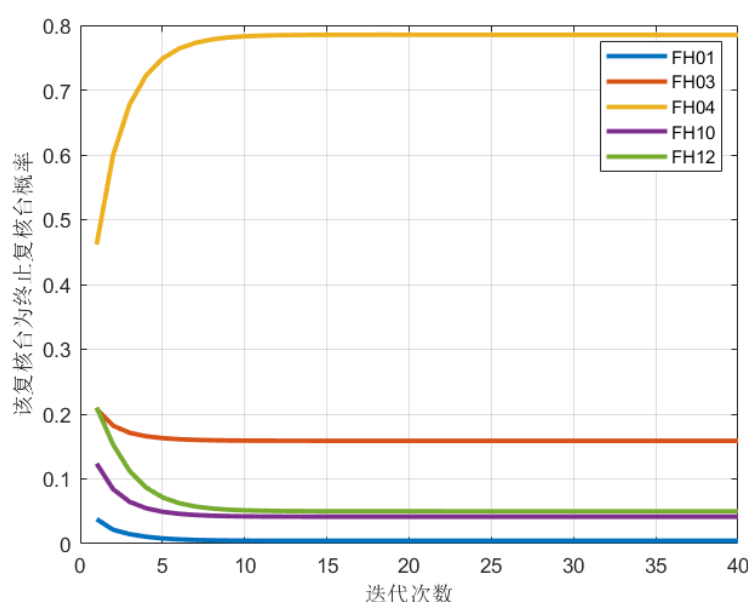


图 14 四个复核台与新增 FH04 的占用率

从图中可以看出：当增加了一个复核台 FH04 时，复核台 FH03 的占用率明显下降，而复核台 FH04 的占用率高达 79%，最后趋向于稳定。

通过对两幅图的比较，虽然 FH04 缓解了 FH03 的拥挤现象，但同时产生了 FH04 复核台的拥堵问题。且将比 FH03 复核台附近的拥挤现象更为严重，严重影响出库效率。下面利用排队论进行定量计算。

9.2 基于 (M/M/s/∞) 排队论的估算模型

排队是在日常生活中经常遇到的现象，如到商店购买物品、到医院看病等等，常常都需要排队。此时要求服务的数量超过服务机构（服务台、服务员等）的容量。也就是说，到达的顾客不能立即得到服务，因而出现了排队现象。排队论（Queuing Theory）也称随机服务系统理论，是为解决上述问题而发展的一门学科。

排队论中包含着不同类型的估算模型，其中 (M/M/s/∞) 模型表示输入过程是 Poisson

流, 服务时间服从负指数分布, 系统有 S 个服务台平行服务, 其系统容量为无穷的等待制排队系统. 通过上文分析, 可以对增加复核台前后的排队时长进行比较. 因而, 可以将评估增加复核台对出库时间的影响问题, 转化为排队论问题. 同时, 采用 $(M/M/s/\infty)$ 模型进行求解.

设拣货员单个到达, 相继到达时间间隔服从参数为 λ 的负指数分布, 系统中共有 s 个复核台, 每个复核台的服务时间相互独立, 且服从参数为 μ 的负指数分布. 记 $\rho_s = \frac{\rho}{s} = \frac{\lambda}{s\mu}$, 当 $\rho_s < 1$ 时, 有

(1) 整个复核台处空闲的概率

$$p_0 = \left[\sum_{n=0}^{s-1} \frac{\rho^n}{n!} + \frac{\rho^s}{s!(1-\rho_s)} \right]^{-1} \quad (51)$$

(2) 平均排队长

$$\begin{aligned} L_q &= \sum_{n=s+1}^{\infty} (n-s)p_n = \frac{p_0 \rho^s}{s!} \sum_{n=s}^{\infty} (n-s)\rho_s^{n-s} \\ &= \frac{p_0 \rho^s}{s!} \frac{d}{d\rho_s} \left(\sum_{n=1}^{\infty} \rho_s^n \right) = \frac{p_0 \rho^s \rho_s}{s!(1-\rho_s)^2} \end{aligned} \quad (52)$$

平均队长

$$L = L_q + \rho \quad (53)$$

(3) 平均等待时间

$$W_q = \left[\sum_{n=0}^{s-1} \frac{\lambda^n}{n! \mu^n} + \frac{\lambda^s}{s! \mu^s \left(1 - \frac{\lambda}{s\mu}\right)} \right]^{-1} \frac{\lambda^{s+1}}{(s+1)! \mu^{s+1} \left(1 - \frac{\lambda}{s\mu}\right)^2} \quad (54)$$

由于未增加复核台时和增加复核台后都只有一个复核台出现拥堵问题, 故 s 都取 1, 且每个订单复核打包的时间都是 30 秒, 故平均服务率 $\mu=1/30$ 人/s.

(1) 未增加复核台时: 假设平均到达率 $\lambda_0=0.02$ 人/s, $\rho_s = \frac{\rho}{s} = \frac{\lambda}{s\mu}$, 则平均等待时间 $W_q=45s$.

(2) 增加复核台后: 平均到达率 $\lambda_1 = \frac{0.79}{0.65} \times \lambda_0 = 0.024$, $\rho_1 = \frac{\lambda_1}{\mu} = 0.72$, 则平均等待时长为 $W_q=80.80s$

至此发现, 增加复核台后的出库时间大于增加前的出库时间. 根据上述计算分析得知: 拥堵的复核台是对出库的时间变化的主要原因, 增加复核台并不一定会缩短出货时间.

十、问题六模型的建立与求解

根据货物的特性将仓库的储位合理安排, 以便快捷地移动货物, 从而实现仓库货物移动时间的最优化和提高仓库容积利用率. 将畅销品放置在离复核台较近的位置, 减少拣货员行走距离, 从而节省时间. 但与此同时, 畅销品所在货架可能发生拥挤的状况, 反而降低拣货效率. 因此, 控制并舒缓因畅销品货架聚集而导致的拥挤以及成为货物移

动优化的焦点问题，

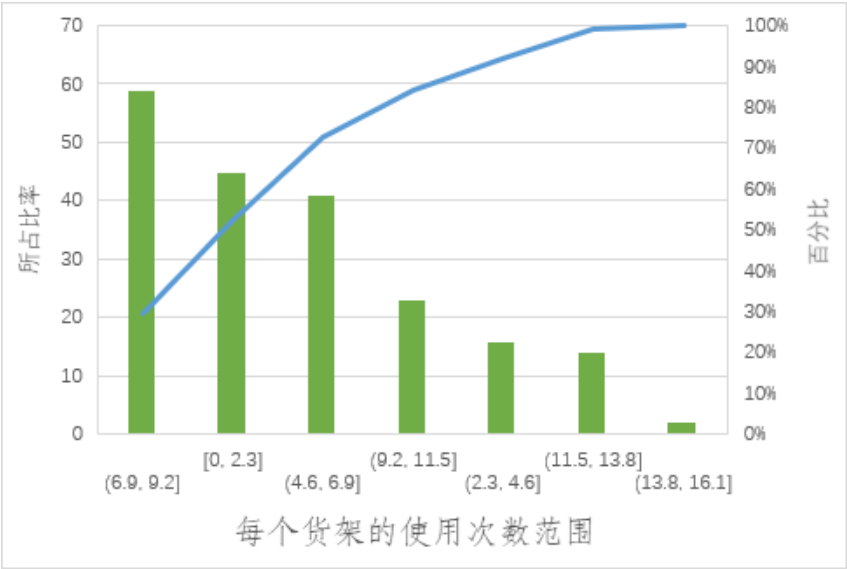


图 15 货架使用次数范围图

首先本文做了每个货架使用次数范围的柱形图，通过上图可以看出所占百分比最大的为使用次数在 (6.9,9.2) 次的，其次为使用次数在 (0, 2.3) 次的，所占百分比最小的为 (13.8, 16.1) 次的。可以看到，货架的摆放方式有很大的改进空间。

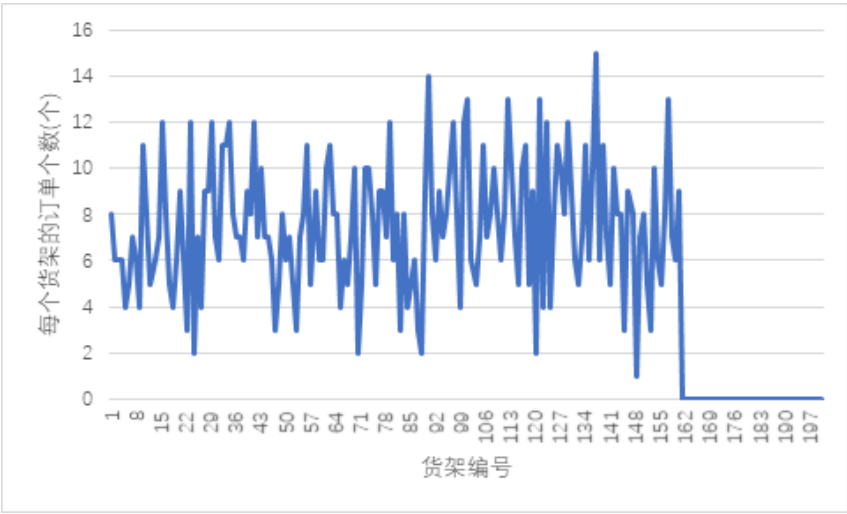


图 16 每个货架使用次数图

接着本文又做出了每个货架的使用次数，可以发现：前排的货架 (编号为 1 到 21) 使用次数并不高，最高的反而 135 到 155 编号的货架。由此，本文的方案是二者互换位置,但互换货架的数量 q 待定。

互换货架位置可以有效减少拣货员的位移时间，但是会对复核台的等待时间造成影响，造成拥堵。这样如何摆放仓内货品问题就变成了减少的位移时间和增加的等待时间二者大小的比较问题。

减少的位移时间 t_s 为

$$t_s = \frac{\Delta S}{V} \quad (55)$$

其中, ΔS 为货架位置变换而减少的位移距离, 由第一问给出; v 为拣货员移动的速度, 即 1.5m/s.

在第 (5) 问中已经得到由排队论导出的排队时间, 这里假设搬移畅销货架的数量 q 与 λ 之间存在线性关系, τ 为比例系数

$$\lambda = \lambda_0 + \tau q \quad (56)$$

已知排队时间 W_q 为

$$W_q = \left[\sum_{n=0}^{s-1} \frac{\lambda^n}{n! \mu^n} + \frac{\lambda^s}{s! \mu^s \left(1 - \frac{\lambda}{s\mu}\right)} \right]^{-1} \frac{\lambda^{s+1}}{(s+1)! \mu^{s+1} \left(1 - \frac{\lambda}{s\mu}\right)^2} \quad (57)$$

在此问题中, 货架位置变换导致了平均到达率 λ 的上升, 会导致复核台的拥堵.

下面应当求出两个时间相同时的搬移货架数量 q_0 , 但方程过于复杂, 难以求出解析解, 故将数值代入求数值解进行比较. 以复核台 FH01 为例, 当 $q=1$ 时, 货架 1 与货架 135 互换位置, 通过 MATLAB 编程得到 $\Delta S = 896m$, 节省时间 $t_s = 896/1.5(s) = 597.3s$. 计算拥堵时间时, 与第五问相同, 取 $\mu=0.033$ 人/s, $\lambda_0 = 0.02$ 人/s, $\tau = 0.01$ 人/s, 通过 LINGO 得到增加的排队时间 $W_q = 303.03 - 46.62 = 256.41s$. 可见, 当迁移货架数量 $q=1$ 时, $t_s > W_q$, 此时迁移畅销品货架为最优选择.

由于 $t_s = W_q$ 时的方程过于难解和时间限制, 本文在这里不再讨论 $q \geq 2$ 时的情况.

十一、模型的评价

11.1 模型的优点

- 问题一使用曼哈顿距离公式优点在于固定直角坐标系上两点所形成的线段对轴产生的投影的距离总和与题中只能走直线相符合.
- 在计算货格间距离时, 采用了分为横坐标, 纵坐标, 拐角三方面计算的方法, 将各种情况进行概括, 大大减少了是否同排, 同列, 同侧的讨论工作量
- 改进的 A-star 算法优点在于快速判断是否为最短路径; 快速查找最小 f 值的路径以及优化路径. A-Star 算法作为启发式算法具有较强的适应能力, 可以根据实际的约束条件来调整启发函数;
- 连续 Hopfield 神经网络的 CHNN 优化模型, 将网络抽象等效为放大电子电路. 利用模拟电路 (电阻、电容以及运算放大器) 实现了对网络的神经元的表述. 创建了神经网络优化的新途径, 而且引入了能量函数的概念. 可以依据 Hopfield 网络模型的工作运行规则不断地进行状态变化, 最终能够到达具有某个极小值的目标函数. •

11.2 模型的缺点

- 问题五中的 M/M/S 模型的 λ 的数值为假设量，会对模型的结果造成一定的影响
- 问题五添加复核台时，只考虑了添加 FH004 这一种情况，实际其余 8 个复核台都可添加，研究情况不全面。
- 问题六中只研究了迁移货架数为 1 的情况，对其他情况没有讨论，会造成结果不完善。
- 问题一中对货格和复核台距离的计算时，没有遵守规定：货格与复核台距离简化为货格中点到复核台最近一条边中点的距离，只是计算了从货格到复核台中心的距离。

参考文献

- [1] 梅晶. 并行分布式系统中的节能调度策略与算法研究 [D]. 湖南大学,2015.
- [2] 赵永全, 牛宝龙. 基于排队论的仓储企业物流中心服务系统优化研究 [J]. 生产力研究,2013(05):163-166.
- [3] 韩兵. 基于排队论的制造企业入厂物流设备优化研究 [D]. 天津大学,2014.
- [4] 范春彦, 贺沁荣, 高歆. 排队论在情报处理方面的应用 [J]. 陕西师范大学学报 (自然科学版),2009,37(S1):5-7.
- [5] 史峰.MATLAB 神经网络 30 个案例分析 [M].1. 北京航空航天大学出版社, 2010.4.
- [6] Frank R.Giordano 等. 数学建模 [M].4. 机械工业出版社, 2011.7.

附 录

附录 1: MATLAB 代码

1.1 问题一

```
clc
clear all
s = xlsread('data.xlsx','counter','A2:A3014');
d1=1;d2=2;d3=3;d4=4;d5=5;d6=6;d7=7;d8=8;
for i=1:size(s,1)
m1=(fix(s(i,1)/100)-d1)/8;m2=(fix(s(i,1)/100)-d2)/8;m3=(fix(s(i,1)/100)-d3)/8;
m4=(fix(s(i,1)/100)-d4)/8;m5=(fix(s(i,1)/100)-d5)/8;m6=(fix(s(i,1)/100)-d6)/8;
m7=(fix(s(i,1)/100)-d7)/8;m8=(fix(s(i,1)/100)-d8)/8;
if m1==fix(m1)||m2==fix(m2)||s(i,1)==1||s(i,1)==2
s(i,1)=1;
elseif m3==fix(m3)||m4==fix(m3)||s(i,1)==3||s(i,1)==4
s(i,1)=2;
elseif m5==fix(m5)||m6==fix(m6)||s(i,1)==5||s(i,1)==6
s(i,1)=3;
elseif m7==fix(m7)||m8==fix(m8)||s(i,1)==7||s(i,1)==8
s(i,1)=4;
end
end
p = xlsread('data.xlsx','counter','B2:C3014');
for i=1:3013
for j=1:3013
%%x轴距离
x(i,j)=abs(p(i,1)-p(j,1));
%%y轴距离
if s(i,1)~=s(j,1)
y(i,j)=abs(p(i,2)-p(j,2));
elseif s(i,1)==s(j,1)&&abs(p(i,1)-p(j,1))==0
y(i,j)=abs(p(i,2)-p(j,2));
else
if p(i,2)+p(j,2)-(3000*round(s(i,1)/4)+14000*(s(i,1)-1))-(3000*round(s(i,1)/4)
+14000*(s(j,1)-1))<12000
y(i,j)=abs(p(i,2)+p(j,2))-(3000*round(s(i,1)/4)+14000*(s(i,1)-1))-(3000*round(
s(i,1)/4)+14000*(s(j,1)-1));
else
y(i,j)=12000*2-(abs(p(i,2)+p(j,2))-(3000*round(s(i,1)/4)+14000*(s(i,1)-1))
-(3000*round(s(i,1)/4)+14000*(s(j,1)-1)));
end
end
end
```

```

%%拐角距离
u=0;
k=fix(abs(p(i,1)-p(j,1))/3100);
if k~=fix(k)%判断是否为同一侧
u=u+2;
end
if abs(p(i,1)-p(j,1))~=0&&abs(p(i,1)-p(j,1))~=800%判断是否为同一列
u=u+2;
end
R(i,j)=750*(2+u);
L(i,j)=x(i,j)+y(i,j)+R(i,j);
if p(i,1)==p(j,1)&& p(i,2)==p(j,2)
L(i,j)=0;
end
end
end
xlswrite('计算结果.xlsx',L);

```

1.2 问题二，三，四

%% 连续Hopfield神经网络的优化—旅行商问题优化计算

```

clc
clear
q=2;
zd=26;
h=zd-q+1;
t=xlsread('data.xlsx','任务单','G2:G26');
load L
for i=1:h
for j=1:h
a=t(i,1);
b=t(j,1);
w(i,j)=L(a,b);
end
end
global A D
citys =xlsread('data.xlsx','任务单','E2:F26');
citys =citys ./100000;
%% 计算相互城市间距离
distance=w./100000;
%% 初始化网络
N=size(citys,1);
A=200;
D=100;

```

```

U0=0.1;
step=0.0001;
delta=2*rand(N,N)-1;
U=U0*log(N-1)+delta;
V=(1+tansig(U/U0))/2;
iter_num=10000;
E=zeros(1,iter_num);
%% 寻优迭代
for k=1:iter_num
% 动态方程计算
dU=diff_u(V,distance);
% 输入神经元状态更新
U=U+dU*step;
% 输出神经元状态更新
V=(1+tansig(U/U0))/2;
% 能量函数计算
e=energy(V,distance);
E(k)=e;
end
%% 判断路径有效性
[rows,cols]=size(V);
V1=zeros(rows,cols);
[V_max,V_ind]=max(V);
for j=1:cols
V1(V_ind(j),j)=1;
end
C=sum(V1,1);
R=sum(V1,2);
flag=isequal(C,ones(1,N)) & isequal(R',ones(1,N));
%% 结果显示
if flag==1
% 计算初始路径长度
sort_rand=randperm(N);
citys_rand=citys(sort_rand,:);
Length_init=dist(citys_rand(1,:),citys_rand(end,:));
for i=2:size(citys_rand,1)
Length_init=Length_init+dist(citys_rand(i-1,:),citys_rand(i,:));
end
% 绘制初始路径
fig2=figure(1);
plot([citys_rand(:,1);citys_rand(1,1)],[citys_rand(:,2);citys_rand(1,2)],'o-');

for i=1:length(citys)
text(citys(i,1),citys(i,2),[' ' num2str(i)])

```

```

end
text(citys_rand(1,1),citys_rand(1,2),[' ' ])
text(citys_rand(end,1),citys_rand(end,2),[' ' ])
title(['优化前路径(长度(100m): ' num2str(Length_init) ')'])
axis([0 1 0 1])
grid on
xlabel('货格位置横坐标(100m)')
ylabel('货格位置纵坐标(100m)')
frame2 = getframe(fig2); % 获取frame
picture2 = frame2im(frame2); % 将frame转换成imwrite函数可以识别的格式
% 计算最优路径长度
[V1_max,V1_ind]=max(V1);
citys_end=citys(V1_ind,:);
Length_end=dist(citys_end(1,:),citys_end(end,:));
for i=2:size(citys_end,1)
Length_end=Length_end+dist(citys_end(i-1,:),citys_end(i,:));
end
disp('最优路径矩阵');V1
% 绘制最优路径
fig=figure(2);
plot([citys_end(:,1);citys_end(1,1)],...
[citys_end(:,2);citys_end(1,2)],'o-')
for i=1:length(citys)
text(citys(i,1),citys(i,2),[' ' num2str(i)])
end
text(citys_end(1,1),citys_end(1,2),[' ' ])
text(citys_end(end,1),citys_end(end,2),[' ' ])
title(['优化后路径(长度(100m): ' num2str(Length_end) ')'])
axis([0 1 0 1])
grid on
xlabel('货格位置横坐标(100m)')
ylabel('货格位置纵坐标(100m)')
frame = getframe(fig); % 获取frame
picture = frame2im(frame); % 将frame转换成imwrite函数可以识别的格式
mkdir image
% 绘制能量函数变化曲线
figure(3)
plot(1:iter_num,E);
ylim([0 2000])
title(['能量函数变化曲线(最优能量: ' num2str(E(end)) ')']);
xlabel('迭代次数');
ylabel('能量函数');
else
disp('寻优路径无效');

```

```

end
xlswrite('test2.xlsx',V_ind,'A4:A4')
imwrite(picture,'image/优化后路径2后.png')
imwrite(picture2,'image/优化后路径2前.png')

```

附录 2：问题三的相关图片

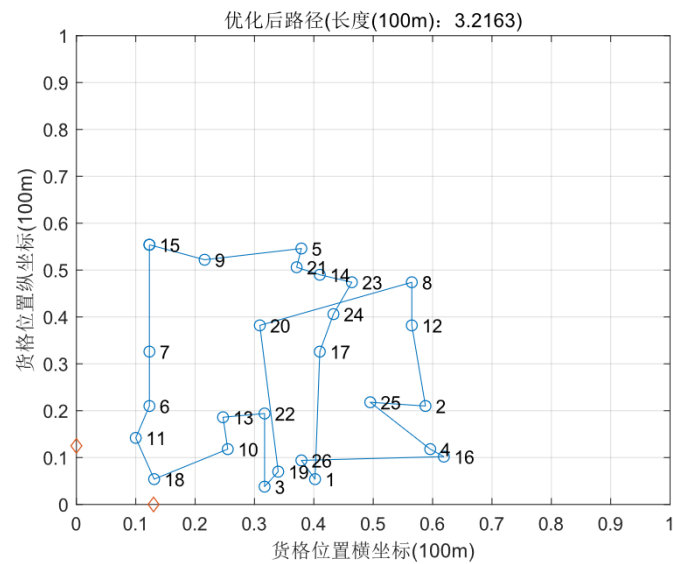


图 17 T0004 的优化路径

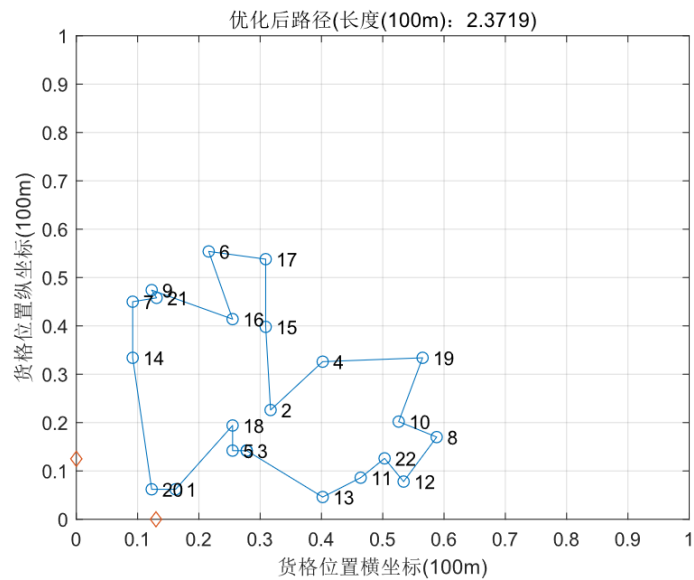


图 18 T0005 的优化路径

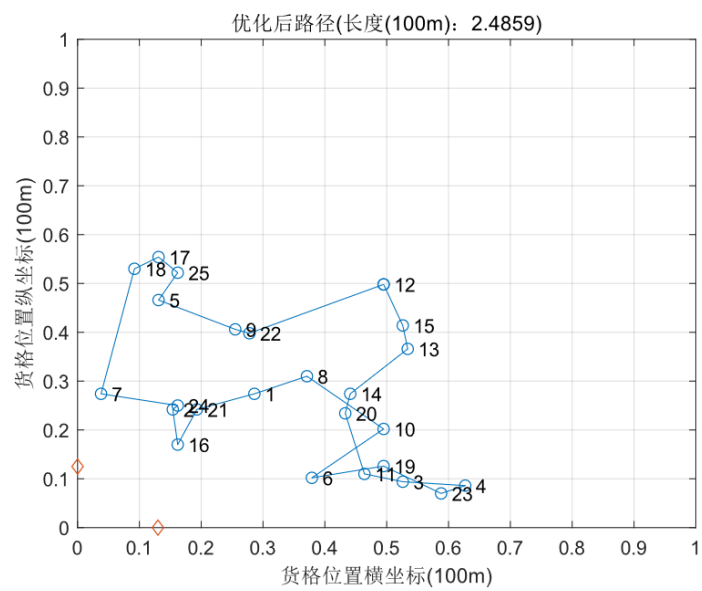


图 19 T0006 的优化路径