

公共自行车服务点管理

[摘要]

本文将单自行车服务点调整方案的最优化问题扩展至区域内多自行车服务点管理的最优化问题。将选定区域通过空间的聚类分析进行分解，对子区域又进行时间的聚类分析进行遍历，根据专家系统确立最终方案以实现理想状态时间的最大化及人力成本最小化之间的最优权衡。

问题一的第一问，在数据处理阶段，将还、借需求量数据转化为毫无限制情况下的理想实时借还曲线（见 5.1.3）。对服务点调整时需要对该曲线进行纵向平移，使曲线限制在车架数 N 以内，计算任务繁重。因此采用相对平移的思想，将曲线的平移转化为坐标轴的平移。通过引入“基准线”（见 5.2.1），不断检测曲线是否位于基准线以上 N 的范围内；采用贪心算法的思想，当曲线处于临界状况时，通过对此后曲线极（最）大（小）值的分析确定调整量。在最后我们给出了这种方案下能够使得一天内调整次数最少的证明。

问题一的第二问，在对调整次数加以限制的情况下，将一天非理想状态时间最少转化为一天非理想时刻数最少（见 5.4.1）。非理想状态的时刻即为曲线不位于基准线以上 N 的范围之内的这些点。通过 0-1 规划模型确定一天各时刻状态，并得到目标函数。遍历所有可能的调整时刻及调整量，计算所有情况下的目标函数值，寻找其最小值所对应的方案即可得获得最优解。

随后我们给出了问题一各种模型的评价。

问题二考虑区域内多个服务点之间的人员调度及调整方案。首先根据服务点所处位置通过 HCA 聚类分析（以各点坐标为变量）确定簇数及其子集（见 6.1.6）。再针对某一个子区域内部，将各服务点间的实体线路选择抽象为图论的最短路径模型，再结合巡逻的实际情况，利用 Dijkstra 算法和遗传算法解决 TSP 模型并确定子区域内的最短巡逻路径（见 6.1.7）。子区域内所有点的理想调整方案可根据问题一的第一问模型确定。在不同簇数的条件下，对调整时刻进行 HCA 聚类分析，以确定一天内的巡逻次数。再在不同簇数下，计算不理想状态之和。最后通过专家系统所给出的综合效益函数确定最终方案。算法的核心是对这些子区域的最优化来实现整体的最优化，并且我们给出了求解的一般步骤。

最后本文还对现实中的借还车顺序进行了考虑，具有非常好的现实意义。

关键词：贪心算法 数学归纳法 遍历法 图论 0-1 规划模型
HCA 聚类分析 TSP 模型 Dijkstra 算法 遗传算法

1 问题重述

杭州自 2008 年在全国首推公共自行车服务系统以来，城市行路停车难问题得到缓解，其中“最后一公里”交通问题得到了有效的解决。截至 2012 年底，杭州共有公共自行车 69750 辆，公共自行车服务点 2962 个，日均租车量达到 25.75 万人次，深受杭州市民和外地游客的欢迎。

公共自行车的借车、还车环节均需在服务点完成。每个服务点设有若干车架，每一个车架只能锁住一辆自行车。当车架上无车时，刷卡可借出自行车；当车架上无车时，可刷卡将车所在车架上，实现还车。服务点的借还车会出现三种情况：

一、服务点所有车架都有车（简称满架）时，只能借车而不能还车；

二、服务点当服务点所有的车架都无车（简称空架）时，不能进行借车而只能进行还车。

三、服务点处于理想状态，既不满架也不空架，出行的人可根据自身需求就近随时借车还车。当服务点处于非理想状态时，工作人员可通过上架或下架操作使服务点重新恢复理想状态。

公共自行车是一项公益性事业，在向市民提供优质、低价服务的同时必须考虑运营成本。目前，杭州除 24 小时值守点的其他自行车服务点是通过工作人员骑自行车进行巡逻管理，以使得车架在尽可能多的时段下处于理想状态。

为了获得理想的公共自行车服务点调整优化方案，要求建立相关模型以解决以下几大问题：

问题一：

1. 假设某服务点车加数量已知，通过实际调查统计掌握了一段时期内从早上 6 时到晚上 9 时间隔 10 分钟的借车、还车需求数据。试给出巡逻人员到该服务点的时刻以及上架或者下架自行车的数量，以达到一天内需对该点进行管理的次数最少的目标。

2. 假设一天内到该服务点的次数不超过两次，如何才能使服务点处于理想状态的时间尽可能长。

问题二：

1. 选定杭州的某一个区域，确定巡逻人员的数量以及调度方案，使得该区域内的公共自行车服务点尽可能保持理想状态。

2 问题分析

本问题主要在三种不同情况下，研究工作人员对自行车服务点的调整方案问题。联系实际，调整方案主要考虑的因素包括调整次数、调整时间、调整数量等因素。为满足为自行车服务点达到最长时间的理想状态，在不同的条件下确定出调整的最佳方案。

针对问题一的第一问，仅考虑调整次数且自行车服务点的借还车需求数据已知的情况下，根据需求计算出无限制情况下的任意时刻的服务点状态，并通过这一中间变量确定需要调整的时刻以及该时刻所需要调整的自行车数量。

进一步考虑实际情况，借还车的顺序对服务点的调整方案也有着一定的影响，因此研究这种情况下的调整方案具有重要的实际意义。

而问题一的第二问则对每天的调整的次数加以限制，在考虑服务点理想时间尽可能大的情况下，确定需要调整的时刻以及该时刻所需要调整的自行车数量。

在考虑第二问的情况下，为使得区域内的服务点理想时间尽可能长、人力成本尽可能小，调整方案需要对区域内各个服务点的地理位置分布以及基于问题一所得到的各项参数指标进行综合考虑，最后再引入专家系统的前提假设下，给出人员调度的最佳方案。

3 模型假设

- (1) 假设 10 分钟之内的借、还车需求量不会超过服务点的自行车架数；
- (2) 假设 10 分钟之内的借、还车行为不分先后，仅考虑起始终止时刻；
- (3) 假设工作人员对服务点调整一次时间为 3 分钟；
- (4) 假设工作人员对区域内各服务点巡逻时的速度为 4m/s。
- (5) 假设工作人员的工作效率相同。
- (6) 假设工作人员的工作不受极端情况影响。
- (7) 假设工作人员严格按照时间安排进行工作。

4 符号系统

N —— 自行车服务点的车架数；
 a_i —— 第 i 个 10 分钟内的借车需求量；
 b_i —— 第 i 个 10 分钟内的还车需求量；
 Δ_i —— 第 i 个 10 分钟内还、借车需求之差；
 c_i —— 第 i 个时刻还、借车需求之差的累加值；
 d_i —— 第 i 个时刻对服务点增加（负则减少）的自行车数量；
 p —— 当天内对服务点调整的总次数；
 λ_j —— 第 j 次对服务点增加（负则减少）的自行车数量的相反数；
 t_j —— 第 j 次对服务点调整的时刻；
 L_j —— 第 j 次对坐标系横轴相对于 0 上移（负则相对下移）的量；
 q_i —— 第 i 个时刻自行车服务点是否处于理想状态；
 Q —— 当天内处于非理想状态时刻数的总和；
 P_k —— 第 k 个自行车服务点；
 s —— 区域内自行车服务点总个数；
 n —— 区域内调度安排的总人数；
 T_m —— 第 m 个人当天内巡逻的次数；
 R_m —— 第 m 个人当天内巡逻的总距离；
 r_m —— 第 m 个人每次巡逻一圈的距离；
 H —— 服务点间最短距离矩阵；

5 公共自行车服务点调整方案优化模型（问题一）

本节主要研究某一公共自行车服务点的调整方案优化的数学模型与算法。

5.1 模型 I 的分析与建立

5.1.1 条件分析

将实地调查统计掌握的一段时期内的需求数据考虑至某一天内的从早上 6:00 至晚上 9:00 每隔 10 分钟的借车、还车需求数据。则一共可以得到该服务点一天内 90 个时间段的需求数据的数组 a_i 与 b_i ，其中 $1 \leq i \leq 90$ 。该自行车服务点的车架数为 N 。

5.1.2 目标分析

模型的目标为管理次数 p 达到最小值，表示为

$$\text{Min } p$$

5.1.3 约束分析

(1) 自行车服务点车架数约束

基于对条件的分析，需要通过每个时段的借还车需求数据计算出每个时刻的实际自

行车数。为此，我们引入新的数组—— c_i 、 d_i 。

c_i 定义为起始时刻 6:00 至第 i 个时刻还、借车需求之差的累加值。

d_i 定义为第 i 个时刻对服务点增加（负则减少）的自行车数量。

根据上述定义得到

c_i 的递推公式

$$c_{i+1} = c_i + b_{i+1} - a_{i+1} (c_0 = 0, 1 \leq i \leq 90)$$

c_i 的通项公式

$$c_i = c_0 + \sum_{k=1}^i b_k - \sum_{k=1}^i a_k (c_0 = 0, 1 \leq i \leq 90)$$

[注]这里需要对 i 进行说明，以免产生歧义。如 $i=1$ 时， c_1 表示 6:10 这第 1 个时刻的需求总量累加值，而 a_1 、 b_1 表示 6:00 至 6:10 这第 1 个 10 分钟借、还车的需求量； i 为其他值时依此类推。

最后得出第 i 个时刻的实际自行车数为

$$c_i + \sum_{k=1}^i d_k + d_0 (1 \leq i \leq 90)$$

[注] d_0 的意义与 $d_i (i \geq 1)$ 的意义略有不同。事实上， d_0 就表示起始时刻 6:00 时的自行车数量。

则自行车服务点车架数的约束条件可表示为

$$0 \leq c_i + \sum_{k=1}^i d_k + d_0 \leq N (1 \leq i \leq 90)$$

即

$$0 \leq c_0 + d_0 + \sum_{k=1}^i d_k + \sum_{k=1}^i b_k - \sum_{k=1}^i a_k \leq N (c_0 = 0, 1 \leq i \leq 90)$$

(2) 自行车调整量约束

考虑到实际情况，每次调整的自行车数量必然限制在区间 $[-N, N]$ 之内，则自行车调整量的约束条件可表示为

$$-N \leq d_i \leq N (1 \leq i \leq 90)$$

5.1.4 模型 I 的建立

根据以上分析，建立的模型表达式如下：

$$\begin{aligned} & \text{Min } p \quad (p \text{ 为 } d_i \text{ 中不为 } 0 \text{ 的量的总个数}) \\ & S.T. \begin{cases} 1 \leq i \leq 90 \\ 0 \leq c_0 + d_0 + \sum_{k=1}^i d_k + \sum_{k=1}^i b_k - \sum_{k=1}^i a_k \leq N (c_0 = 0) \\ -N \leq d_i \leq N \end{cases} \end{aligned}$$

5.1.5 模型 I 的评价

模型 I 的优点在于模型非常直观并且易于理解，这是由于它用数学语言直接描述了问题一第一问所给出的所有条件。

模型 I 的缺点在于：

(1) 变量 p 的值在模型中无法直接用数学语言体现出来，如需表达还需引入以“调整次数”为自变量的新的数组；

(2) 模型中每次的调整量都必须以上一次调整为基础，在这一点上就显得不是非

常直观；

(3) 由于涉及到每一个时刻是否要调整及调整量，计算机实现起来显得非常复杂，程序编写繁琐。

根据上述的缺点，我们对模型 I 加以改进，并建立模型 II。

5.2 模型 I 的转化及模型 II 的分析与建立

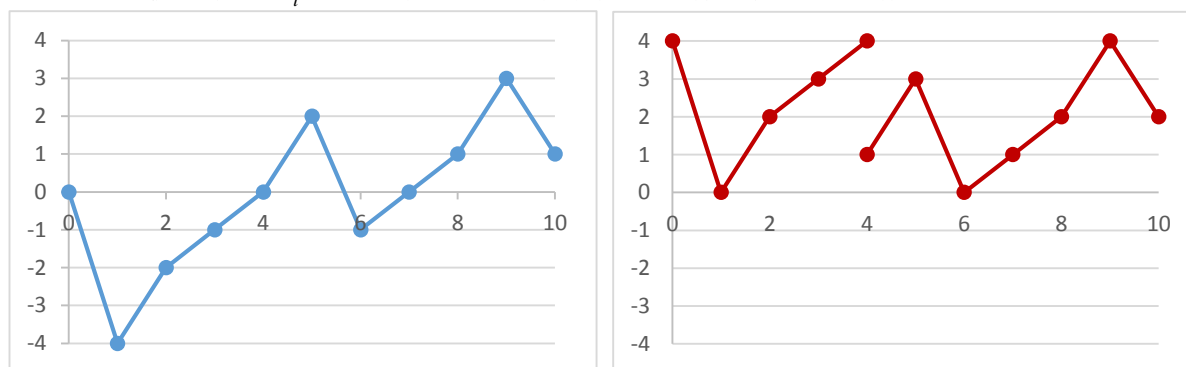
5.2.1 模型 I 的转化的例子引入

为使得我们对模型转化的过程更加直观可见，我们先引入一个例子。

假设 $1 \leq i \leq 90$ ， $N = 4$ 。通过 a_i 、 b_i 已经得出了 c_i 的值（我们假设的）如下表所示。

i	0	1	2	3	4	5	6	7	8	9	10
c_i	0	-4	-2	-1	0	2	-1	0	1	3	1

以 i 为横坐标， c_i 为纵坐标（下图中未标出）作图如下左图所示：



则在该范围内对曲线进行两次调整。

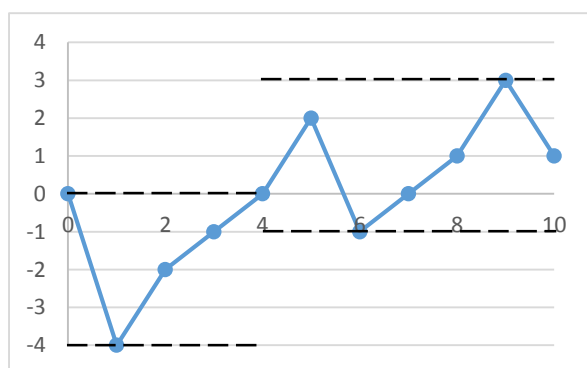
第一次： $d_0 = 4$ ；第二次： $d_4 = -3$ ；其余的 d_i 值均为 0。

调整后实际的曲线如上右图所示，即对曲线进行了分段平移。

由于对曲线的平移相对复杂，则根据相对性，我们考虑将曲线的分段平移转化为坐标系中横轴的纵向平移。

具体描述为：

第一次的平移相当于以 $c_i = -4$ 作为“基准线”，第二次的平移相当于以 $c_i = -1$ 作为“基准线”。



这么做的优点在于，对横轴的相对纵向平移使得整个问题显得更加直观，不仅易于理解，从编程的角度上看也更加容易实现。

则原来所建立的模型可以转化为按这个思路建立的新模型。

5.2.2 模型 II 的分析

由于此处的分析是以建立模型 I 时的分析为基础的，故此点不再根据条件分析、目标分析、约束分析分点叙述。

为建立新模型，我们引入三个新的数组。

λ_j 定义为第 j 次对服务点增加（负则减少）的自行车数量的相反数，以取代原来变量中的 d_0 ，并将“第 i 个时刻”转化为“第 j 次”；

t_j 定义为第 j 次对服务点调整的时刻，且有 $0 \leq t_j \leq 90$ ；

L_j 定义为第 j 次对坐标系横轴相对于 0 而上移（负则相对下移）的量，也就是在上述例子中的“基准线”。

在上述变量中， $j=0,1,2,\dots,p$ 。

[注]当 $j=0$ 时仅代表 6:00 这一初始时刻的状态（如最初的“基准线”、最初所要置入的自行车数等）。

事实上，新模型中的 L_j 是由 λ_j 累加而得，即每一次增加（或减少）的自行车量都是相对于上一次所确定的基准而言的，则可得递推公式

$$L_j = L_{j-1} + \lambda_j (1 \leq j \leq p) \text{ 且 } L_0 = \lambda_0$$

通项公式

$$L_j = \sum_{k=0}^j \lambda_k (0 \leq j \leq p)$$

新模型中 c_i 的通项公式依旧为

$$c_i = c_0 + \sum_{k=1}^i b_k - \sum_{k=1}^i a_k (c_0 = 0, 1 \leq i \leq 90)$$

新模型的目标依旧为

$$\text{Min } p$$

但新模型的约束条件发生改变。

(1) 自行车服务点车架数约束

由于在两次调整的时刻之内，必须保证曲线 c_i 的位于区间 $[L_j, L_j + N]$ 之内，因而可以得到

$$\forall t_j \leq i \leq t_{j+1}$$

有

$$L_j \leq c_i \leq L_j + N (0 \leq i \leq 90)$$

(2) 自行车调整量约束

由于 λ_j 取代了原来变量中的 d_i ，因此有

$$-N \leq \lambda_j \leq N (0 \leq j \leq p)$$

5.2.3 模型 II 的建立

根据“5.2.2 模型 II 的分析”，建立的模型表达式如下：

$$\begin{aligned} & \text{Min } p \\ & S.T. \begin{cases} \forall t_j \leq i \leq t_{j+1}, L_j \leq c_i \leq L_j + N \\ -N \leq \lambda_j \leq N \\ 0 \leq i \leq 90 \\ 0 \leq j \leq p \end{cases} \end{aligned}$$

5.2.4 模型 II 的评价

模型 II 的优点：

(1) 在理解相对平移情况下，模型 II 的“基准线”平移取代了曲线分段平移；不对原曲线进行改变，使模型显得更加形象而生动。

(2) λ_j 取代了原来变量中的 d_i 使得变量 p “数学地”融入模型中;

(3) 引入了数组 L_j 表达调整量的累加值, 大大方便了算法的实现;

(4) 该模型能够通过贪心算法非常直接地给出需要调整的时刻及调整量。

模型 II 的缺点在于假设了 10 分钟之内的借、还车行为不分先后, 仅考虑起始终止时刻。事实上, 对实际情况进行考虑, 对于 10 分钟之内“先借后还”与“先还后借”的极端情况, 我们还需要对 L_j 进行进一步的约束使得该服务点在上述的极端情况下仍然能保持理想状态。当然这也是模型 I 和模型 II 共同的缺点。

5.3 模型 II 的求解 (程序和模拟检验结果见附件 9.1、9.2)

5.3.1 模型 II 求解的方法

求解本模型的方法基于贪心算法。

第一步, 输入数组 a_i 与 b_i ;

第二步, 根据 a_i 和 b_i 计算出 c_i ;

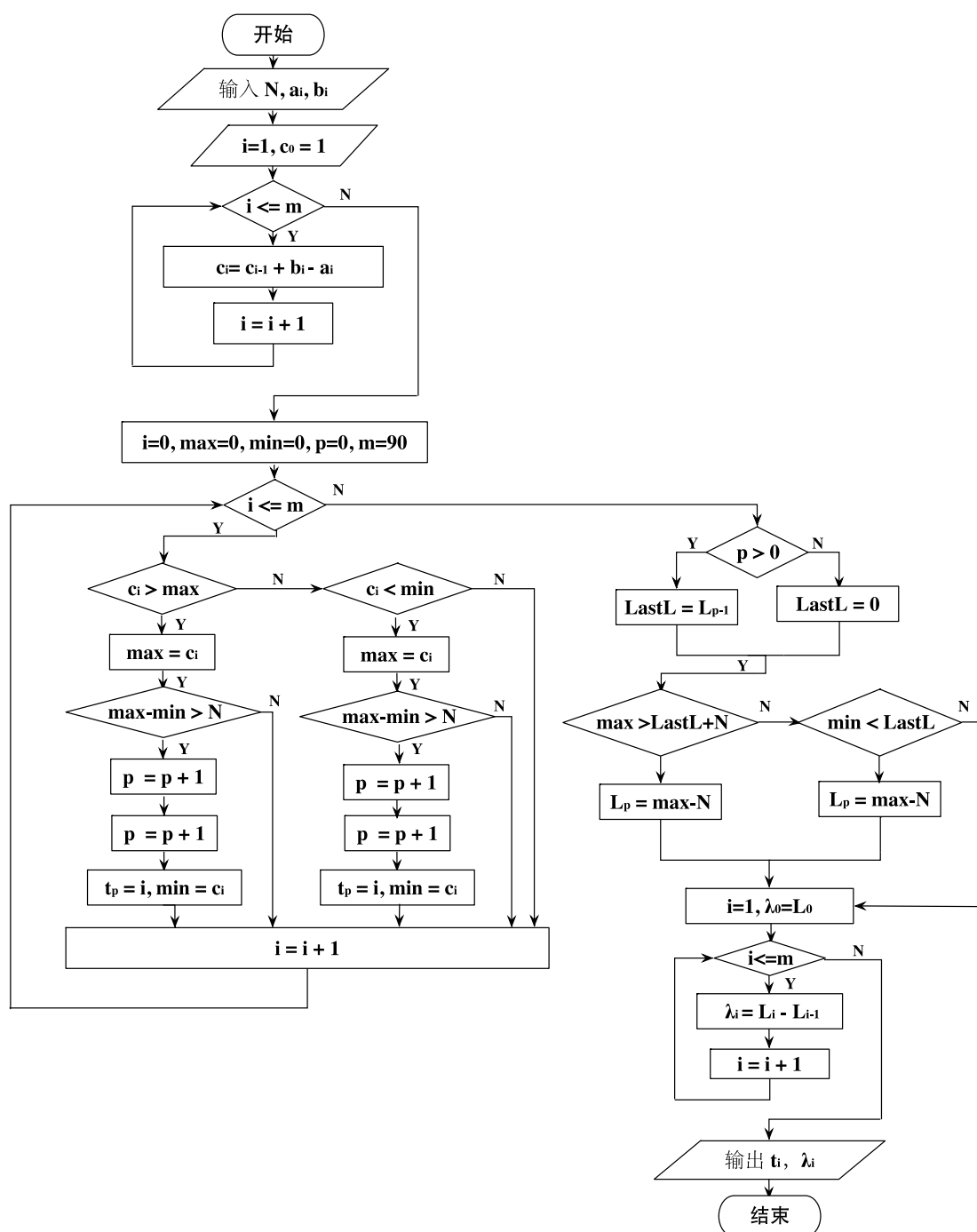
第三步, 选取第 0 个时间 (即 6:00) 为起始点;

第四步, 从起始点开始, 并不断向后查看每一个 c_i , 记录下 c_i 的最大值 $Max(c_i)$ 和最小值 $Min(c_i)$ 。当我们找到 $Max(c_i)$ 与 $Min(c_i)$ 之差大于自行车站点容量 N 时, 转到下一步。当查看到最后一个 c_i 值的时候, 转到第六步并退出循环;

第五步, 增加总调整的次数 p , 记录下此次需要的调整的时刻 t_j , 再根据 $Max(c_i)$ 和 $Min(c_i)$ 的值, 确定出上一次应该调整到的位置 $L_j - 1$, 并将当前点记为起始点, 重复第四步;

第六步, 计算最后一次应该调整到的位置 L_p , 并根据 L_j 数组计算出每次调整的量 λ_j , 输出结果。

5.3.2 模型 II 求解的流程图



5.3.3 贪心算法下目标达成的证明

假设我们已经通过贪心算法得出了 λ_j 、 t_j 及 p 的值，现用数学归纳法证明通过这种算法得出的 p 确实达成了模型 II 所要求达成的目标 $\text{Min } p$ 。

假设现在有另一方案的 λ'_j 、 t'_j 及 p' 的值。

首先我们来证明 $\exists j, t'_j \leq t_j \Rightarrow t'_{j+1} \leq t_{j+1}$ 成立。

假设 $\exists j, t'_j \leq t_j$ 成立，则在 t_j 之后的某一段时间内有 $L_j \leq c_i \leq L_j + N, t_j \leq i \leq t_{j+1}$ 。

(1) 若 $c_{t'_j} \in [L_j, L_j + N]$ ，则 $(t'_{j+1})_{\max} = t_{j+1}$ 。

这是由于如若 $t'_{j+1} > t_{j+1}$ ，则在 $t_{j+1} \leq i \leq t'_{j+1}$ 的时间段内，必然会出现非理想状态，这不符合模型 II 规定的约束条件。因此有 $t'_{j+1} \leq t_{j+1}$ 成立。

(2) 若 $c_{t'_j} \notin [L_j, L_j + N]$ ，则 $t'_{j+1} < t_{j+1}$ 。

这是由于如若 $t'_{j+1} \geq t_{j+1}$ ，则在 $t_{j+1} \leq i \leq t'_{j+1}$ 的时间段内，必然会出现非理想状态，这不符合模型 II 规定的约束条件，因此有 $t'_{j+1} < t_{j+1}$ 成立。且在这种情况下，并有可能出现“原方案调整一次，带撇方案调整两次”的极端情况。

综上所述，有

$$\exists j, t'_j \leq t_j \Rightarrow t'_{j+1} \leq t_{j+1}$$

成立。而基于模型 II，又有

$$t_0 = t'_0$$

成立，因此有

$$\forall 0 \leq j \leq \min(p, p'), t'_j \leq t_j$$

而 t'_j 总是发生在 t_j 之前，因此

$$\min(p, p') = p$$

那么就有

$$p \leq p'$$

证毕。

5.4 模型 III 的分析与建立

5.4.1 新的目标及约束条件分析

针对问题一的第二问，我们需要在模型 II 的基础上再增加（改变）若干约束条件，并且改变达成的目标。

模型 II 中的目标 $\text{Min } p$ 在模型 III 中改变为一个新的约束条件。因为第二问要求一天内调整的次数不超过 2 次，因此我们得到

$$p \in \{0, 1, 2\}$$

而 $L_j \leq c_i \leq L_j + N (0 \leq i \leq 90)$ 的约束条件将在本问中删去。

同时，为了表达一天内处于理想状态的时长，我们引入一个新的数组 q_i 与一个新的变量 Q ，构成一个 0-1 整数规划。

q_i 定义为第 i 个时刻自行车服务点是否处于理想状态。 $q_i = 0$ 表示第 i 个时刻自行车服务点处于理想状态，而 $q_i = 1$ 表示第 i 个时刻自行车服务点处于满架或空架的非理想状态。即

$$\forall t_j \leq i \leq t_{j+1}$$

有

$$q_i = \begin{cases} 0, & L_j \leq c_i \leq L_j + N \\ 1, & \text{其他} \end{cases}, 0 \leq i \leq 90$$

Q 定义为当天内处于非理想状态时刻数的总和。则根据定义就有

$$Q = \sum_{i=0}^{90} q_i$$

问题一的第二问要求当天内处于理想状态的时间尽可能长，意即处于非理想状态的时刻数要尽可能少。因此模型 III 的目标即使得 Q 达到最小，记为

$$\text{Min } Q$$

5.4.2 模型III的建立

根据“5.4.1 新的目标及约束条件”，建立的模型表达式如下：

$$\begin{aligned} & \text{Min } Q \\ & S.T. \begin{cases} -N \leq \lambda_j \leq N \\ p \in \{0,1,2\} \\ 0 \leq i \leq 90 \end{cases} \end{aligned}$$

5.4.3 模型III的评价及改进方向

模型III的优点：

(1) 引入了 q_i 这一统计非理想时刻个数的变量，而不用考虑非理想时刻的起始时刻和结束时刻；

(2) 引入的 q_i 使得模型约束条件的表达是显得非常简洁。

模型III的缺点在于采用了穷举法进行分析，因此程序的复杂度较高，其复杂度与 p 的上限值呈现指数增长。

5.5 模型III的求解方法（程序和模拟检验结果见附件 9.3、9.4）

模型III由于限制了调整次数，并且可以出现不理想状态，因此我们选择历遍法对其进行求解。即历遍所有的

$$\begin{cases} -N \leq \lambda_j \leq N \\ p \in \{0,1,2\} \\ 0 \leq i \leq 90 \end{cases}$$

对于每一种情况都能计算出其 Q ，只需在其中找出最小 Q 对应的那个方案即可。

6 多公共自行车服务点人员调度方案优化模型（问题二）

本节主要研究多个公共自行车服务点人员调度方案优化的数学模型与算法。

6.1 模型IV的分析与建立

6.1.1 条件分析

(1) 所给区域内所有自行车服务点的网络结构图；

(2) 所给区域内第 k 个自行车服务点的车架数 N_k ，其中 $1 \leq k \leq s$ ；

(3) 所给区域内第 k 个自行车服务点一天内从 6:00 至 21:00 每隔 10 分钟的还、借车需求量之差 Δ_{ik} 数组，其中 $1 \leq i \leq 90$ ， $1 \leq k \leq s$ 。

6.1.2 目标分析

事实上，问题二所要达成的两个目标是矛盾的。为使不理想时间较少就必须加大人力成本；反之减少了人力成本也很可能导致不理想时间的增多。因此问题二的目标分析成为了一个多目标规划问题。

目标一：使所有服务点的不理想时间最小。即使得总和 $\sum_{k=1}^s Q_k$ 最小，记为

$$\text{Min } \sum_{k=1}^s Q_k$$

目标二：使人力成本（以工作时间作为衡量）最小。由于已经假设了工作人员的巡逻速度固定为 4m/s，也假设了工作人员对服务点调整一次时间为 3 分钟；而 $\sum_{k=1}^s P_k$ 表示

整个过程中的总调整次数，又有 $\sum_{m=1}^n R_m$ 表示所有工作人员巡逻的总路程，那么总工作时

间（分钟）最小可记为

$$\text{Min } (3\sum_{k=1}^s p_k + \frac{1}{240}\sum_{m=1}^n R_m)$$

但这两个目标不可能同时达到。

6.1.3 约束分析

在对约束条件进行分析之前，我们需要对各变量之间的关系进行说明。

但在该问题中，似乎只有一个约束条件——第 k 个自行车服务点的第 j 次调整数量应满足 $-N_k \leq \lambda_{jk} \leq N_k$ 。但事实上，对工作人员巡逻服务点及路径的安排，都会对 R_m 以及 p_k 等诸多变量的取值产生非常大的影响的。

而这些变量之间的函数关系的构建又需要引入更多的中间变量，因此我们在这里也仅仅用数学语言来描述问题二而不对以下模型的求解进行更深入的探讨。由此一来我们就非常有必要通过实际情况对模型进行合理的化简。

6.1.4 模型IV的建立

这是根据上述的分析所建立的复杂模型表达式：

$$\text{多目标规划} \begin{cases} \text{Min } \sum_{k=1}^s Q_k \\ \text{Min } (3\sum_{k=1}^s p_k + \frac{1}{240}\sum_{m=1}^n R_m) \\ \text{S.T.} \begin{cases} -N_k \leq \lambda_{jk} \leq N_k \\ 0 \leq i \leq 90 \\ 1 \leq j \leq p_k \\ 1 \leq k \leq s \\ 1 \leq m \leq n \end{cases} \end{cases}$$

6.1.5 模型IV按实际情况简化

事实上，从上述分析的过程中可得该模型的实现难度较大，涉及到的变量个数较多，并且未考虑许多现实的情况。因此，我们针对模型实际情况进行了考察，在原有模型的基础上提出了如下假设：

- (1) 假设工作人员有且仅有属于自己的工作区域；
- (2) 假设工作人员的工作区域不重合；
- (3) 假设工作人员仅在属于自己的工作区域内进行巡逻；
- (4) 假设工作人员在工作区域内按固定的路径进行巡逻；
- (5) 假设工作人员巡逻的路径满足：①一次巡逻能够不重复地走完工作区域内的所有服务点并回到起点；②条件①成立下的最短路径。
- (6) 假设工作人员所辖区域内的自行车服务点数目相近。

根据以上假设，我们就将整个区域的最优化目标求解转化为每个子区域的最优化目标求解。

按上述假设，一个大的服务点网络结构被分解为 n 个子区域并分配给 n 个工作人员。根据每个子区域的网络结构，我们可以计算出该网络结构中假设（5）中所确定的最短路径 r_m 。如果再用 T_m 表示第 m 个人一天内巡逻次数，那么目标二的表达式就可以简化为三个子目标，分别记为

$$\text{Min } n; \text{ Min } T_m; \text{ Min } p_k$$

事实上， p_k 不会超过工作人员对子区域巡逻的次数 T_m ，因此我们认为在考虑 $\text{Min } T_m$

的目标同时就已经将 $\text{Min } p_k$ 包含进去了。则三个子目标又可以化简为两个字目标，分别记为

$$\text{Min } n; \text{Min } T_m$$

如此一来，在 n 不变的情况下，子区域内的人力成本就仅由工作人员在该区域内的巡逻次数所决定。当所有子区域在满足自身最优的前提下，整个区域的最优也就实现了。

6.1.6 区域的划分

假设区域的划分以 \bar{M} 作为每个工作人员应管辖的自行车服务点数的大致均值，则区域所需的子区域个数（簇）应满足

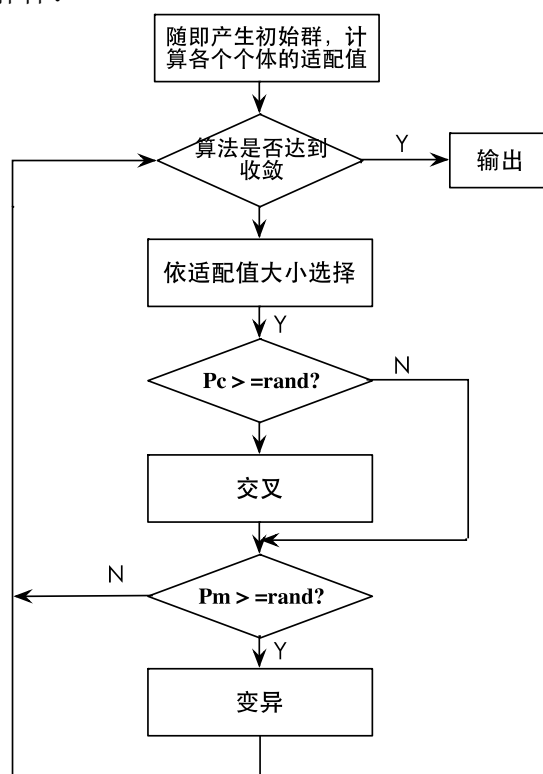
$$n = \frac{S}{\bar{M}}$$

通过 HCA 聚类分析可以将区域划分为 n 个子区域，每个子区域由一个工作人员负责管理 \bar{M} 个左右的自行车服务点。系统聚类分析可以通过 **IBM SPSS Statistics 20** 软件实现。

6.1.7 最短巡逻方案确定（TSP 模型代码详见附件）

在测算距离之前，需要将大区域的网络结构根据划分情况拆解为 n 个子区域的网络结构。再用遗传算法求解 TSP 模型，确定每个子区域的最短巡逻方案 r_m 。

遗传算法流程图解释：



6.1.8 子区域各服务点理想调整时间求解

根据模型 II 计算出子区域各个服务点的理想调整时刻。

假设安排一天内工作人员的巡逻次数为 T_m ，则以 T_m 为 K 值对所有需要调整的时刻进行 HCA 聚类分析，得到 T_m 个簇。

[注]此节中的 HCA 聚类分析的维度为一，区别于“6.1.5 区域的划分”。

因此，在简化后的模型中， T_m 显得尤为重要。这是因为 T_m 不仅决定着一天内不理

想状态的时间，也决定着人力成本。

6.1.9 专家系统引进

由于在实际情况下，评估不理想状态带来的社会效应无统一的标准，通常需要通过专家来评估考察。因此，我们将引进一个专家系统来将社会效应和人力成本化为同样的量纲，并进行比较分析得出最优方案。

假设专家系统评估所获得的综合效益函数为

$$z = f(\sum Q_k, T_m)$$

其中 $\sum Q_k$ 求和符号的范围为所辖区域。

另外，由于最短巡逻方案 r_m 的唯一性， $\sum Q_k$ 的值仅与 T_m 有关，记

$$\sum Q_k = g(T_m)$$

则

$$z = f(g(T_m), T_m) = Z(T_m)$$

为求得 z 最大值时 T_m 的值，只需对 T_m 求导即可。即解方程

$$\frac{dz}{dT_m} = 0$$

即可得 T_m 。

6.1.10 模型IV简化模型总结

第一步，我们对选定区域根据距离进行聚类分析，确定工作人员人数和各个子区域；

第二步，针对某一个子区域，根据图论抽象出结构图后，再根据遗传算法计算出巡逻一圈（遍历各点）的最短距离；

第三步，根据子区域内各个服务点的一天内的需求曲线，利用模型 II 计算出各个服务点的理想调整时间和调整数量；

第四步，假设一天内巡逻次数 T_m 的值，对子区域内所有服务点的理想调整时间进行 HCA 聚类分析，得到一天内巡逻的各个时间段；

第五步，对于某一次巡逻，根据到达各个服务点的时间对该服务点的调整数量历遍，求出不理想时间最小值 $g(T_m)$ ；

第六步，引入专家系统，对不理想时间和人力成本之间的关系进行评估，得到综合效益函数 $z = f(\sum Q_k, T_m) = f(g(T_m), T_m) = Z(T_m)$ ，并求其极大值点。

至此，可以确定安排人员数 n 、巡逻次数 T_m 、巡逻的路径及每次巡逻的时刻，进而确定了人员调度方案。

6.2 模型IV简化模型的代入

6.2.1 区域的选定（地图详见附件 9.5）

对于区域的选定，我们出于方便和实际的角度，以及杭州市轨道交通 1 号线的开通，考虑了以下因素：

- (1) 选定区域的范围尽量规则；
- (2) 选定区域所包含的自行车服务点数尽量满足在 50 个左右；
- (3) 选定区域道路的走向尽量满足东西和南北走向；
- (4) 选定区域内尽量包含轨道交通站点。

综合考虑了以上因素，我们最终选定的区域为：

以环城北路为北边界；以环城西路和湖滨路（六公园处有拐点）为西边界；以解放路为南边界；以中和路（中和高架）为东边界。

区域的形状较为规则，其中的道路基本满足东西和南北走向。区域内包含了 52 个自行车服务点。区域内还包括了轨道交通 1 号线的武林广场站、凤起路站和龙翔桥站。

从百度地图上截得该区域的地图详见附件，我们还在附件的图中标记了区域内所包含的所有自行车服务点，并在图中以编号的形式给出。

6.2.2 区域的划分（坐标系建立、划分日志及详细结果详见附件 9.6、9.7）

下面我们给出区域划分的步骤。

首先在地图中选取原点并建立直角坐标系，并对大区域内的所有自行车服务点进行坐标的测算（测算结果详见附件 9.6）。得到这 52 个自行车服务点的坐标后，我们需要决定所分的子区域（簇）的个数。

得到 52 个自行车服务点的坐标后，我们就可以通过 **IBM SPSS Statistics 20** 软件使用 HCA 聚类分析对这 52 个自行车服务点进行聚类分析。而分析中的变量即为服务点的两个坐标值。

由于需要尽量减少人力成本，因此选择簇的个数应尽可能少。因此我们选择 2 个簇的分类结果。

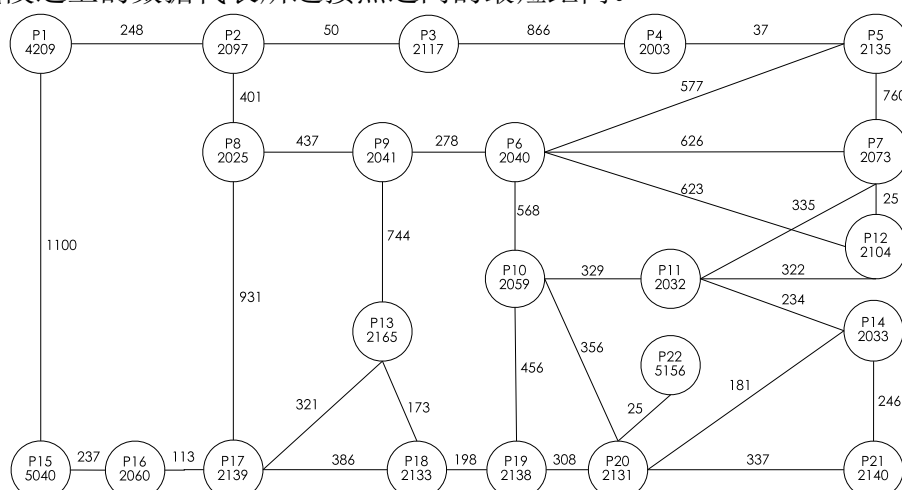
根据 2 个簇的分类结果，得到第一个簇包含有 22 个服务点，第二个簇包含有 30 个服务点。通过观察每个服务点的分配结果，我们发现这两簇以凤起路为分界线（每个点的具体划分结果详见附件 9.7）。

由于接下来对于 2 个簇的算法过程是一样的，为了简化我们的计算步骤及说明，我们仅对其中一个簇进行分析。我们选定的子区域为大区域中凤起路以北的这块区域，也就是聚类分析法后的簇 1。

6.2.3 距离测算及线路抽象处理（子区域详情详见附件 9.8）

在我们确定的子区域（簇 1）内共有 22 个自行车服务点，用符号 $P_k (1 \leq k \leq 22)$ 来表示。为了使得距离的测算更加方便，我们假设了工作人员在巡逻时只能走环城北路、环城西路、凤起路、中和北路这四条边界，还有武林路、体育场路、中山北路、延安路这四条交通量比较大的路以及狮虎桥路、百井坊巷、屏风街（这三条路实则为一路线）。

通过杭州市公交出行实时信息服务系统 (<http://www.hzbus.cn/>)，我们根据图论将各服务点抽象为一个网络如下图所示，其中 P_k 上的四位数字代表其对应的自行车服务点编号，线段之上的数据代表所连接点之间的最短距离。



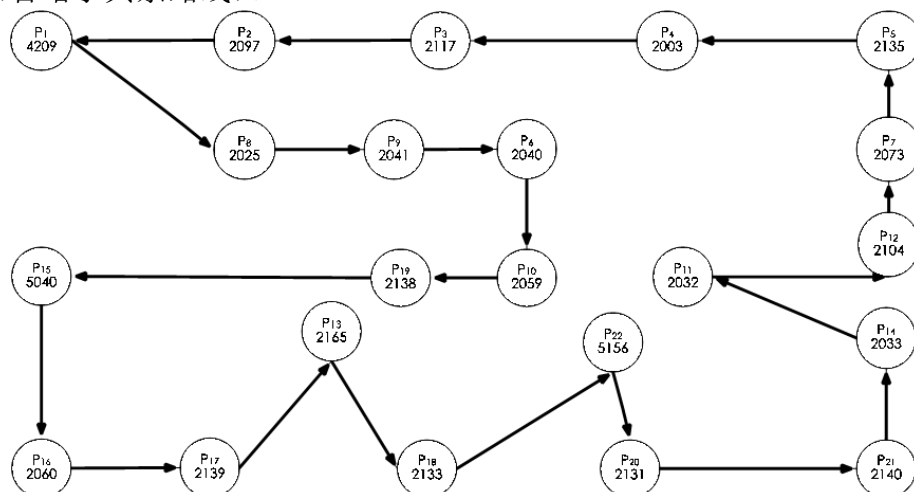
但图中无法表达出各个服务点相互之间的最短距离。由此，我们引入服务点间最短距离矩阵并记为 H ， H 中的某一元素 h_{ij} 或 h_{ji} 代表第 i 个服务点与第 j 个服务点之间的最短距离。显然 H 是一个 22 阶的对称矩阵，且主对角元均为 0。

根据上述抽象图，我们可以利用 **Dijkstra 算法**（详见附件 9.9）计算 H 矩阵。 H 矩阵的计算结果如下矩阵所示：

$$H = \begin{bmatrix} 0 & 248 & 298 & 1164 & 1201 & 1364 & 1961 & 649 & 1086 & 1932 & 2261 & 1986 & 1771 & 2469 & 1100 & 1337 & 1450 & 1836 & 2034 & 2288 & 2625 & 2313 \\ 248 & 0 & 50 & 916 & 953 & 1116 & 1713 & 401 & 838 & 1684 & 2013 & 1738 & 1582 & 2221 & 1348 & 1445 & 1332 & 1718 & 1916 & 2040 & 2377 & 2065 \\ 298 & 50 & 0 & 866 & 903 & 1166 & 1663 & 451 & 888 & 1734 & 1998 & 1688 & 1632 & 2232 & 1398 & 1495 & 1382 & 1768 & 1966 & 2090 & 2427 & 2115 \\ 1164 & 916 & 866 & 0 & 37 & 614 & 797 & 1317 & 892 & 1182 & 1132 & 822 & 1636 & 1366 & 2264 & 2070 & 1957 & 1809 & 1638 & 1538 & 1612 & 1563 \\ 1201 & 953 & 903 & 37 & 0 & 577 & 760 & 1292 & 855 & 1145 & 1095 & 785 & 1599 & 1329 & 2270 & 2033 & 1920 & 1772 & 1601 & 1501 & 1575 & 1526 \\ 1364 & 1116 & 1166 & 614 & 577 & 0 & 626 & 715 & 278 & 568 & 897 & 623 & 1022 & 1105 & 1693 & 1456 & 1343 & 1195 & 1024 & 924 & 1261 & 949 \\ 1961 & 1713 & 1663 & 797 & 760 & 626 & 0 & 1341 & 904 & 664 & 335 & 25 & 1429 & 569 & 1992 & 1755 & 1642 & 1256 & 1058 & 750 & 815 & 775 \\ 649 & 401 & 451 & 1317 & 1292 & 715 & 1341 & 0 & 437 & 1283 & 1612 & 1338 & 1181 & 1820 & 1281 & 1044 & 931 & 1317 & 1515 & 1639 & 1976 & 1664 \\ 1086 & 838 & 888 & 892 & 855 & 278 & 904 & 437 & 0 & 846 & 1175 & 901 & 744 & 1383 & 1415 & 1178 & 1065 & 917 & 1115 & 1202 & 1539 & 1227 \\ 1932 & 1684 & 1734 & 1182 & 1145 & 568 & 664 & 1283 & 846 & 0 & 329 & 651 & 827 & 537 & 1390 & 1153 & 1040 & 654 & 456 & 356 & 693 & 381 \\ 2261 & 2013 & 1998 & 1132 & 1095 & 897 & 335 & 1612 & 1175 & 329 & 0 & 322 & 1094 & 234 & 1657 & 1420 & 1307 & 921 & 723 & 415 & 480 & 440 \\ 1986 & 1738 & 1688 & 822 & 785 & 623 & 25 & 1338 & 901 & 651 & 322 & 0 & 1416 & 556 & 1979 & 1742 & 1629 & 1243 & 1045 & 737 & 802 & 762 \\ 1771 & 1582 & 1632 & 1636 & 1599 & 1022 & 1429 & 1181 & 744 & 827 & 1094 & 1416 & 0 & 860 & 671 & 434 & 321 & 173 & 371 & 679 & 1016 & 704 \\ 2469 & 2221 & 2232 & 1366 & 1329 & 1105 & 569 & 1820 & 1383 & 537 & 234 & 556 & 860 & 0 & 1423 & 1186 & 1073 & 687 & 489 & 181 & 246 & 206 \\ 1100 & 1348 & 1398 & 2264 & 2270 & 1693 & 1992 & 1281 & 1415 & 1390 & 1657 & 1979 & 671 & 1423 & 0 & 237 & 350 & 736 & 934 & 1242 & 1579 & 1267 \\ 1337 & 1445 & 1495 & 2070 & 2033 & 1456 & 1755 & 1044 & 1178 & 1153 & 1420 & 1742 & 434 & 1186 & 237 & 0 & 113 & 499 & 697 & 1005 & 1342 & 1030 \\ 1450 & 1332 & 1382 & 1957 & 1920 & 1343 & 1642 & 931 & 1065 & 1040 & 1307 & 1629 & 321 & 1073 & 350 & 113 & 0 & 386 & 584 & 892 & 1229 & 917 \\ 1836 & 1718 & 1768 & 1809 & 1772 & 1195 & 1256 & 1317 & 917 & 654 & 921 & 1243 & 173 & 687 & 736 & 499 & 386 & 0 & 198 & 506 & 843 & 531 \\ 2034 & 1916 & 1966 & 1638 & 1601 & 1024 & 1058 & 1515 & 1115 & 456 & 723 & 1045 & 371 & 489 & 934 & 697 & 584 & 198 & 0 & 308 & 645 & 333 \\ 2288 & 2040 & 2090 & 1538 & 1501 & 924 & 750 & 1639 & 1202 & 356 & 415 & 737 & 679 & 181 & 1242 & 1005 & 892 & 506 & 308 & 0 & 337 & 25 \\ 2625 & 2377 & 2427 & 1612 & 1575 & 1261 & 815 & 1976 & 1539 & 693 & 480 & 802 & 1016 & 246 & 1579 & 1342 & 1229 & 843 & 645 & 337 & 0 & 362 \\ 2313 & 2065 & 2115 & 1563 & 1526 & 949 & 775 & 1664 & 1227 & 381 & 440 & 762 & 704 & 206 & 1267 & 1030 & 917 & 531 & 333 & 25 & 362 & 0 \end{bmatrix}$$

6.2.4 确定最短巡逻方案（遗传算法程序详见附件 9.10）

通过遗传算法解决 TSP 模型所得巡逻方案的最短路程为 7847m，其具体方案如下图所示（省略了其余路线）。



6.2.5 各服务点实时数据获取与处理（自动采集程序详见附件 9.11）

（1）数据获取

由于杭州市公交出行实时信息服务系统 (<http://www.hzbus.cn/>) 上的所谓实时数据并未进行实时的更新，因此我们选择了基于微信平台的杭州市公共自行车服务点实时查询系统进行数据采集。我们编写了相应的程序，在 6:00 至 21:00 每隔 10 分钟进行实时数据的采集。

我们采集的数据范围是自 2013 年 05 月 12 日起连续三天的 6:00 至 21:00 每隔 10 分钟全部 22 个服务点的实时自行车数据（原始数据详见 9.12）。

[注]微信平台网页版：<https://wx.qq.com/>。

（2）数据处理

数据采集完毕后，我们按如下步骤进行数据处理：

第一步，选择一个日期作为曲线拟合的基础参照，我们选择了 2013 年 05 月 14 日；

第二步，对 2013 年 05 月 14 日每一个服务点每一个时刻（除起始时刻）的可借自行车数据进行了前向差分的处理，得到每个时刻与前一个时刻还、借车需求的差值；

第三步，对 2013 年 05 月 14 日每一个服务点每一个时刻（除起始时刻）的自行车服务点是否处于理想状态进行标记。若处于满架或空架的非理想状态，则将该时刻标记为 1，否则标记为 0；

第四步，对于被标记为 1 的那些点，其前向差分数据应舍弃，用 2013 年 05 月 13 日同一个服务点同一个时刻的前向差分数据进行代替。若 2013 年 05 月 13 日同一个服务点同一个时刻仍然处于非理想状态，那么采用 2013 年 05 月 12 日的的数据；若三日的同一个服务点的同一个时刻均为非理想状态，则我们只能根据该时刻前后的自行车数量来对这个时刻进行估计（但该情况在本次数据处理中并未发生）。

第五步，以上三步中即可获得模型 I、模型 II 中所提到每个点一天内各个时刻的 $b_{ik} - a_{ik}$ 。对其进行时刻上的累加就可以得到我们所需要的 c_{ik} 曲线（详见附件 9.13）。

6.2.6 数据分析及模型求解

根据上面得到的 c_{ik} 曲线，通过模型 II 可以计算出每个自行车服务点的理想调整时间及调整数量。对所有需要调整的时刻按时间顺序列出（详见附件 9.14）。

假设一天内巡逻次数 T_m 的值，对子区域内所有服务点的理想调整时间进行聚类分析，得到一天内巡逻的各个时间段（详见附件 9.15）。

分别计算不同 T_m 的值下对应的不理想时间 $\sum Q_k$ （程序详见附件 9.16）。

T_m	2	3	4	5	6	7	8
$\sum Q_k$	428	357	284	220	186	153	139
T_m	9	10	11	12	13	14	15
$\sum Q_k$	116	101	87	68	43	21	19

最后将 T_m 与 $\sum Q_k$ 代入根据专家系统，得出最终所确定的方案。

7 模型的改进方向

我们对模型的第二个假设——不考虑 10 分钟之内的借还车顺序可以成为我们模型的一大改进放方向。

对实际情况进行考虑，对于 10 分钟之内“先借后还”与“先还后借”的极端情况，我们还需要对 c_j 进行进一步的约束使得该服务点在上述的极端情况下仍然能保持理想状态。

这里只需将约束条件改为 $L_j + b_{i+1} \leq c_i \leq L_j + N - a_{i+1}$ 即可。

8 参考文献

- [1] 《运筹学》教材编写组. 运筹学[M]. 第三版. 清华大学出版社, 2005.
- [2] (美) Kenneth H. Rosen. 离散数学及其应用[M]. 第六版. 机械工业出版社, 2011.