

# 目录

一、 问题重述 .....	1
1.1 问题背景 .....	1
1.2 问题概述 .....	1
二、 模型假设 .....	1
三、 符号说明 .....	2
四、 问题一模型的建立与求解 .....	2
4.1 问题描述与分析 .....	2
4.2 模型的建立 .....	3
4.2.1 经典旅行商模型 .....	3
4.3 模型的求解 .....	3
4.3.1 遗传算法 .....	3
4.3.2 动态赌轮 .....	4
五、 问题二模型的建立与求解 .....	5
5.1 问题描述与分析 .....	5
5.2 模型的建立 .....	5
5.2.1 带约束的多旅行商模型 .....	5
5.3 模型的求解 .....	6
5.3.1 插板式编码遗传算法 .....	6
六、 灵敏度分析 .....	7
七、 模型的评价 .....	7
7.1 模型的优点 .....	7
7.2 模型的缺点 .....	7
7.3 模型改进 .....	7
附录 A 模型的代码实现 .....	9
A.1 数据可视化–python 源代码 .....	9

# 一、问题重述

## 1.1 问题背景

在物资调运过程中，完成指定点的调运任务是最基本的要求，在完成基本的任务之外，往往有更高的追求，比如如何使总运费最省？怎样才能使得运输时间最短？如何选择运输路径使得运输总距离最短等等。这些更高的追求往往是企业期望达到的目标，为了解决这些类似问题，有必要对物资调运的过程进行数学模型的建立，以期通过模型来理解和分析物资调运的过程，并为其找到解决的方法。现以具体的食品调运案例进行分析研究。

某食品公司有 19 个食品销售点，销售点的地理坐标和每天的需求量见附件。每天凌晨都要从仓库（第 20 号站点）出发将食品运至每个销售点，运送物品后最终返回仓库。现有运送食品的运输车，每台车每日工作 4 小时，运输车重载运费 2 元/吨公里，并且假定街道方向均平行于坐标轴，任意两站点间都可以通过一次拐弯到达。

## 1.2 问题概述

围绕相关附件和条件要求，研究食品运输车在各仓库间的调度方案，依次提出以下问题：

**问题一：**若只有一辆载重 100 吨的大型运输车，运输车平均速度为 40 公里 / 小时，每个销售点需要用 20 分钟的时间下货，空载费用 0.6 元/公里。它送完所有食品并回到仓库，求最少需要时间及其对应的总距离，总运费。

**问题二：**有一种小型运输车，运输车平均速度为 50 公里 / 小时，每个销售点需要用 5 分钟的时间下货，载重为 6 吨，空载费用 0.4 元/公里；要使它们送完所有食品并回到仓库，运输车应如何调度使总体调度效率最高？

**问题三：**如果有载重量为 4 吨、6 吨两种运输车，空载费用分别为 0.2、0.4 元/公里，其他条件均相同，又如何安排车辆数和调度方案。

# 二、模型假设

- (1) 为保证预测结果精确性，假设题目所给出数据真实可信。
- (2) 假设重点防控的区域和人群中，发病、死亡人数的增长率比其基数更加重要

### 三、符号说明

符号	说明
$X^{(i)}$	人数时间序列
$a$	发展灰度
$u$	内生控制灰度

### 四、问题一模型的建立与求解

#### 4.1 问题描述与分析

问题一要求规划大型运输车的行驶路径，使得货物运输时间达到最短。该问题本质是旅行商问题，基于街道方向均平行于坐标轴，我们求解任意两点间的曼哈顿距离作为其间的距离，并设计动态赌轮遗传算法对其进行求解。

其思维流程图如图 2 所示：

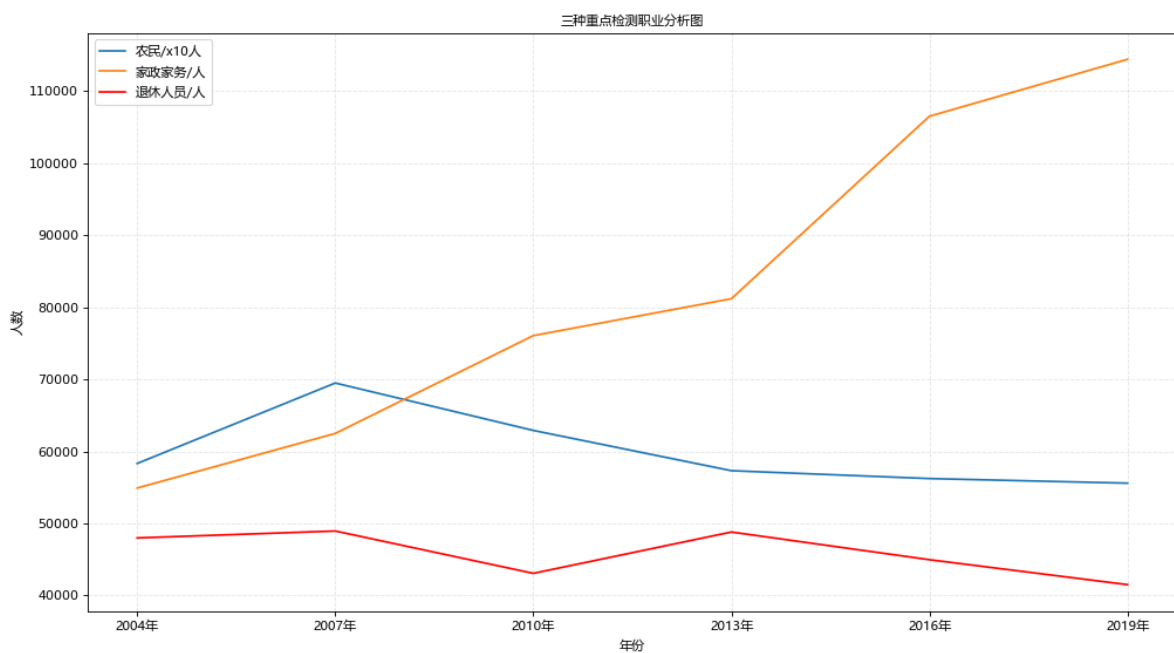


图 1 问题一思维流程图

## 4.2 模型的建立

### 4.2.1 经典旅行商模型

分析问题一，由于大型运输车的行驶速度固定，优化行驶路径使得运输车行驶路程最短时，即可求得最小运输时间。遍历路径可表示为二维有限序列如下：

$$P_n = [p_1, p_2, \dots, p_i, \dots, p_{19}] \quad (1)$$

其中  $p_i (1 \leq i \leq 19, i \in Z)$  表示处运输起点外的的仓库坐标, 且对于  $\forall i \neq j$  都有  $p_i \neq p_j$ 。任意两坐标点  $p_i(x_i, y_i), p_j(x_j, y_j)$  间的曼哈顿距离可表示为：

$$d(p_i, p_j) = |x_i - x_j| + |y_i - y_j| \quad (2)$$

将运输车行驶路程作为优化目标即可得到目标函数如下：

$$L(P_n) = d(p_0, p_1) + d(p_0, p_{19}) + \sum_{i=1}^{18} d(p_i, p_{i+1}) \quad (3)$$

即可得到整体优化模型如下：

$$L(P_n) = d(p_0, p_1) + d(p_0, p_{19}) + \sum_{i=1}^{18} d(p_i, p_{i+1}) \quad (4)$$

$$\begin{cases} 1 \leq i \leq 19, i \in Z \\ \forall i \neq j, p_i \neq p_j \end{cases} \quad (5)$$

## 4.3 模型的求解

### 4.3.1 遗传算法

**初始化编码** 对于表示为二维有限序列的遍历路径  $P_n$ ，对其进行整数编码为

$$A_n = [a_{n1}, a_{n2}, \dots, a_{ni}, \dots, a_{n19}], \quad (6)$$

$$1 \leq a_i \leq 19, i \in Z; \forall i, \neq j \Rightarrow p_i \neq p_j \quad (7)$$

定义  $A_n$  为解序列  $P_n$ ，其中  $a_{ni}$  表式对应仓库的访问顺序，例如  $ai = 7$  表示第  $i$  次访问 7 号仓库。即随机生成初始解集  $A = \{A_n\}$  其中  $n = 1, 2, \dots, w, w$  为的种群容量。

**交叉** 在原染色解集  $\{A_n\}$  中的染色体按照随机顺序配对，按照以下的方式交叉 (补全)，生成交叉解集  $\{H_n\}$

## 2. Order Crossover (顺序交叉)

### a. 在父代样本 1 中选择交换部分

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

父代 2

5	4	6	9	2	1	7	8	3
---	---	---	---	---	---	---	---	---

### b. 根据交叉部分生成子代 1

		3	4	5	6			
--	--	---	---	---	---	--	--	--

### c. 将父代样本 2 中未被选择到的基因按顺序复制到子代 1 中

9	2	3	4	5	6	1	7	8
---	---	---	---	---	---	---	---	---

5	4	6	9	2	1	7	8	3
---	---	---	---	---	---	---	---	---

然后对父代样本 2 选择交叉部分生成子代 2，将父代样本 1 中未被选择到的基因按顺序复制到子代 2 中。

图 2 交叉

**变异** 鉴于序列式染色体的特殊性，为了在变异阶段内尽可能不破坏原有的基因段，采取改良圈算法的思路进行变异操作。即在染色体  $A$  中随机选取  $a_i$  与  $a_j (1 \leq i < j \leq j)$ ，颠倒  $a_{ni}$  与  $a_{nj}$  间顺序的顺序，即：

$$M = [a_1, \dots, a_i, a_{j-1}, a_{j-2}, \dots, a_{i+1}, a_i, \dots, a_{19}] \quad (8)$$

$M$  为染色体  $A$  对应的变异染色体，对于每个染色体  $A_n$ ，都设定相同的变异概率  $\gamma$  去执行上述的变异操作。

### 4.3.2 动态赌轮

将第  $g$  代的染色体与其交叉和变异产生的子代并入同一解集  $G_g = \{A, H, M\}$ 。  $k$  表示原解集  $A$ ，交叉解集  $H$  和变异解集  $M$  中解的数量之和，即为  $G_g$  中的解的数目。设置  $G_g$  第  $i$  个解  $G_g(i)$  被选择进入下一代的概率为：

$$P(G_g(i)) = \frac{w f^{-g/\gamma}(G_g(i))}{\sum_{j=1}^k f^{-g/\gamma}(G_g(j))} \quad (9)$$

$f^{-g/\gamma}(G_g(i))$  为  $G_g(i)$  对应的目标函数值，即为  $G_g(i)$  对应的适应度。其中参数  $\gamma$  为衰减系数， $w$  为种群容量。即适应度值相对较小的解保留概率将逐渐增大，即算法初始阶

段将保留丰富度尽可能多的解，而愈到算法后期，策略就越接近于精英策略，加快算法的收敛速度，即有：

$$\lim_{g \rightarrow \infty} P(G_g(min)) = \lim_{g \rightarrow \infty} \frac{wf^{-g/\gamma}(min)}{\sum_{j=1}^k f^{-g/\gamma}(G_g(j))} \rightarrow 1 \quad (10)$$

该式表明当迭代次数  $g$  足够大时，选择策略将趋近于为精英策略，将加速算法的收敛。

## 五、问题二模型的建立与求解

### 5.1 问题描述与分析

问题二要求设计小型运输车的调度方案，从而使得总体调度效率最高。我们将时间限制设置为约束，着重优化方案的经济效率。在问题一的基础上重新建立模型，鉴于第二问的决策变量是多段序列的和，我们设计了插板编码对决策变量进行编码，并基于问题一中的动态赌轮遗传算法以运输总成本为目标进行优化。

### 5.2 模型的建立

#### 5.2.1 带约束的多旅行商模型

当雇佣  $b$  辆小型运输车时，总决策序列可表示为

$$P_n = \{[p_1, p_2]_1, [\dots, p_i]_k, [\dots]_{b-1}, [p_{19}]_b\} \quad (11)$$

即将类似于模型一中的坐标序列分割为多段，每一段分别由一辆小型运输车单独完成，定义小型运输车  $k$  的运输路径为

$$B_k = [p_i, p_{i+1}, \dots, p_{j-1}, p_j]_k \quad (12)$$

即总路径决策变量可表示为：

$$P_n = \{B_1, B_2, \dots, B_b\} \quad (13)$$

小车  $k$  的载重总和可表示为

$$W(B_k) = \sum_{p_i \in B_k} w(p_i) \quad (14)$$

其中  $w(p_i)$  为仓库  $p_i$  的订货量。其货运总时间可表示为：

$$T(B_k) = L(B_k)/50 + size(B_k)/12 \quad (15)$$

将运输成本作为目标函数可表示为：

$$\Gamma(P_n) = \sum_{k=1}^b \left[ \sum_{j=1}^{size(B_k)} (0.4 + 2 \sum_{i=1}^j w(p_i)) \times d(p_i, p_{i+1})_{p_i \in B_k} \right] \quad (16)$$

小型运输车的负载量约束可表示为

$$\max_{B_k \in P_n} W(B_k) \leq 6 \quad (17)$$

其运输时间约束可表示为时间约束可表示为

$$\max_{B_k \in P_n} T(B_k) \leq 4 \quad (18)$$

即整体模型可以表示为

$$\Gamma(P_n) = \sum_{k=1}^b \left[ \sum_{j=1}^{size(B_k)} (0.4 + 2 \sum_{i=1}^j w(p_i)) \times d(p_i, p_{i+1})_{p_i \in B_k} \right] \quad (19)$$

$$\begin{cases} \max_{B_k \in P_n} W(B_k) \leq 6 \\ \max_{B_k \in P_n} T(B_k) \leq 4 \end{cases} \quad (20)$$

### 5.3 模型的求解

#### 5.3.1 插板式编码遗传算法

由于问题中的决策变量  $P_n$  表示多个不同小型运输车的运输路径，我们将染色体变量中插入无意义基因作为隔板，例如在派遣两辆运输车时，决策变量可表示为

$$P_n = \{[p_1, p_2, \dots, p_i]_1, [p_{i+1}, \dots, p_{19}]_2\} \quad (21)$$

即在染色体中插入一个无意义基因作为挡板即可表示出染色体：

$$A_n = [a_{n1}, a_{n2}, \dots, a_{ni}, \varphi, a_{ni+1}, \dots, a_{n19}] \quad (22)$$

其中  $\varphi$  为无意义挡板基因。当雇佣小型运输车数量为  $N$  时，在染色体中插入  $N - 1$  个挡板基因即可将染色体分成  $N$  段，即能分别代表每辆小车的行驶的路径。之后将无意义挡板基因作为普通基因进行交叉，变异即可优化求解每一辆小车的行驶路径。引入冗余惩罚因子  $\theta$ ：

$$\theta(P_n) = \begin{cases} 1, (\exists (Loc(\varphi_i) + 1) = Loc(\varphi_{i+1})) \vee (\exists (Loc(\varphi_i) = 0) \vee (\exists (Loc(\varphi_i) = size(A_n))) \\ 0, otherwise \end{cases} \quad (23)$$

其中  $Loc(\varphi_i)$  表示无意义挡板基因  $\varphi_i$  在染色体  $A_n$  中的位置， $size(A_n)$  表示染色体  $A_n$  的维数。即当决策变量  $P_n$  中的挡板基因存在于其开头或末尾时，或者当两个挡板基因相邻时，表明有被雇佣车辆并没有参加运输工作，此时将惩罚因子  $\theta(P_n)$  置一，否则置零。求解第二问时，我们沿用第一问的动态赌轮遗传算法进行优化计算。即将目标函数

替换为

$$F(P_n) = \Gamma(P_n) + \theta(P_n)M_1 + \max_{B_k \in P_n} (\max W(B_k) - 6, 0)M_2 + \max_{B_k \in P_n} (\max T(B_k) - 4, 0)M_3 \quad (24)$$

其中  $M_1$ 、 $M_2$  和  $M_3$  为较大的正系数，即可达到罚函数约束功能。

## 六、灵敏度分析

## 七、模型的评价

### 7.1 模型的优点

- (1) 利用马尔可夫模型改进后的灰度预测值与实际值拟合度更高，波动性保持一致，预测的效果更好。
- (2) 针对支持向量回归参数选取，利用灰色关联度筛选合适指标，相较于主观选取指标具有客观性、严谨性。

### 7.2 模型的缺点

问题一、二中的灰色预测模型只能做短期预测，并不适用于长期预测。

### 7.3 模型改进

可以通过序列最小优化算法 (Sequential Minimal Optimization, SMO) 作为样本的训练算法，进而建立序列最小优化支持向量回归模型，从而减小算法复杂度，提高算法的求解速度。



## 参考文献

- [1] 张斯嘉, 郭建胜, 钟夫, 等. 基于蝙蝠算法的多目标战备物资调运决策优化 [J]. 火力与指挥控制, 2016, 41(1): 58-61.
- [2] 李健, 张文文, 白晓昀, 等. 基于系统动力学的应急物资调运速度影响因素研究 [J]. 系统工程理论与实践, 2015, 35(3): 661-670.
- [3] Wang J, Ersoy O K, He M, et al. Multi-offspring genetic algorithm and its application to the traveling salesman problem[J]. Applied Soft Computing, 2016, 43: 415-423.
- [4] 陶丽华, 马振楠, 史朋涛, 等. 基于 TSP 问题的动态蚁群遗传算法 [J]. 机械设计与制造, 2019 (12): 39.

## 附录 A 模型的代码实现

### A.1 数据可视化–python 源代码

```
_xtick_labels = ["{}年".format(int(i)) for i in x]
plt.xticks(x, _xtick_labels, fontproperties=my_font)
# plt.yticks(range(0, 9))

# 绘制网格
plt.grid(alpha=0.3, linestyle="--") # alpha为透明度 0-1
plt.title("三种重点检测职业分析图", fontproperties=my_font)
plt.xlabel("年份", fontproperties=my_font)
plt.ylabel("患病人数", fontproperties=my_font)
# 标注图例
plt.legend(prop=my_font, loc=0)
plt.show()
```