

# 邮政运输网络中的邮路规划和邮车调度

## 一、问题重述

邮政运输问题是邮政生产过程四大环节的物质基础。时限与成本是邮政运输问题的两个重要指标，时限是指邮件、报刊处理、传递的最大时间限制，是邮车调度需要满足的基本要求，成本影响着企业的经营，包括道路成本以及空车成本，在邮路设计时，在满足时限的前提下，需要使成本最小。时限和成本对于邮路规划和邮车调度有着重要的影响。

中国的邮政运输网络采用以邮区中心局作为基本封发单元和网络组织的基本节点，负责处理、封发、运输邮件，在此基础上组织分层次的邮政网。邮路是邮政运输网络的基本组成单元，它是指利用各种运输工具按固定班期、规定路线运输邮件，并与沿线有交接频次的邮政局、所交换邮件总包所行驶的路线。

本文中要考虑的问题是：某地区的邮政分为地市局、县局和支局三级机构，该地区的邮政运输网络由区级邮政运输网和县级邮政运输网构成。区级邮政运输网由从地市局出发并最终返回地市局的区级邮车所行驶的全部邮路构成，县级邮政运输网由从县局出发并最终返回县局的县级邮车所行驶的全部邮路构成。该地区地市局为 D，周围共有 5 个县局 X1，……，X5，每个县局包含若干个支局 Z1，……，Z73。区级邮政运输网至少负责收发 5 个县局以及所在地市的 16 个支局 Z58, Z59, ……，Z73 的邮件；各县局邮政运输网必须覆盖本县内区级邮车不能到达的支局。见图 1，红线为区级邮政运输网，黑线为县级邮政运输网。区级邮政运输网贯穿各个县局，收寄邮件；县级邮政运输网贯穿本县支局。邮件的流动方向如图 2 所示，箭头表示邮件的流向，不表示实际路径。

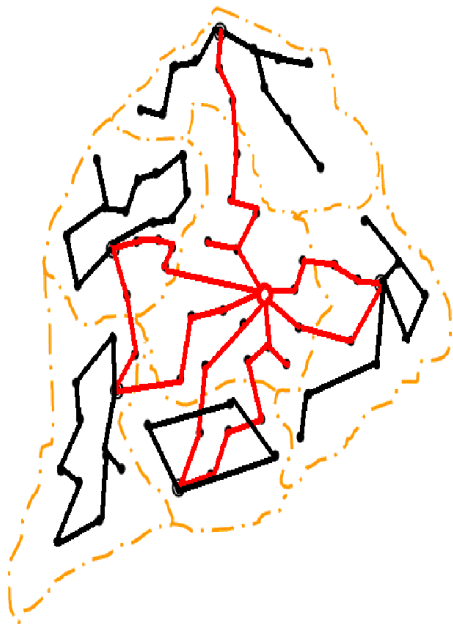


图 1

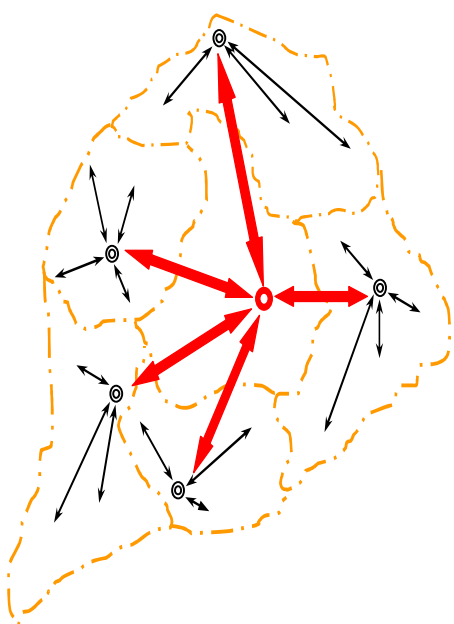


图 2

该地区从地市局到县局每天两班车，从县局到支局每天仅有一班车。该地区的邮政运输流程如下：

- Step1: 区级第一班次邮车从地市局 D 出发将邮件运送到各县局  $X_i$  和沿途支局, 并将各县局  $X_i$  和沿途支局收寄的邮件运送回地市局 D;
- Step2: 区级邮车离开后, 县局  $X_i$  将当天区级第一班次邮车及前一天的区级第二班次邮车所送达的本县邮件进行集中处理, 按寄达支局装上相应的县级邮车;
- Step3: 各县级邮车将邮件运送到其负责的支局并将这些支局收寄的邮件运送回县局  $X_i$ ;
- Step4: 区级第二班次邮车从地市局 D 出发将邮件运送到各县局  $X_i$  和沿途支局, 并将各县局  $X_i$  收寄的邮件和沿途支局收寄的邮件运送回地市局 D。

这个问题里, 对县局  $X_i$ , 它的时限包括: 区级第一班邮车开走一小时后才可发出县级邮车, 沿途在各支局耗时 5 分钟, 县级邮车须在区级第二班邮车开走一小时前返回。对于地市局 D, 它的时限包括: 第一班邮车出发时间必须在 6:00 之后, 沿途在各支局耗时 5 分钟, 在各县局需要等到县级邮车返回一小时之后才能离开, 返回地市局 D 必须在 11:00 之前。当邮车运载能力有限时, 每辆邮车在各个支局、省局装载后都不应超过其运载能力。

## 二、模型假设

1. 区级两个班次邮车的行驶路线相同, 行驶时间及费用也相同。
2. 区级邮政运输网必须至少覆盖该地市附近的 16 个支局 Z58, Z59, …… , 和 5 个县局 X1, …… , X5。
3. 各县邮政运输网必须覆盖本县内区级运输车不到达的区域。
4. 县级邮车平均时速为 30km/h, 区级邮车的平均速度为 65km/h, 邮车在各支局装卸邮件耗时 5 分钟, 在各县局装卸邮件耗时 10 分钟, 不考虑任何突发事件。
5. 邮车在邮路上相邻两点邮局间行使时只走最短路径。
6. 第一问中假设 X1 区每天每个支局收发的邮件数保持不变。

## 三、问题一模型与求解

### 3.1 模型的建立

对于这个问题我们只关心完成这个县邮运任务的最少的邮车数量以及在最少的邮车数量实现邮路总长度最短或者总耗时间最短。针对此问题我们假设第一辆地市区邮政运输车不到达 X1 内的任何支局。在这个假设下要求县邮车需要到达 X1 内所有的支局。在这个问题中我们理解的每天邮车只有一个班次指的是允许多辆车在同一时间内出发, 但每天只允许出发一次。每辆邮车的最大装载量不超过 63, 最长行使时间不超过 6 小时。

设 X1 县内共有  $N$  条邮路, 分别为

$$\{X1, Y_{1,1}, Y_{1,2} \cdots Y_{1,n_1}, X1\}, \{X1, Y_{2,1}, Y_{2,2} \cdots Y_{2,n_2}, X1\}, \cdots, \{X1, Y_{N,1}, Y_{N,2} \cdots Y_{N,n_N}, X1\}, \quad n_i \text{ 表示}$$

第  $t$  条邮路上一共需要收发邮件邮支局的总数且  $\sum_{t=1}^N n_t = 16$  (每个支局都要经过),

$Y_{i_1, j_1} \neq Y_{i_2, j_2}$  if,  $i_1 \neq i_2$  and  $j_1 \neq j_2$ , 这里我们的邮路是指一辆邮车按顺序经过并收发邮件的支局序列, 序列两端加上出发总局。  $Zr(i)$ ,  $Zs(i)$  分别表示第  $i$  个邮局接收与发出去的邮件,  $T_k, R_k$  分别表示第  $k$  条邮路中邮车全程包括收发邮件的总时间与邮车全程中最重时刻的装载量,  $D(i, j)$  表示第  $i$  个邮局到第  $j$  个邮局的最短距离则,  $W_{k,t}$  表示第  $k$  条邮路在第  $t$  个邮局时的装载量,  $W_{k,0}$  表示出发时的装载量则:

$$T_k = \frac{\sum_{i=2}^{n_k} D(Y_{k,n_{i-1}}, Y_{k,n_i}) + D(X1, Y_{k,1}) + D(Y_{k,n_k}, X1)}{30} + \frac{5}{60} \cdot n_k$$

$$W_{k,t} = \sum_{j=1}^{n_k} Zr(Y_{k,n_j}) + \sum_{i=1}^t Zs(Y_{k,n_i}) - \sum_{j=1}^t Zr(Y_{k,n_j}), \quad t=1, 2, \dots, n_k, \quad W_{k,0} = \sum_{j=1}^{n_k} Zr(Y_{k,n_j}), \text{ 显然}$$

$$\text{有 } R_k = \max \{W_{k,0}, W_{k,1}, \dots, W_{k,n_k}\}$$

显然在保证邮路最少的情况下, 邮路最短时我们有多目标规划模型

$$\text{Min} Z = w_1 N + w_2 \sum_{k=1}^N T_k$$

$$\text{st.} \begin{cases} \max \{R_1, R_2, \dots, R_N\} \leq 65 \\ \max \{T_1, T_2, \dots, T_N\} \leq 6 \\ \sum_{t=1}^N n_t = 16 \\ Y_{i_1, j_1} \neq Y_{i_2, j_2} \quad \text{if, } i_1 \neq i_2 \text{ and } j_1 \neq j_2 \\ w_1 \gg w_2, \quad w_1 > 0, w_2 > 0 \end{cases}$$

最少空车损失的模型为

$$\text{Min} E =$$

$$2 \sum_{k=1}^N \left[ \left( \sum_{t=1}^{n_k-1} (65 - W_{k,t}) D(Y_{k,t}, Y_{k,t+1}) + (65 - W_{k,0}) D(X1, Y_{k,1}) + (65 - W_{k,n_k}) D(Y_{k,n_k}, X1) \right) \right]$$

$$\text{st.} \begin{cases} \max \{R_1, R_2, \dots, R_N\} \leq 65 \\ \max \{T_1, T_2, \dots, T_N\} \leq 6 \\ \sum_{t=1}^N n_t = 16 \\ Y_{i_1, j_1} \neq Y_{i_2, j_2} \quad \text{if, } i_1 \neq i_2 \text{ and } j_1 \neq j_2 \end{cases}$$

这里由于问题的规模比较小，我们这里及以后各问题都利用枚举法求最小的  $N$ 。

### 3. 2 模型求解

#### 3. 2. 1. 粒子群优化算法 (PSO) 简介

PSO 是从模拟鸟群的捕食行为中得到启示的算法。设想这样一个场景：一群鸟在随机搜索食物。在这个区域里只有一块食物。所有的鸟都不知道食物在那里。但是他们知道当前的位置离食物还有多远。那么找到食物的最优策略是什么呢。最简单有效的就是搜寻目前离食物最近的鸟的周围区域。PSO 从这种模型中得到启示并用于解决优化问题。PSO 中，每个优化问题的解都是搜索空间中的一只鸟。我们称之为“粒子”。所有的例子都有一个由被优化的函数决定的适应值，每个粒子还有一个速度决定他们飞翔的方向和距离。然后粒子们就追随当前的最优粒子在解空间中搜索。PSO 初始化为一群随机粒子(随机解)。然后通过叠代找到最优解。在每一次叠代中，粒子通过跟踪两个“极值”来更新自己。第一个就是粒子本身所找到的最优解。另一个极值是整个种群目前找到的最优解。在附录中我们将详细介绍 PSO 问题的优化技术。

#### 3. 2. 1 粒子群优化算法求解带有约束条件的多回路组合优化问题

##### 3. 2. 2. 1 邮路的编码

本文中，路径使用整数型编码方法表示。把所有的邮路都写成一行，不同的邮路中间用 0 隔开。  $N$  条邮路

$\{X1, Y_{1,1}, Y_{1,2} \cdots Y_{1,n_1}, X1\}, \{X1, Y_{2,1}, Y_{2,2} \cdots Y_{2,n_2}, X1\}, \cdots, \{X1, Y_{N,1}, Y_{N,2} \cdots Y_{N,n_N}, X1\}$  可以编码

成:  $Y_{1,1}, Y_{1,2} \cdots Y_{1,n_1}, 0, Y_{2,1}, Y_{2,2} \cdots Y_{2,n_2}, 0 \cdots, 0, Y_{N,1}, Y_{N,2} \cdots Y_{N,n_N}$ 。例如，对本文中县局  $X1$ ,

该地区支局有 16 个，若最少需要 3 辆车运输邮件，编码 3 2 1 13 12 11 0 10 8 7 6 5 4 0 14 9 16 15，表示的邮路为：

第 1 辆车:  $X_1 \rightarrow Z_3 \rightarrow Z_2 \rightarrow Z_1 \rightarrow Z_{13} \rightarrow Z_{12} \rightarrow Z_{11} \rightarrow X_1$

第 2 辆车:  $X_1 \rightarrow Z_{10} \rightarrow Z_8 \rightarrow Z_7 \rightarrow Z_6 \rightarrow Z_5 \rightarrow Z_4 \rightarrow X_1$

第 3 辆车:  $X_1 \rightarrow Z_{14} \rightarrow Z_9 \rightarrow Z_{16} \rightarrow Z_{15} \rightarrow X_1$

##### 3. 2. 2. 2 粒子适应度函数的定义

当  $N$  给定时，我们直接利用目标函数中  $\sum_{i=1}^N T_k$  的倒数来定义粒子适应度函数

发现，当我们随机取初始群体时，由于可行解的数量相对于任意解的数量非常少，一般来讲，所有的粒子都不符合约束条件，这样我们利用目标函数来定义粒子适应度函数，一般最后的结果都是不满足约束条件的解。所以我们首先要解决的问题是怎么设计粒子适应度函数来实现粒子从非可行解向可行解过度。

**粒子适应度函数定义：** 设某粒子  $i$  由  $m$  条邮路组成，假设强制完成这  $m$  所需要的各自所需时间为  $T_1, T_2, \cdots, T_n$ ，各自全程最大负荷为  $R_1, R_2, \cdots, R_n$ ，设函数

$p(x-\beta)=\begin{cases} x-\beta, & \text{if } x>\beta \\ 0 \end{cases}$  则，粒子 $i$ 的适应度函数为：

$$f(i)=\frac{1}{1+\xi_1\cdot\sum_j^m f(T_j-6)+\xi_2\cdot\sum_j^m f(R_j-65)}, \quad \xi_1, \xi_2 \text{ 都大于 } 0。$$

其中， $\sum_j^m f(T_j-6)+\sum_j^m f(R_j-65)$  可以看成是一个一个粒子（解）到可行解集合的距离。

上述公式引入 $\xi_1, \xi_2$ 的目的是避免 $R_j, T_j$ 的值相差太大而引起的信息损失。怎么确定 $\xi_1, \xi_2$ 的值呢？我们在 $m$ 给定下随机抽取了大量样本计算了 $R_j, T_j$ 的平均值，我们发觉它们一般都相差10倍左右，（和速度一个数量级），所以我们一般让 $\xi_1=10\xi_2$ 。 $\xi_1, \xi_2$ 具体的值要根据最终优化的目标函数的值的范围而定。在实际的操作中我们发现，这样定义的适应度函数可以非常高效的找到可行解，这样我们就可以首先建立一个大的可行解的数据库，每次从数据库中提取部分解，和随机产生的部分解一起作为初试种群，利用如下适应度函数

$$F(i)=\frac{1}{1+10000\cdot\sum_j^m f(T_j-6)+1000\cdot\sum_j^m f(R_j-65)+\sum_j^m T_j},$$

就可以进行总时间（总邮路长，总运输费用）的最终寻优了。

$$F(i)=\frac{1}{1+10000\cdot\sum_j^m f(T_j-6)+1000\cdot\sum_j^m f(R_j-65)+E}$$

这里 $E$ 表示空车的损失。

### 3. 2. 2 求解结果

定义：总成本=空车损失+总运行成本（3元/公里）

我们分别给出了空车率损失最少（方案1）与总成本最少（方案2）的两个方案

方案1：空车率损失最小

	邮路	空车率损失	运行费用	最大运输量	耗时
邮车1	X1-6-5-7-(8)-16-10- X1	13.785	417	65	5.05
邮车2	X1-13-1-2-3-4-(10)-11- X1	17.169	492	64	5.6667
邮车3	X1-(10)-9-8-15-(14)-12-14- X1	33.446	417	62	5.05

在这种方案下总的由空车率造成的损失为64.4（元），为所有路径中最小的。总运行费用为1326（元）。

方案 2: 总成本最小

	邮路	空车率 损失	运行 费用	最大运 输量	耗时
邮车 1	X1-3-2-1-13-12-11- X1	53.354	489	59	5.933
邮车 2	X1-10-8-7-6-5-4- X1	22.8	315	63	4
邮车 3	X1-14-9-16-15- X1	21.938	273	61	3.667

在这种方案下总的由空车率造成的损失为 98（元），总运行费用为 1077（元）。

可以算出方案 1 的总成本为 1390.4（元），方案 2 的总成本为 1175（元）因此可以看出方案 2 比方案 1 在考虑了运输成本时更好，可以节约成本 215.4（元）。所以我们认为，只考虑空车损失是不优化的，我们认在邮车数量确定下没有必要考虑空车损失。

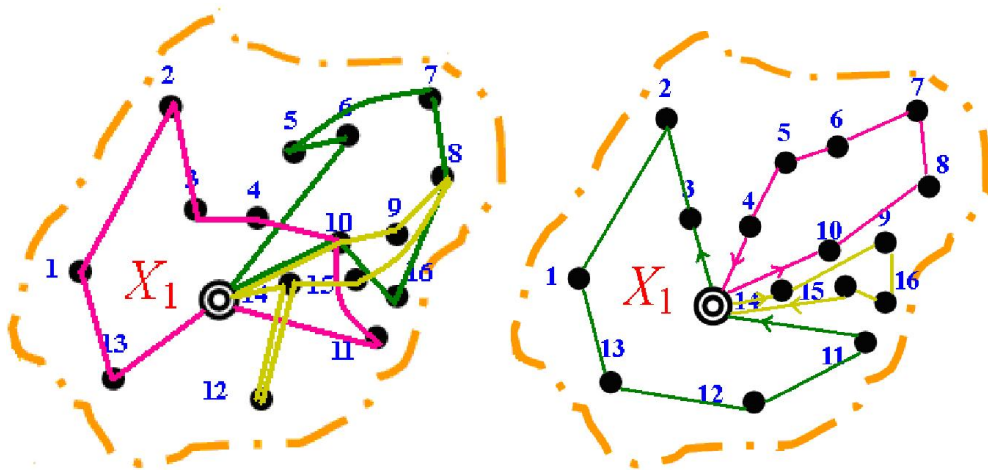


图 3: 方案 1 邮路图

方案 2 邮路图

## 四、问题二模型与求解

### 4. 1 模型的建立

对于这个问题我们只关心完成这个区邮运任务的最少的邮车数量以及在最少的邮车数量实现邮路总长度最短或者总耗时间最短。假设市局对 D 于  $X_i$  县运送邮件的邮车已经安排好了，地市局第一班次邮车到达  $X_i$  并卸装完邮件的时间为  $t_1$ ，从  $X_i$  返回到地市局的时间为  $t_2$ ，通过  $X_i$  县的地市局第一班次邮车邮路总时长为  $t = t_1 + t_2$ ，那么  $X_i$  县可用于运输的时间为  $T$  则：

$$T = (18 - t_2 - 1) - (6 + t_1 + 1) = 10 - t$$

如图所示：

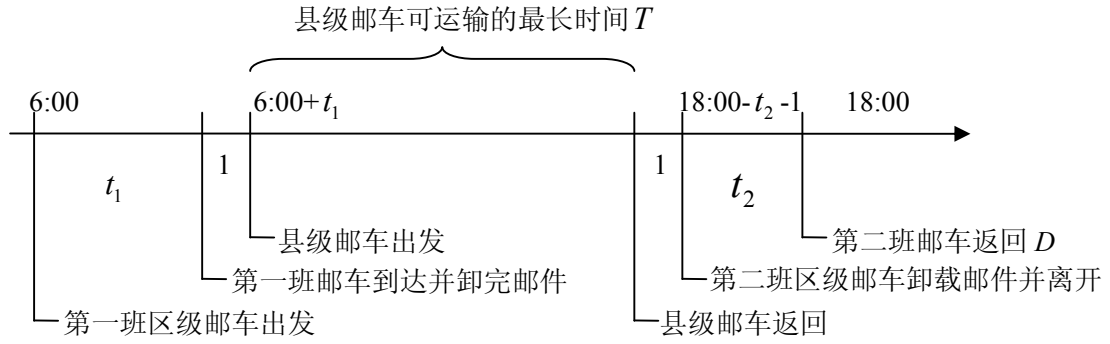


图 4：县级邮车可用时间与区级邮车邮路运行时间关系图

这说明地市局的邮车在邮路上运行的时间越短对县局越有利。

从而我们的优化原则简化为：首先，把地市支局与县局（不包括县支局）放在一起，看成一个多回路组合优化问题，其目标是尽可能用最少的车，总邮路最短而且每条邮路的时间要限制在 5 个小时内。其次，在地市支局与县局邮路优化好了之后，就能利用上面公式算出每个县局可用于运输的时间，从而各县的优化问题都转化成了一个带有时间限制的多回路组合优化问题。带有时间限制的多回路组合优化问题比问题一在形式上简单。这里我们不分别对市、县给出模型，我们只以某县为例给出一个统一的模型。对于市局的优化问题上，只需目标函数与约束条件要多考虑收发邮件在每个县局多 5 分钟与速度是 65 千米/小时且每辆车的可利用时间为 5 小时。

设：某县待收发邮件的支局为共有  $h$  个，县局为  $X$ ，需要  $N$  辆车运输（对应  $N$  条邮路），且可用于运输的时间为  $T$  则。分别为：

$\{X, Y_{1,1}, Y_{1,2} \cdots Y_{1,n_1}, X\}, \{X, Y_{2,1}, Y_{2,2} \cdots Y_{2,n_2}, X\}, \cdots, \{X, Y_{N,1}, Y_{N,2} \cdots Y_{N,n_N}, X\}$ ,  $n_t$  表示第  $t$  条

邮路上一共需要收发邮件邮局的总数且  $\sum_{t=1}^N n_t = h$ （每个支局都要经过），

$Y_{i_1, j_1} \neq Y_{i_2, j_2}$  if,  $i_1 \neq i_2$  and  $j_1 \neq j_2$ ，这里我们的邮路是指一辆邮车按顺序经过并

收发邮件的支序列，两端是出发总局。  $T_k$  分别表示第  $k$  条邮路中邮车全程包括

收发邮件的总时间与邮车全过程中最重时刻的装载量，  $D(i, j)$  表示第  $i$  个邮局到第

$$j \text{ 个邮局的最短距离则, 则: } T_k = \frac{\sum_{i=2}^{n_k} D(Y_{k, n_{i-1}}, Y_{k, n_i}) + D(X, Y_{k, 1}) + D(Y_{k, n_k}, X)}{30} + \frac{5}{60} \cdot n_k$$

显然在保证邮路最少的情况下，邮路最短时我们有模型

$$MinZ = w_1 N + w_2 \sum_{k=1}^N T_k$$

$$st. \begin{cases} \max \{T_1, T_2, \dots, T_N\} \leq T \\ \sum_t^N n_t = h \\ Y_{i_1, j_1} \neq Y_{i_2, j_2} \quad \text{if } i_1 \neq i_2 \text{ and } j_1 \neq j_2 \\ w_1 \gg w_2, \quad w_1 > 0, w_2 > 0 \end{cases}$$

同样我们对  $N$  的优化采用枚举法，邮路长度采用 POS 算法，所有的情况都与问题一的求解一样，只是不考虑邮车最大负重的影响。寻找可行解数据库的时候我们

们利用适应度函数为： $f(i) = \frac{1}{1 + \sum_j^m f(T_j - 6)}$ 。在有可行解数据库的支持下

寻找最有解的适应度函数为： $F(i) = \frac{1}{1 + 1000 \cdot \sum_j^m f(T_j - 6) + \sum_j^m T_j}$ 。

## 4.2 求解结果：

### 区级邮车路线

	到达县	邮路	路程	耗时
邮车 1	X1, X2	D-62-9-10-X1-12-X2-18-63-66-67-D	252	4.8769
邮车 2	X3	D-68-65-64-X3-31-30-34-35-71-D	269	4.9718
邮车 3	X4	D-72-73-42-41-X4-36-70-69-D	186	3.6115
邮车 4	X5	D-61-59-52-X5-53-58-60-D	269	4.8051

总路程 976 千米，总耗时 18.265 小时。

### 县级邮车路线

县编号	可用时间	邮路编号	邮路	路程 (千米)	耗 时 (小时)
X1	5.1231	邮车 1	X1-14-X1	36	1.2833
		邮车 2	X1-3-4-2-1-13-X1	137	4.9883
		邮车 3	X1-5-6-7-8-16-15-11-X1	134	5.05
X2	5.1231	邮车 1	X2-17-19-26-20-X2	141	5.0333
		邮车 2	X2-22-23-24-25-21-X2	127	4.65
X3	5.0282	邮车 1	X3-27-X3	68	2.35
		邮车 2	X3-28-29-32-33-X3	134	4.8
X4	6.3885	邮车 1	X4-37-38-39-40-43-X4	140	5.0833
X5	5.1949	邮车 1	X5-54-55-56-57-X5	132	4.7333
		邮车 2	X5-51-46-44-X5	144	5.05
		邮车 3	X5-50-49-48-45-47-X5	133	4.85



县级邮路总长度 1326 千米，总耗时 47.871 小时。

总结于下表：

总邮 车数	区级：4 辆	15 辆
	县级：11 辆	
总路程=2×区级路程+县级路程		3278 千米

邮路示意图：

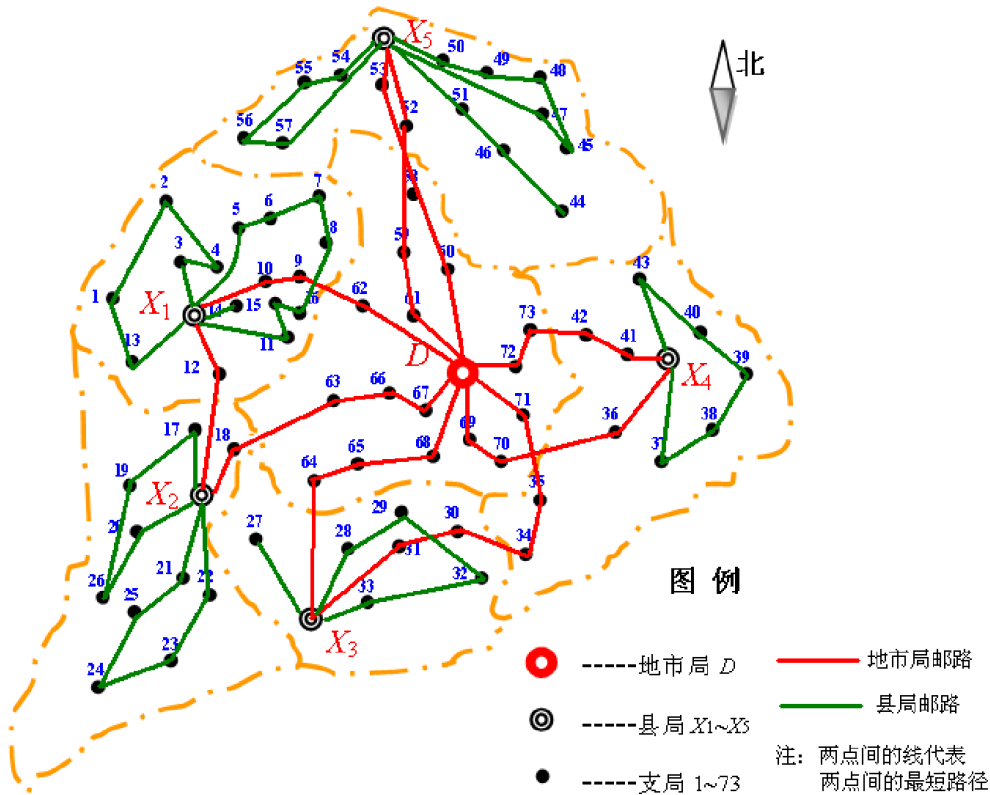


图 5：不跨县运输最短路程邮路图

由于公路可以重复行使，因此会出现一个邮局有多条邮路经过的情况。为了更清楚的表示出在一条邮路上哪些点是装卸邮件的，图中所示直线只代表两点间的最短距离，不代表公路。

## 五、问题三模型与求解

### 5.1 模型的建立

该问题以第二道题为基础，在使邮路尽量少、尽量短的基础上考虑如何通过重新规划区域进一步优化邮路。当允许在一定程度上打破行政区域限制的时候，首先考虑边界地带支局的划分。这里采用聚类的方法来重新划分 5 个行政区域，不包含地市局 D 的重新划分。设支局  $Z_i$  到县局  $X_j$  的距离  $D(Z_i, X_j)$  为  $Z_i$  到  $X_j$

的最短路径距离， $i=1, \dots, 57$ ， $j=1, \dots, 5$ 。我们将距离作为重新划分行政区域的依

据。

聚类过程如下：

- (1) 计算支局  $Z_i$  到各县局的距离  $Dis[Z_i, X_j], j = 1, \dots, 5$ ;
- (2) 找到距离取最小值  $Min[Dis[Z_i, X_j]]$  时所对应的  $j$ ，将  $Z_i$  与  $X_j$  归为一类；
- (3) 转 (1)，计算下一支局与各县局的最短距离，直到所有支局都划分到各个县局；

通过考查最短距离，聚类结果如下图所示：

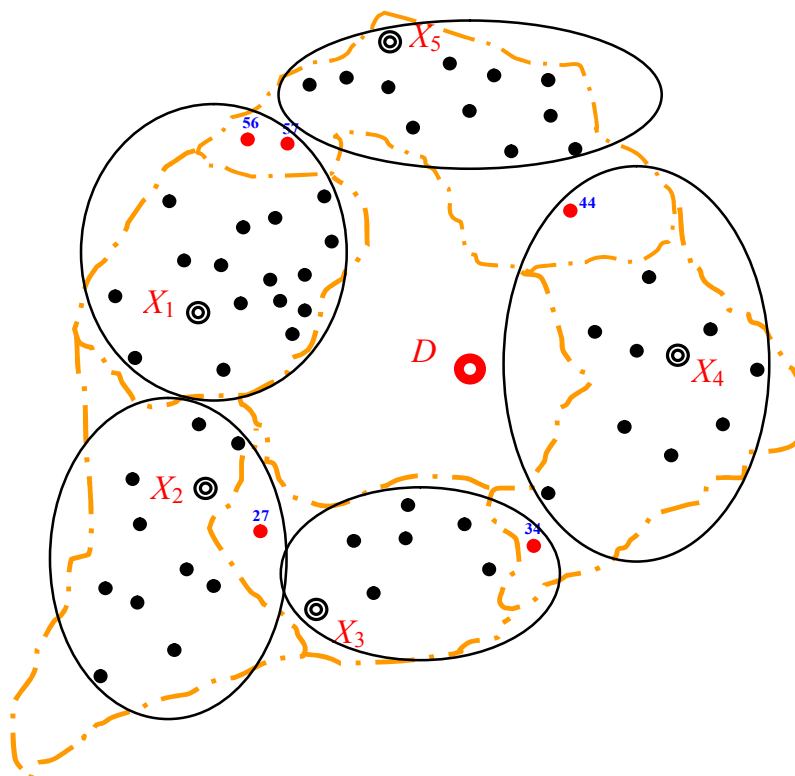


图 6：聚类结果

可见，原来区域的划分基本没有改变，但是对于边界地带的支局，则被分到了不同的区域。Z56 和 Z57 由原来的 X5 区域划分到了 X1 区域；Z27 由原来的 X2 划分到了 X3；Z34 由原来的 X4 区域划分到了 X3 区域；Z44 由原来的 X5 区域划分到了 X4 区域。

从聚类图上，我们可以得到四个有益的建议：

(1) 通过第二题的解答，我们知道由 X3 有一条单独邮路通往 Z27，如果能够用 X2 区域内的某条邮路将 Z27 覆盖，则会减少 X3 内的邮路数量，同时要使 X2 区域内的邮路保证时间限制。

(2) 对于 Z44，因为其与 X5 的距离很远，因此，在 X5 区域内为了能够将邮件输送到 Z44 而特意开辟出了 X5-51-46-44 这样一条辐射型邮路。若可以将该支局转移到县局 X4 进行邮件处理，X5 地区东部的若干个支局将可以用一条邮路串联起来，从而减少邮路量。此时，也要保证 X4 区域内的邮路依旧满足时间限制，我们可以通过调整区级前往 X4 区域的邮车路线来实现，适当缩短路线以增大 X4

区域内邮车的可运输时间。

(3) 对于 Z34，它与 X4 的最短距离为 71，最短路径为 X4-36-35-34，而 Z34 与 X3 的最短路径为 X3-31-30-34，距离为 62，但由于 X3-31-30-34 已经在区级邮路上，因此 Z34 是否适合划分到 X3 并不确定。

(4) Z56 和 Z57 由原来的 X5 区域划分到了 X1 区域，Z56、Z57 到 X5 的最短距离分别为 64，55，到 X1 的最短距离分别为 57，52，距离差异并不大。并且对于 X5，其邮路至少要 3 条，Z56、Z57 两点已在最短路径上，而若将 Z56、Z57 两加入到 X1 中，形成的新的邮路是否满足时间要求不确定。因此调整 Z56 和 Z57 到 X1 区域是否可以使总路程变短，需要通过程序演化计算得出。

### 5.2 求解结果

区级邮车路线

	到达县	邮路	路程	耗时
邮车 1	X1, X2	D-62-9-10-X1-12-X2-18-63-66-67-D	252	4.8769
邮车 2	X3	D-68-65-64-X3-31-34-35-70-69-D	269	4.9718
邮车 3	X4	D-72-73-42-41-X4-37-36-71-D	186	3.6115
邮车 4	X5	D-61-59-52-X5-58-60-D	265	4.6603

总路程 972 千米。

县级邮车路线

县编号	可用时间	邮路编号	邮路	路程 (千米)	耗 时 (小时)
X1	5.1231	邮车 1	X1-14-X1	36	1.2833
		邮车 2	X1-3-4-2-1-13-X1	137	4.9883
		邮车 3	X1-5-6-7-8-16-15-11-X1	134	5.05
X2	5.1231	邮车 1	X2-17-19-26-20-X2	141	5.0333
		邮车 2	X2-27-22-23-24-25-21-X2	138	5.1
X3	5.0282	邮车 1	X3-28-29-30-32-33-X3	134	4.8833
X4	6.3885	邮车 1	X4-44-43-40-39-38-X4	161	5.7833
X5	5.3397	邮车 1	X5-54-55-56-57-53-X5	140	5.0833
		邮车 2	X5-50-49-48-47-45-46-51-X5	137	5.15

县级邮路总长度 1158 千米。

总结于下表：

总邮 车数	区级：4 辆	13 辆
	县级：9 辆	
总路程=2×区级路程+县级路程		3102 千米

在打破县界的情况下，经过优化得到总邮车数减少了 2 辆，并且可以缩短 176 千米，节省 528 元运费。其中路线变化如下图所示：

变化的邮路示意图：

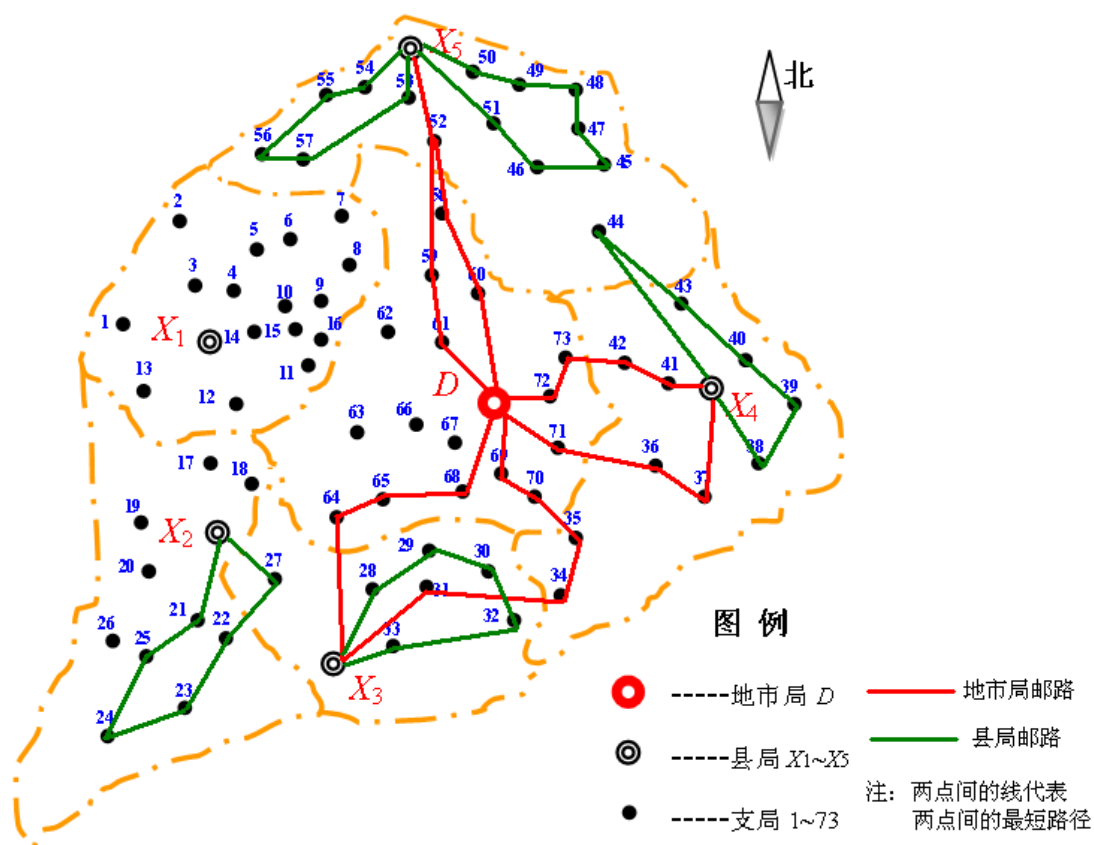


图 7：允许跨县运输后优化结果中改变的邮路图

## 六、问题四模型与求解

### 6.1 模型的建立

首先分别对 X1 至 X5 五个县找新的中心，方法为在一个县的区域内选择一个点使其他的点到他的最短路径之和最小。结果显示为 X1 区的新中心为 Z4，X2 的新中心为 Z21，X3 的新中心为 Z31，X4 的新中心不变，X5 的新中心为 Z51。从新应用问题 2 中的算法进行计算。

然后分别对更改的四个新中心求出到地市局的距离然后与原县局到地市局的距离进行比较。结果如下表所示：

县	中心	距离
X1	原址 X1	92
	新中心 Z4	87
X2	原址 X2	89
	新中心 Z21	109
X3	原址 X3	98
	新中心 Z31	66
X5	原址 X5	124
	新中心 Z51	93

从表中可以看出，除 X2 以外其他三个新中心到地市局的距离都小于旧址到地市局的距离。但 X1 县的变化很小，X3 和 X5 变化较大。由于 X1 和 X2 共用一辆区级邮车，如果 X1 迁址，那么将会加大此辆区级邮车的路程，从而减少了县级邮车可利用的时间。从而将不满足时间限制条件。因此应首先考虑 X3 和 X5 的迁址。进一步优化后的结果仍要覆盖原区县邮路覆盖的所有邮局。于是目标变为在一个小区域求解在一定约束条件下的最短路径问题。

设  $X_i$  县的原邮路条数（区级邮车数加上县级邮车数）为  $n_i$ ，模型为

$$\text{Min}L = L_{0 \rightarrow i} + L_i \quad i = 3, 5$$

$$S.T. \begin{cases} \text{新方案邮车数辆等于 } n_i \\ n_i \text{ 条新邮路需覆盖原邮路上的所有点（邮局）} \end{cases}$$

### 6.2 求解结果

区级邮车路线

	到达县	邮路	路程	耗时
邮车 1	X1, X2	D-62-9-10-X1-12-X2-18-63-66-67-D	252	4.8769
邮车 2	X3 (31)	D-68-31-30-32-34-35-70-69-D	173	3.4115
邮车 3	X4	D-72-73-42-41-X4-37-36-71-D	186	3.6115
邮车 4	X4 (52)	D-61-59-52- 58-60-D	203	3.6231

总路程 814 千米。

县级邮车路线

县编号	可用时间	邮路编号	邮路	路程 (千米)	耗 时 (小时)
X1	5.1231	邮车 1	X1-14-X1	36	1.2833
		邮车 2	X1-3-4-2-1-13-X1	137	4.9883
		邮车 3	X1-5-6-7-8-16-15-11-X1	134	5.05
X2	5.1231	邮车 1	X2-17-19-26-20-X2	141	5.0333
		邮车 2	X2-27-22-23-24-25-21-X2	138	5.1
X3 (31)	6.5885	邮车 1	31-29-65-64-X3-33-28-31	166	6.033
X4	6.3885	邮车 1	X4-44-43-40-39-38-X4	161	5.7833
X5 (52)	6.3769	邮车 1	52-53-54-55-56-57-52	163	5.85
		邮车 2	52-50-49-48-47-45-46-51-52	133	5.0167

县级邮路总长度 1209 千米。

总结于下表：

总邮 车数	区级：4 辆	13 辆
	县级：9 辆	
总路程=2×区级路程+县级路程		2837 千米

迁址后的优化结果显示，邮车运输的总路程长度每天可以减少 265 千米，即可节约运费 795 元/天。

## 2.3 县局迁址建议书

作为邮件运输行业，我们的目标是以最少的成本完成邮件运输任务。因此合理的优化的选址及邮路设计工作是非常必要的。其中选择邮局地址尤为重要，特别是那些具有中转功能的邮局地址。对于本市来说，从邮局分布图上可以看出，地市局与各个县局相距都比较远，这就为地市局的邮件运输工作带来了不便。因此为了能够尽可能的减少邮件运输成本，希望可以对部分县局的位置进行调整，以达到全局优化的目的。

图中的县 X3 是选址最不合理的县局。X3 县内的支局主要分布在偏东北方向，而县局却设在了西南方向。这导致了它到各个支局的距离非常大，造成了邮件运输过程中的费用损失。因此非常不合理。县 X3 中的点  $Z_{31}$  是整个县的中心，即所有其他点到他的距离之和最短。又因为它与地市局的距离较 X3 近很多，因此将县局迁址到此会节省相当一部分的邮件运输费用。迁址后的最佳邮路为：

◎区级邮路：D-68-31-30-32-34-35-70-69-D      总路程 173 千米

◎县级邮路：31-29-65-64-X3-33-28-31      总路程 166 千米

这样会比原来的旧地址旧邮路每天节省 160 千米的路程，即可节省 480 元运费。

本市除了可以在 X3 县迁址外，还可以在 X5 县迁址。因为在 X5 县，支局分布比较松散，一条过县局的南北向直线将其分为东、西方向两个部分。在这条直线附近支局  $Z_{52}$  与地市局的距离最近，如果以这个点为县局新址的话，也可以减少相当的运输成本。以  $Z_{52}$  为新址的最优邮路为：

◎区级邮路：D-61-59-52- 58-60-D      总路程 203 千米

◎县级邮路 1：52-53-54-55-56-57-52      总路程 163 千米

◎县级邮路 2：52-50-49-48-47-45-46-51-52      总路程 133 千米

比原来的县局旧地址旧邮路每天可以节省 105 千米的路程，即可节省 315 元运费。

如果将这两处县局地址更改后，每天可节省 795 元运输费用，相当可观。

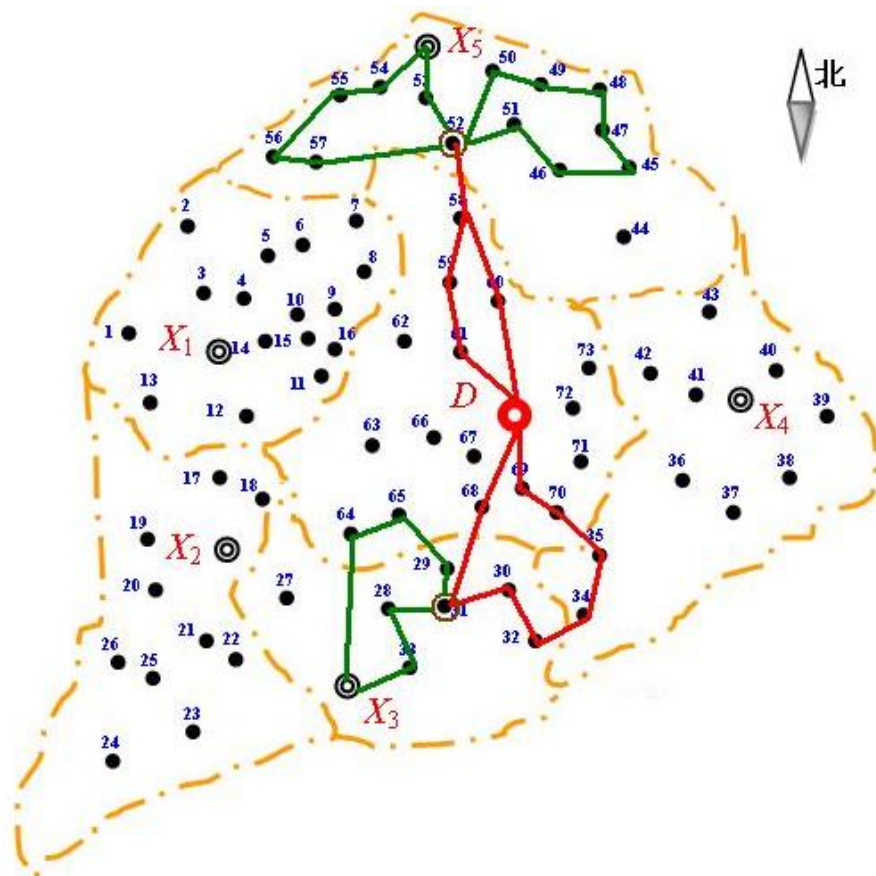


图 8：迁址后改变的邮路图

## 九、拓展性问题研究

**拓展 1：**可以考虑区级邮车两次的线路不同，这样有很大可能可以降低成本。因为对于各个支行来说每天收发一次就够了。如果两次路线相同，那么那些区级邮车经过的支行每天就会被收发两次，造成资源浪费。

**拓展 2：**对于问题 2 可以将多增加一辆车的成本加入到目标函数中直接进行整体优化，比如，区级邮车增加一辆的成本为 100 元，县级邮车增加一辆的成本为 50 元，那么目标函数就变为

$$\text{Minimize } L = 2L_0 + \sum_{i=1}^5 L_i + 10N_0 + 5\sum_{i=1}^5 N_i$$

$N_0$  表示区级邮车数， $N_i$  表示  $i$  县邮车数。

**拓展 3：**我们总是希望区级第一班次邮车以最快的速度到达县局，第二班次邮车能够尽量晚的离开县局，这样就保证了县局能有更多的时间运输，可以节省县局车辆。为达到这样的目的，我们可以假设第二班次区级邮车可以先走耗时长的路径到达县局，然后走耗时短的路径返回地市局。这就使得第二班次邮车可以逆着第一班次邮车的路线走。这是对于县局  $X_i$  它可用的时间为：

$$Time = 10 - 2 \cdot \min(T_1, T_2)$$

## 参考文献

- [1] 郭文忠、陈国龙, 求解 TSP 问题的模糊自适应粒子群算法, 计算机科学 Vol. 33, No. 6, 2006
- [2] 高海兵、周驰、高亮, 广义粒子优化模型, 计算机学报 Vol. 28, No. 12, Dec. 2005
- [3] 肖健梅、黄有方、李军军、王锡淮, 基于离散微粒群优化的物流配送车辆路径问题, 系统工程 Vol. 23, No. 4, Apr. 2005
- [4] 黄岚、王康平、周春光、庞巍, 董龙江, 彭利, 粒子群优化算法求解旅行商问题, 吉林大学学报, Vol. 41, No. 4, 477~480, 2003
- [5] 程录庆、胡涛, 车辆路由问题在邮路规划中的应用研究, 中国管理信息化, Vol. 9, No. 1, Jan. 2003
- [6] 费蓉、崔杜武, 中国邮递员问题的动态规划算法研究, 计算机研究与发展, 42 (2): 294~299, 2005
- [7] 陈龙、王国胤、刘心松、聂能, 一种启发式邮政运输调度优化方法, 电子学报, Vol. 28, No. 8, Aug. 2000
- [8] 钱颂迪等《运筹学》, 清华大学出版社, 2004 年 9 月

## 附录 1

### 粒子群优化算法 (PSO) 简介

PSO 是从模拟鸟群的捕食行为中得到启示的算法。设想这样一个场景：一群鸟在随机搜索食物。在这个区域里只有一块食物。所有的鸟都不知道食物在那里。但是他们知道当前的位置离食物还有多远。那么找到食物的最优策略是什么呢。最简单有效的就是搜寻目前离食物最近的鸟的周围区域。PSO 从这种模型中得到启示并用于解决优化问题。PSO 中，每个优化问题的解都是搜索空间中的一只鸟。我们称之为“粒子”。所有的例子都有一个由被优化的函数决定的适应值，每个粒子还有一个速度决定他们飞翔的方向和距离。然后粒子们就追随当前的最优粒子在解空间中搜索。PSO 初始化为一群随机粒子(随机解)。然后通过叠代找到最优解。在每一次叠代中，粒子通过跟踪两个“极值”来更新自己。第一个就是粒子本身所找到的最优解。这个解叫做个体极值 *pbest*。另一个极值是整个种群目前找到的最优解。这个极值是全局极值 *gbest*。在找到这两个最优值时，粒子根据如下的公式来更新自己的速度和新的位置：

$$v = w \cdot v + c_1 \cdot r \cdot (pbest - present) + c_2 \cdot r \cdot (gbest - present). \quad (1)$$

$$present = present + v \quad (2)$$

$v$  是粒子的速度， $present$  是当前粒子的位置。 $pbest$  与  $gbest$  如前定义  $r$  是介于



(0, 1) 之间的随机数.  $c_1, c_2$  是学习因子.  $w$  为惯性权重. 这里我们只给出 PSO 算法的一个基本框架在附录中我们将详细介绍对于 TSP 问题的优化技术。

PSO的基本算法步骤如下:

- (1) 初始化粒子群, 即随机设定各粒子的初始位置  $X$  和初始速度  $V$ ;
- (2) 计算每个粒子的适应度值;
- (3) 对每个粒子, 比较它的适应度值和它经历过的最好位置  $P_{id}$  的适应度值,

如果更好, 更新  $P_{id}$ ;

- (4) 对每个粒子, 比较它的适应度值和群体所经历最好位置  $P_{gd}$  的适应度值,

如果更好, 更新  $P_{gd}$ ;

- (5) 根据 (1.1) 式和 (1.2) 式调整粒子的速度和位置;
- (6) 如果达到结束条件(足够好的位置或最大迭代次数), 则结束; 否则转步骤 (2)。

### PSO算法的改进

本文中, 我们利用文献[1]中提到的交换子和交换序的概念, 对基本PSO算法进行改进。

**定义 1:** 设  $n$  个节点的 TSP 问题的解序列为  $S = (a_i), i = 1, \dots, n$ 。定义交换子  $SO(i_1, i_2)$  为交换解  $S$  中的点  $a_{i_1}$  和  $a_{i_2}$ , 则  $S' = S + SO(i_1, i_2)$  为解  $S$  经过交换算子  $SO(i_1, i_2)$  操作后的新解。

**定义 2:** 一个或多个交换子的有序队列就是交换序, 记作  $SS$ 。

$SS = (SO_1, SO_2, \dots, SO_n)$ , 其中  $SO_1, SO_2, \dots, SO_n$  是交换子, 它们之间的顺序是有意义的。交换序作用于一个解上, 意味着交换序中的所有交换子一次作用于该解上。

**定义 3:** 若干个交换序可以合并成一个新的交换序, 定义  $\bullet$  为两个交换序的合并算子。

**定义 4:** 不同的交换序作用于同一解上可能产生相同的新解, 所有有相同效果的交换序的集合成为交换序的等价集。

**定义 5:** 在交换序等价集中, 拥有最少交换子的交换序成为该等价集的基本交换序。

例如, 现有一解 (9 8 7 6 5), 交换子为  $SO(2, 5)$ , 则通过交换子操作后的新解为 (9 5 7 6 8); 若交换序为  $SS = (SO(1, 2), SO(4, 3), SO(1, 5))$ , 则通过交换序操作后的新解为 (5 9 6 7 8)。

现在重新构造基本 PSO 算法中的速度公式:

$$V_{id}' = \gamma V_{id} \bullet \alpha(P_{id} - X_{id}) \bullet \beta(P_{gd} - X_{id}) \quad (1.6)$$

其中  $\alpha, \beta, \gamma$  为随机数。  $\gamma V_{id}$  表示交换序  $V_{id}$  中所有交换子以概率  $\gamma$  保留；  
 $\alpha(P_{id} - X_{id})$  表示基本交换序  $(P_{id} - X_{id})$  中所有交换子以概率  $\alpha$  保留；  $\beta(P_{gd} - X_{id})$   
表示基本交换序  $(P_{gd} - X_{id})$  中所有交换子以概率  $\beta$  保留。

求解步骤描述如下：

(1) 初始化粒子群，即给群体中的每个粒子赋一个随机的初始解和一个随机的交换序；

(2) 如果满足结束条件，转步骤(5)；

(3) 根据粒子当前位置  $X_{id}$ ，计算其下一个位置  $X_{id}'$ ，即新解；

① 计算  $P_{id}$  和  $X_{id}$  之间的差  $A$ ，  $A = P_{id} - X_{id}$ ，其中  $A$  是一个基本交换序，  
表示  $A$  作用于  $X_{id}$  得到  $P_{id}$ ；

② 计算  $B = P_{gd} - X_{id}$ ，其中  $B$  也是一基本交换序；

③ 根据 (1.3) 式计算速度  $V_{id}'$ ，并将交换序  $V_{id}'$  转换为一个基本交换序；

④ 计算搜索到的新解：  $X_{id}' = X_{id} + V_{id}'$  (1.7)

⑤ 计算  $X_{id}'$  的适应度值，与个体经历的最好位置  $P_{id}$  的适应度值比较，若  
更好，则更新  $P_{id}$ ；

(4) 计算所有个体  $P_{id}$  的适应度值，与群体经历的最好位置  $P_{gd}$  的适应度值比  
较，若更好，则更新  $P_{gd}$ ，转步骤(2)。

(5) 显示求出的结果值。

在本文中，我们解决所有问题的思路是优先考虑区级邮路，因为其路程相对县级邮路较长，邮路遍及 5 各县局，是主要传输的路径；其次再考虑县级邮路，使其在数量和长度上都达到最小量。即先寻求区级邮路最优解，在此基础上求解县级邮路最优解。

## 5. 参数调节

在算法运行中，参数  $\alpha = \beta = \gamma = 0.2$  时，可以较快的找到可行解和最优解