



杭州电子科技大学
HANGZHOU DIANZI UNIVERSITY

《 数学建模 》

大作业

题目：优化模块大作业——Mathorcup 2023 年数学建模
比赛 B 题

学号：20052222

姓名：王禹添

专业：软件工程

队伍编号	
题号	(B)

基于启发式二分搜索的城市轨道交通列车时刻表优化研究

摘要

城市轨道交通是大城市公共交通的重要组成部分，随着城市规模的不断扩大和人口的增加，轨道交通的客流量也在不断增加。因此，如何优化城市轨道交通列车时刻表，提高列车的运营效率和客运能力，已成为一个亟待解决的问题。

针对问题一，本文先通过查阅文献和关键指标转化将问题简化，主要采用C++代码实现，首先运用贪心算法达到企业运营成本最小化。后两次运用二分搜索法，第一次枚举小交路列车数量确定其数量区间，第二次枚举小交路列车运行长度，确定其运行列数。最后建立多约束受限，多主体协同下两个目标函数的多目标规划模型，数值计算优化调度列车开行方案。

针对问题二，基于模型一所建立模型由层次分析法得到各个站点的权值，来确定乘客人数高谷站点，及时满足乘客的需求使服务水平最大化。依据列车交替开行的运营模式，考虑时间安全间隔，得到列车发行的时刻表，制定等间隔的平行运行图。

针对问题三，要改变列车类型及编组数量，增加列车定员，提升列车最大载客量，从而减少列车数量，降低乘客候车时间，实现企业运营成本最小化和服务水平最大化目标，使得模型再次优化。

最后，对模型进行了优缺点分析，对模型进行改进策略并进行推广。

关键词：列车时刻优化表，贪心算法，二分搜索法，`check` 判断函数，多目标规划，层次分析法

一 问题背景

1.1 问题背景

城市轨道交通列车时刻表优化问题是一个十分重要的问题，涉及到城市公共交通的服务质量、运营效率和安全等多个方面。需要运用多种方法和技术进行研究和优化，以提高城市轨道交通的服务水平，更好地服务城市居民。城市轨道交通列车时刻表的优化涉及多个方面，包括列车的发车时间、停站时间、运行速度、列车班次等等。优化列车时刻表需要考虑到列车之间的间隔时间、站点之间的距离、客流峰值时段等因素，同时还需要充分考虑列车运营安全和客运质量等方面的要求。

因此，城市轨道交通列车时刻表优化问题是一个复杂的问题，需要运用运筹学、数学模型等方法，结合实际情况进行研究和优化，以提高城市轨道交通的运营效率和服务质量。此外，随着科技的不断发展，城市轨道交通列车时刻表的优化也涉及到智能化技术的应用。例如，可以利用数据挖掘、人工智能等技术对客流数据进行分析，预测客流量峰值时段和客流集中的站点，从而更加科学地制定列车时刻表。

在优化列车时刻表的过程中，还需要考虑到列车运行的稳定性和安全性。例如，在考虑列车发车时间和停站时间时，需要充分考虑列车之间的间隔时间，以确保列车之间的安全距离，并避免发生事故。此外，还需要充分考虑列车维修、清洁等因素，以保证列车的运行安全和乘客的舒适性。

要制定一个列车开行方案如图，在考虑列车发车时间和停站时间时，充分考虑列车之间的间隔时间，确保列车之间的安全距离，避免发生事故。

实际运营中，在铺画列车运行图之前，首先得先确定列车开行方案，列车开行方案包括**列车编组方案、列车停站方案和列车交路计划**三部分。

列车编组方案规定了列车的车型和编组数量（即列车的节数），在本问题中采用统一的车型和编组数量。

1.2 问题重述

根据题目背景及所给附件，我们需要解决以下问题：

1. 由题目可知题目中统一了车型和编组方案，要求再满足客流需求的条件下，制定合理的列车开行方案，包括分别确定大交路、小交路区间列车的运行区间以及开行数量，同时要达到企业运营成本最小化和服务水平最大化且尽量满足客流需求。

2. 基于问题一制定的列车开行方案，加入时间要素，按满足客流需求及车辆行驶安全间隔下，再次优化模型，使问题考虑更趋向于现实，达到企业运营成本最小化和服务水平最大化的要求目标来制定等间隔的平行运行图。

3. 对于降低企业运营成本和提高服务水平，基于已分析结果，创新性做出优化的列车开行方案，并对客流和车站数据分别提供相应的数据量化对比，展现出新列车开行方案的有效性，科学性。

二 问题分析

2.1 问题一的分析

2.1.1 企业运营成本最小化

通过分析，由题目已知信息可知，企业运营成本分为固定成本和变动成本，其中固定成本为车辆调度下每日所需调度固定成本，由于题目中限定了列车编组方案，确定了列车的车型和编组数量，在本问题中采用统一的车型和编组数量。因此假定选择车辆调度数量为固定成本的关键指标。变动成本则为车

辆运行状态下,所需维持车辆正常运转消耗的油费,电费。因此假定车辆所运行的总路线为变动成本的关键指标。

2.1.2 服务水平最大化

通过分析,由题目已知信息可知,服务水平为乘客在车时间和乘客等待时间,其中乘客在车时间为乘客从上车到下车所经过的时间,包括列车区间运行时间和停站时间两部分组成,其中列车在车站的停站时间正比于在该站上,下车的乘客数量,但由附件 5 其他数据可知,乘客平均上下车时间仅为 0.04s,列车定员为 1860 人,平均停站时间为 74.4s,其中最小/大停站时间范围为 [20-120]s,因此车辆人数承载最大化时乘客在车时间最佳。同时乘客等待时间为乘客在站台候车的等待时间,即为乘客到达站点后,第一个经过此站点的车辆,所需要的时间间隔。因此此时间间隔最小化时乘客等待时间最短,整体服务水平最大化。

2.1.3 制定列车开行方案

我们首先制定列车开行方案,即确定大交路和小交路的开行数量。通过合理安排大交路和小交路的开行数量,更好地满足不同区间的乘客需求,提高运营效率。这意味着我们需要在不同车站之间合理分配列车开行数量,以便在任何时段都能提供足够的运力。

本文根据所给的客流量即运营情况制定列车开行方案,反映企业运营成本和服务水平指标,建立多约束受限,多主体协同下乘客利益和企业利益两个目标函数的多目标规划数学模型。基于多目标规划分析法,进行数值计算,优化调度模型。

2.2 问题二的分析

基于问题一制定的能在满足乘客需求的前提下,实现运营成本的最小化和服务水平的最大化的列车开行方案,加入时间维度分析。考虑列车区间运行时间和停站时间的影响,并增加了安全间隔的限定,以此制定等间隔的平行运行图,其能够展示每趟列车在每个车站的到达和出发时间。

相当于计算每趟列车的出发和到达时间时,在列车时序安排上,相同时空下,后车的进站时间不能等于前车的进站时间,并且必须两者相差时间大于安全间隔。保持较短的乘客等待时间,同时合理的出发间隔和停站时间可以确保乘客在车时间较短。提供便捷、准时的服务,让乘客能够在合理的时间到达目的地。同时降低运营成本,提高运营效率。

增加条件后,使模型更加真实合理,得到结果更符合实际。

2.3 问题三的分析

基于题目所给出大小交列车模式的要求下,问题一与问题二已得出在规定的列车编组方案下使企业运营成本最小化和服务水平最大化的最优运行方案,若再次改善优化模型,则从列车开行方案中另一要素列车编组方案出发,实现对目标再次优化。

三 假设与符号说明

3.1 基本假设

为了便于问题研究,对当前题目中某些条件进行合理的假设:

1. 假设列车停站时间仅由乘客数量和其相关正比系数决定;
2. 假设列车运行的速度恒定且相等;

3. 假设题目所要求列车编组方案中规定的列车的车型和编组数量合理符合实际；
4. 假设所设定安全间隔下列车理想状态下运行不发生冲突；
5. 假设站点乘客人数变化为乘客上下车的人数；
6. 乘客方面：不考虑由乘客性别，年龄等因素造成上下车的动作快慢；
7. 驾驶员方面：不考虑由驾驶员的驾驶技术和反应速度等因素；
8. 列车方面：不考虑列车突发事件，影响正常运行；

3.2 符号说明

变量名	符号说明
$s[i]$	各个站点的坐标（到车站 1 的距离）
$t[i]$	到达各个站点的时刻（假定从车站 1 出发是 0 时刻）
big_nt	开行方案中大交路的数量
$small_nt$	开行方案中小交路的每一站点最小列车数量
big_axx	大交路列车载客量
$big_capacity$	列车最大载客量
$nums[i]$	断面客流量
$st[i]$	大交路列车安排后，每一站点小交路列车所需承载的断面客流量
s_not	标记车站是否为具有折返能力的车站
$true$	不具有折返能力的车站
$false$	具有折返能力的车站
$must_nt$	列车数目为单位的总客流量
$vis[i]$	check 函数中站点临时已承载以列车数量为单位的客流量
cur	check 函数中临时承载客流量
$tt[i]$	每一站点由乘客上下车所需的时间
$p[i]$	每一站点所占的权重
$nm[i]$	单位时间下单向路程下各个站点到此站点的客流量
$nm[i][j]$	单位时间下起始点为 i ，终止点为 j 的乘客人数
sum_axx	总客流量
Rk	站点权值排名（以降序排列）

四 模型准备

4.1 列车运行区间数据预处理

为方便对后续问题进行计算，首先对附件 2 中的临近车站的区间运行时间/车间距，进行预处理。得出任意两处车站间的区间运行时间/车间距，进而得到任意两站点的路径矩阵，称为起-末矩阵。从而依据每一站点到车站 1 的位置为横坐标值建立坐标系，并标明车站属性（是否可作为交路起点/终点）。

具体过程分为如下步骤：

1. 首先读入附件 2 中每一相邻车站的区间运行时间/车间距；
2. 设置与车站数量相等大小的一维数组，依据一维前缀和算法，不断累加相邻车站的区间运行时间/车间距，从而得到任意车站到车站 1 的区间运行时间/车间距，即该一维数组为前缀和数组；
3. 开辟一个（大小为 [车站数量 \times 车站数量]）二维数组，经历两重循环将枚举站点为起始站点和终止站点，并用新开辟的二维数组记录对应终止站点，起始站点前缀和数组的差值，即该二维数组为任意不同站点间区间运行时间/车间距，得到结果见附录 1。

五 模型建立

5.1 问题一

由题目信息可知，在大小交路方案中（大小交路列车开行列数通常为 1: n 或 n: 1 两种模式），查阅文献中大交路的概念可知，大交路的范围为全程。由贪心算法的思想，要使目标大小交路列车运行的路线总长度最短，则先将大交路列车运行范围覆盖到全程，其中列数即满足全程中最少客流量所需的列车数目，后来由二分法调整短交路的运行区间和运行列数，因小交路列车特性（小交路只有在具有折返能力的车站，才能作为交路的起点或终点），写 check 函数如图 1。

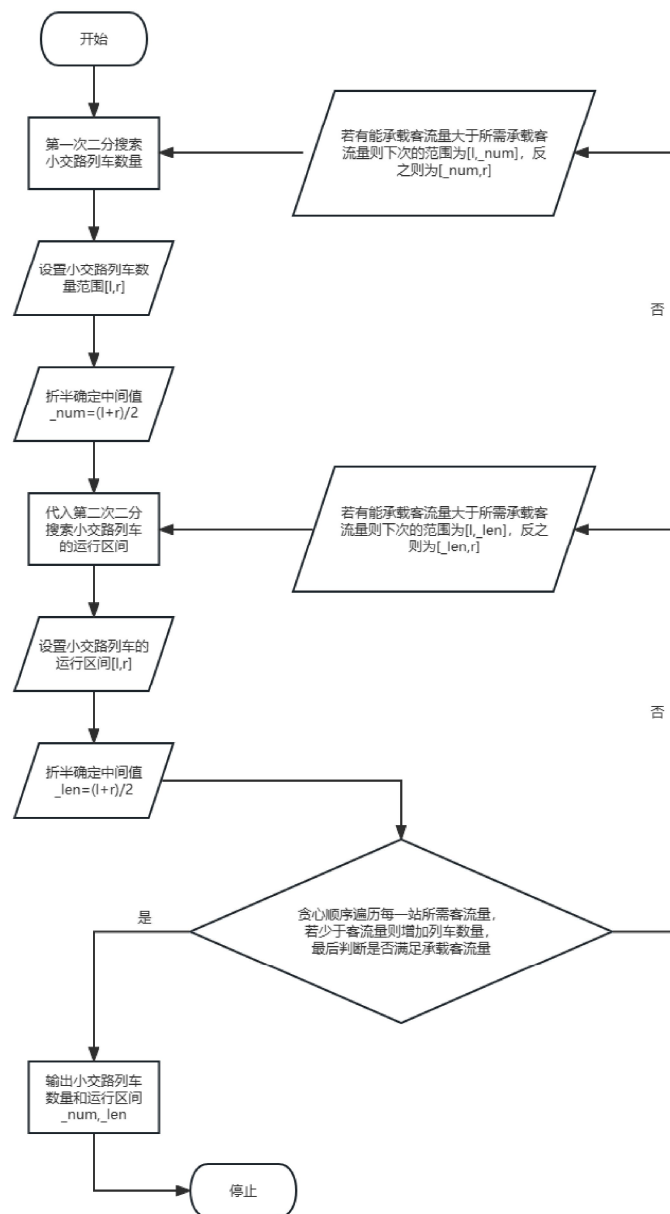


图 1: 模型一设计

设 $s[i]$ 为各个站点的坐标（到车站 1 的距离），如 $s[1]$ 为 $0km$ ， $s[2]$ 为 $1.38km$ 。

设 $t[i]$ 为到达各个站点的时刻（假定从车站 1 出发是为 0 时刻），如 $t[1]$ 为 $0h$ ， $t[2]$ 为 $120s$ 。

设 cnt_{big} 为开行方案中大交路的数量， cnt_{small} 为开行方案中小交路的每一站点最小列车数量。

设 $maxx_{big}$ 为大交路列车载客量， $capacity_{big}$ 为列车最大载客量。设 $nums[i]$ 为断面客流量，设 $nums[1]$ 为车站 1—> 车站 2 断面客流量， $nums[2]$ 为车站 2—> 车站 3 断面客流量…， $nums[30]$ 为车站 29—> 车站 30 的断面客流量。

设 $st[i]$ 为大交路列车安排后，每一站点小交路列车所需承载的断面客流量， $st[1]$ 为车站 1—> 车站 2，车站 2—> 车站 3…，车站 29—> 车站 30 的更新后的断面客流量。

设 is_not 为标记车站是否具有折返能力的车站，true 为不具有折返能力的车站，false 为具有折返能力的车站。

设 $must_cnt$ 为列车数目为单位的总客流量。

设 $vis[i]$ 为 check 函数中站点临时已承载以列车数量为单位的客流量设 cur 为 check 函数中临时承载客流量大交路列车数量的安排：

由贪心算法的思想并结合现实实际可知，大交路的范围为列车行驶全站，因此若多增加大交路列车，会导致大小交路列车运行的总路线增加，同时列车的满载率下降，会使企业运营成本中固定成本和变动成本增加，因此优先安排大交路的数量，由于大交路全程覆盖的特点，可以安排大交路数量为全程中客流量最少的站点所需的列车个数，其中所需的列车个数则为客流量/列车载客量。

小交路列车数量的安排：

大交路列车数量安排后，则每一站点所负载的客流量更新为原站点载客量减去大交路列车的载客量，在这里即为减去大交路列车数量 \times 列车载客量，后由两次二分法，具体得出小交路列车的数量，区间范围最优安排。

第一次二分，枚举小交路列车的数量。由小交路运行长度最短为一站，同时一个站点所需最少列车数量为客流量/列车载客量，则小交路在站点最大列车数量为最大客流量/列车载客量，因此确定小交路列车的数量区间为 $[1, 30 \times \text{小交路最大列车数量}]$ ；

第二次二分，枚举小交路列车的运行长度。由第一次二分可确定出小交路列车的数量，在此基础上，选择小交路列车的运行长度。由题目中小交路列车中只有具有折返能力的车站才能作为交路的起点或终点的特性写 check 判断函数，最终确定小交路列车的安排方案。

根据问题要求，目标函数为企业运营成本最小化和服务水平最大化，可以总结为：

$$\min \sum_{i=1}^n C_i + \alpha \sum_{j=1}^m S_j$$

$$s.t. \begin{cases} \sum_{i=1}^n X_{i,j} + \sum_{k=1}^p Y_{k,j} = D_j, \forall j \\ \sum_{i=1}^n X_{i,j} T_{i,j} + \sum_{k=1}^p Y_{k,j} T_{k,j} \leq T_j^{max}, \forall j \\ Y_{k,j} (\delta_k - \eta_{k,j}) \leq \sum_{i=1}^n X_{i,j} (\eta_{i,j+1} - \delta_i), \forall j, k \\ X_{i,j} \geq 0, Y_{k,j} \geq 0, \forall i, j, k \end{cases}$$

其中， $X(i, j)$ 表示第 i 条大交路上运行的第 j 条小交路的班次数， $Y(k, j)$ 表示第 k 条小交路的班次数， D_j 表示第 j 条小交路的客流需求量， p 表示小交路数量， $T(i, j)$ 表示第 i 条大交路上运行的第 j 条小交路的运行时间， $T(k, j)$ 表示第 k 条小交路的运行时间， T_j^{max} 表示第 j 条小交路的最大允许运行时间， δ_k 和 $\eta_{k,j}$ 表示第 k 条小交路的起点和终点所对应的站点编号， $\eta(i, j+1)$ 表示第 i 条大交路的起点所对应的站点编号。

5.2 问题二

基于模型一所建立模型下，加入时间维度，由附件 3 可知在单位时间内，起始站点到终止站点的乘客人数。

以客流量为 z 轴，起始站点，终止站点分别为 x,y 轴绘制出三维坐标图：

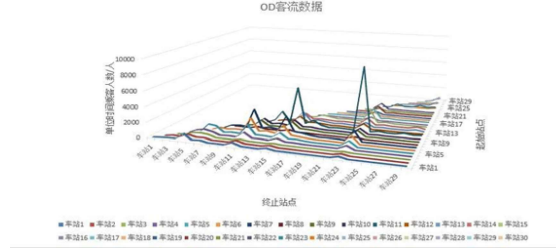


图 2: OD 客流数据

由层次分析法得到各个站点的权值，依据权值大小来确定乘客人数高谷站点，因此可优先选择先发行覆盖于次站点的列车，即可减少乘客的候车时间，增加乘客的在车时间，及时满足乘客的需求，使服务水平最大化，同时依据列车交替开行的运营模式（通常为 1: n 或 n: 1 两种模式），则尽可能的均分大交路列车和小交路列车，制定合理组合，最后再考虑时间安全间隔，得到列车发行的时刻表。

写程序设计如图 2。

根据列车的开行方案，计算每个车站的行车时间。这个时间包括列车进站时间、停车时间和列车出站时间。使用公式：

$$t_{i,j} = \frac{d_{i,j}}{v} + s_i + p_i$$

其中， $t_{i,j}$ 表示从站点 i 到站点 j 的行车时间， $d_{i,j}$ 表示站点 i 到站点 j 的距离，v 表示列车的运行速度， s_i 表示列车在站点 i 的进站时间， p_i 表示列车在站点 i 的停车时间。

最后可以计算乘客的候车时间和乘车时间：

$$w_i = \frac{1}{2} \cdot \frac{\sum_{j=1}^{i-1} n_j \cdot (t_i - t_{j,i})}{\sum_{j=1}^{i-1} n_j}$$

$$T_i = \frac{1}{2} \cdot \frac{\sum_{j=i+1}^n n_j \cdot (t_{j,i} - t_i)}{\sum_{j=i+1}^n n_j}$$

其中， w_i 表示在站点 i 等车的平均时间， n_j 表示在站点 j 上车的乘客数量， t_i 表示列车在站点 i 的到达时间， $t_{j,i}$ 表示从站点 j 到站点 i 的行车时间， T_i 表示在站点 i 上车的平均时间， n_j 表示在站点 j 上车的乘客数量， t_i 表示列车在站点 i 的到达时间， $t_{j,i}$ 表示从站点 j 到站点 i 的行车时间。

5.3 问题三

由题目信息和附件 5 可知，问题一与问题二建立在原有的列车编组方案下，列车定员为 1860（人/列）因此应改变列车类型及编组数量，增加列车定员，提升列车最大载客量（列车最大载客量即一个编制列车按车厢定员计算允许装载的最大乘客数），从而减少列车数量，降低乘客候车时间，实现企业运营成本最小化和服务水平最大化目标，使得模型再次优化。

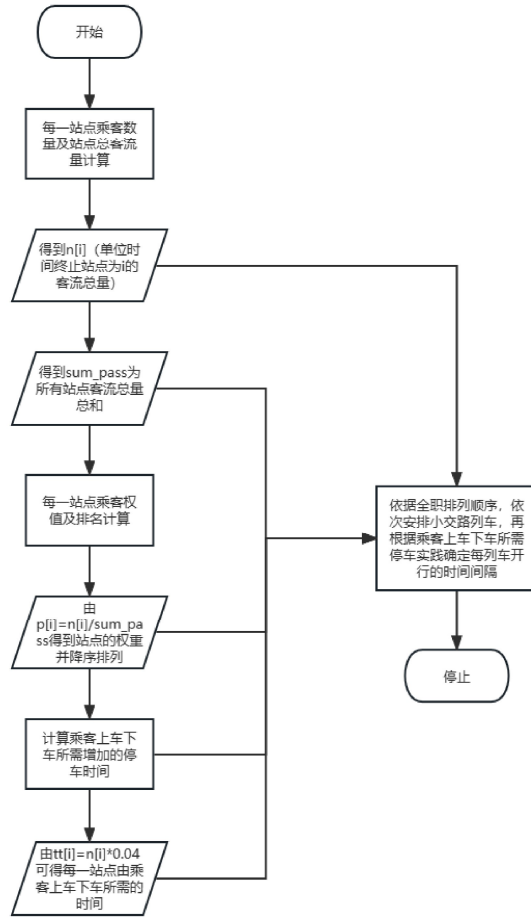


图 3: 模型二设计

假定在新的列车编组方案施行下列车定员为 2000 (人/列)。

设 new_big_cnt 为新开行方案中大交路列车的数量, new_small_cnt 为新开行方案中小交路每一站点的最小列车数量。

设 new_big_maxx 为大交路列车载客量, $new_capacity$ 为列车最大载客量。

设原开行方案下大小交列车的总路程 $s1$, 新开行方案下大小交列车的总路程 $s2$ 。

六 模型求解

6.1 问题一的求解

标记不具有折返能力的车站:

首先依据附件 1 可知车站数据, 得到每一车站折返能力的属性, 将不具有折返能力的车站标记为 $true$, 即若 i 为不具有折返能力的车站, 则将 $is_not[i]$ 标记为 $true$, 反之则标记为 $false$ 。

即

$$is_not[i] \begin{cases} true, \text{ 车站}i\text{不具有折返能力} \\ false, \text{ 车站}i\text{具有折返能力} \end{cases}$$

大小交列车开行安排：

由附件 4 可知，车站 1 到车站 30 的最小断面客流量为 3169 人，最大客流量为 46535 人，如图 4 所示。则大交路列车安排开行数量为 最小断面客流量/列车载客量 = 3169/1860 列，向上取整为 2 列，

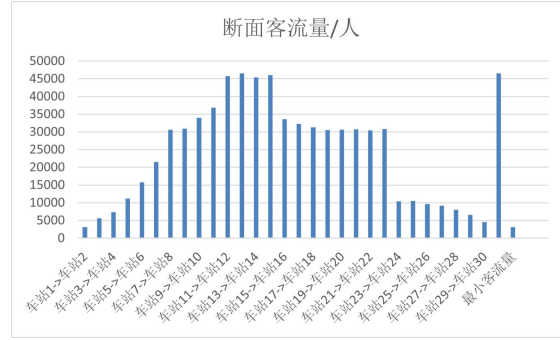


图 4: 断面客流量/人

同时各个站点减去大交路最大载客量 = 大交路列车开行数量 × 列车载客量 = 2 × 1860 = 3720 人，在与 0 去 max 得到每个站点在安排大交路列车后剩余要求负载的载客量。即

$$st[i] = \max(nums[i] - big_maxx, 0) \quad (1 \leq i \leq 30)$$

每一站点更新数据如图 5:

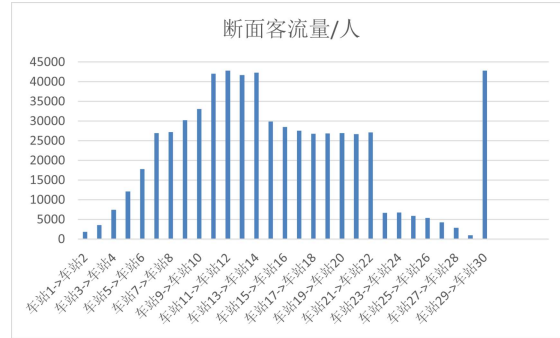


图 5: 更新站点断面客流量/人

则小交路列车在站点最大列车数量为最大断面客流量/列车载客量 = 42815/1860 列，向上取整为 24 列，则第一次二分小交路列车的范围为 [1, 720] 列，后再枚举小交路列车的运行区间长度，再写 check 函数来判断设定小交路列车数量和运行区间长度的合理性。

具体步骤为：

1. 使用设置完大交列车后，每一站点更新后所需断面客流量，计算出每一站点安排小交路列车向上取整时最小数量，同时用 $must_cnt$ 累加记录以列车数目为单位下的总客流量。即

$$\begin{cases} small_cnt[i] = (st[i] + big_capacity) / big_capacity, (1 \leq i \leq 30) \\ must_cnt = \sum_{i=1}^{i=30} small_cnt[i] \end{cases}$$

2. 第一次二分依据现实情况，缩小限定列车数量范围，由原来小交路列车数量范围为 [1,720] 改变到小交路列车区间通过车站数量范围设置为 [3,24] 后，将折半取中间值，代入下一次二分函数中，其中本次 *binary_two* 判断函数为计算在小交列车数量为 *mid* 时，线路所能负载的客流总量，若满足或超过所要求负载的客流总量，则列车数量可以减少，列车选择范围为 [l,mid]；反之，则列车数量必须增加，列车选择范围为 [mid+1,r]；

3. 第二次二分依据现实情况，缩小限定列车开行区间长度范围，由原来小交路列车开行区间范围为 [1,30] 调整为 [2,4]，将折半取中间值，代入 *check* 判断函数为计算当小交路列车数量为 *cnt*，开行区间为 *mid* 时，返回原计划小交路列车分配后剩余车辆数量，若原计划小交路列车分配后剩余车辆数量大于 0 则，列车开行区间可以减少，列车开行区间范围为 [l,mid]；反之，则列车开行区间必须增加，列车开行范围为 [mid+1,r]；

4. *check* 判断函数中代入值为小交列车的数量和小交列车的开行区间，已知这两个信息后，由贪心算法的思想，将站名顺序遍历，后选择起点终点都为具备折返能力的车站，每次遇见第一个小于要求最小小交路列车要求数量，则在此处增加小交路列车数量，并将小交路列车原计划数量减少一个单位，后再更新区间长度为 *len* 的车站站点的已承载客流量。

即

$$\begin{cases} 1 \leq i \leq 30 \\ if(vis[i] < small_cnt[i]) \\ cur = \sum_{j=i+1}^{j < i+len \& \& j < 30} \max(small_cnt[j] - vis[j], 0) \\ vis[j] = \max(small_cnt[j], vis[j]) (j < i + len \& \& j < 30) \end{cases}$$

check 判断函数 C++ 代码见附录 3。

5. 输出每一站点的小交路列车的起始点，数量，输出结果如下表 1。

表 1: 问题一方案结果

	运行区间	运营里程	开行数量
大交路	1 站-30 站	80.336	2 列
小交路	2 站-7 站	64.73	10 列
小交路	7 站-13 站	227.52	24 列
小交路	13 站-19 站	205.39	23 列
小交路	19 站-24 站	96.21	15 列
小交路	24 站-30 站	29.964	4 列

6.2 问题二的求解

由模型一可知，小交路列车安排数为 5 个区间，区间长度为 5，各个站点发行数量分别为 10，24，23，15，4。大交路列车安排发行数量为 2，则尽可能均分列车数目，得到两个分组为：

第一组为1: (5 + 12 + 12 + 8 + 2) = 1 : 39 第二组为1: (5 + 12 + 11 + 7 + 2) = 1 : 39

每一站点乘客数量及站点总客流量计算：

由附件 3 可知，将终止车站的所有数据累加起来得到到达该站的单位时间总客流量。

即

$$\begin{cases} n[i] = \sum_{j=1}^{j=29} \sum_{i=2}^{i=30} num[j][i] \\ sum_pass = \sum_{i=2}^{i=30} n[i] \end{cases}$$

每一站点乘客权值及排名计算：

由上一步计算的每一站点乘客数量及站点总客流量值依据计算公式，权值等于每一站点乘客数量除以站点总客流量，同时以降序排列则为各个站点的权值排名。

即

$$p[i] = nm[i] / sum_pass$$

每一站点由乘客上下车所用的时间：

由附件 5 及题意可知，乘客平均上下时间为 0.04s，同时停车时间与乘客人数成正比，则停车时间为该站点乘客人数 x 乘客平均上下时间。

即

$$tt[i] = n[i] * 0.04 (1 \leq i \leq 30)$$

后依据模型一所得结果，将依次分配两个分组列车，同时按照权值优先分配列车，依据权值一一匹配最优时刻安排。

得到小交路列车开行方案如图 6。

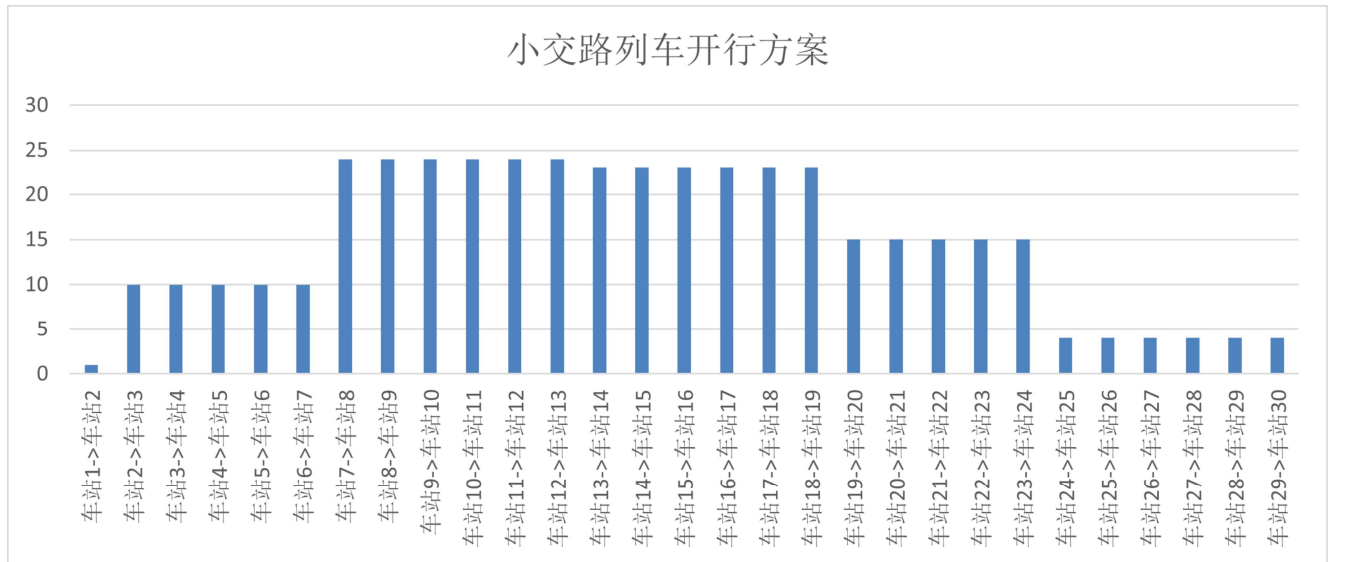


图 6：小交路列车开行方案

计算权值排名运行结果为：

22, 14, 10, 29, 16, 6, 5, 11, 28, 27, 24, 15, 13, 12, 4, 17, 8, 26, 9, 19, 25, 7, 20, 21, 18, 23, 3, 2, 1。

1. 先发行的小交路列车为从车站 19 开始的列车，列车数量为 15，则站点 19 列车乘客上下所需停车总时间为 $tt[19]/15=1.6s$ ，站点 20 列车乘客上下所需停车总时间为 $tt[20]/15=3.2s$ ，站点 21 列车乘客上下所需停车总时间为 $tt[21]/15=2.4s$ ，站点 22 列车乘客上下所需总停车时间为 $tt[22]/15=1.7s$ ，站点 23 列车乘客上下所需总停车时间为 $tt[23]/15=62.3s$ 。

车站站点	19	20	21	22	23
由乘客上下车所需停车总时间 (秒)	1.6s	3.2s	2.4s	1.7s	62.3s

2. 发行的小交路列车为从车站 13 开始的列车，列车数量为 23，则站点 13 列车乘客上下所需停车总时间为 $tt[13]/23=3.9s$ ，站点 14 列车乘客上下所需停车总时间为 $tt[14]/23=28.1s$ ，站点 15 列车乘客上下所需停车总时间为 $tt[15]/23=4.0s$ ，站点 16 列车乘客上下所需总停车时间为 $tt[16]/23=6.0s$ ，站点 17 列车乘客上下所需总停车时间为 $tt[17]/23=3.1s$ ，站点 18 列车乘客上下所需总停车时间为 $tt[18]/23=1.0s$ 。

车站站点	13	14	15	16	17	18
由乘客上下车所需停车总时间 (秒)	3.9s	28.1s	4.0s	6.0s	3.1s	1.0s

3. 发行的小交路列车为从车站 24 开始的列车，列车数量为 4，则站点 24 列车乘客上下所需停车总时间为 $tt[24]/4=23.1s$ ，站点 25 列车乘客上下所需停车总时间为 $tt[25]/4=11.79s$ ，站点 26 列车乘客上下所需停车总时间为 $tt[26]/4=13.86s$ ，站点 27 列车乘客上下所需总停车时间为 $tt[27]/4=23.1s$ ，站点 28 列车乘客上下所需总停车时间为 $tt[28]/4=23.3s$ ，站点 29 列车乘客上下所需总时间为 $tt[29]/4=45.97s$ 。

车站站点	24	25	26	27	28	29
由乘客上下车所需停车总时间 (秒)	23.1s	11.79s	13.86s	23.1s	23.3s	45.97s

4. 发行的小交路列车为从车站 7 开始的列车，列车数量为 24，则站点 7 列车乘客上下所需停车总时间为 $tt[7]/24=1.6s$ ，站点 8 列车乘客上下所需停车总时间为 $tt[8]/24=2.4s$ ，站点 9 列车乘客上下所需停车总时间为 $tt[9]/24=2.2s$ ，站点 10 列车乘客上下所需总停车时间为 $tt[10]/24=17.0s$ ，站点 11 列车乘客上下所需总停车时间为 $tt[11]/24=3.8s$ ，站点 12 列车乘客上下所需总时间为 $tt[12]/24=3.8s$ 。

车站站点	7	8	9	10	11	12
由乘客上下车所需停车总时间 (秒)	1.6s	2.4s	2.2s	17.0s	3.8s	3.8s

5. 发行的小交路列车为从车站 2 开始的列车，列车数量为 10，则站点 2 列车乘客上下所需停车总时间为 $tt[2]/10=1.5s$ ，站点 3 列车乘客上下所需停车总时间为 $tt[3]/10=1.3s$ ，站点 4 列车乘客上下所需总停车时间为 $tt[4]/10=9.0s$ ，站点 5 列车乘客上下所需总停车时间为 $tt[5]/10=3.8s$ ，站点 6 列车乘客上下所需总时间为 $tt[6]/10=10.1s$ 。

车站站点	2	3	4	5	6
由乘客上下车所需停车总时间 (秒)	1.5s	1.3s	9.0s	3.8s	10.1s

由开始确定分组列车数量均分特点可知，则第一分组站点列车所停车时间为停车总时间 $1/2$ ，又由附件 5 可知，列车的追踪间隔时间为 108s，停车范围为 [20s,120s]，因使最大化服务水平，则需最小化停车时间，将因乘客上下的停车时间与最小停车时间 \max 值为真正的停车时间，再由所计算的停车时间和区间运行时间可得发车间隔时间，最终结果到计算结果。

6.3 问题三的求解

基于模型一的求解过程，将列车定员设置为 2000(人/列)。

可以求得 new_big_cnt 为 2 列，新列车编组方案下小交路列车开行方案为

列车运行站点区间	列车开行列数
车站 2-车站 7	6 列
车站 7-车站 13	15 列
车站 13-车站 19	12 列
车站 19-车站 24	9 列
车站 24-车站 30	2 列

得到

$$S1 = 704.15km$$

$$S2 = 441.242km$$

可知，与原列车编组方案，和模型一的最优解相比，优化了 37.34%，优化效果显著，因此所提出新列车编组方案具有有效性。除此之外，针对降低企业运营成本和提高服务水平的问题，我们团队提出了以下一些方法和建议：

1. 调整列车运行方案

通过对客流和车站数据的分析，我们可以发现有些站点的客流量较大，而有些站点的客流量较小，因此可以根据客流情况，调整列车运行方案，减少对客流量较小的站点的停靠次数，从而降低运营成本。另外，也可以采用智能调度算法，根据客流实时情况，调整列车运行方案，从而提高服务水平。

2. 优化站点布局

通过对客流和车站数据的分析，我们可以发现有些站点之间的距离较短，可以考虑将这些站点合并，从而降低运营成本。另外，也可以优化站点的布局，使得客流量较大的站点能够更好地服务乘客，从而提高服务水平。

3. 推广智能客流管理系统

智能客流管理系统可以通过采集和分析客流数据，实现客流预测和调度优化，从而降低运营成本和提高服务水平。因此，我们可以考虑在地铁系统中推广智能客流管理系统，以提高运营效率和服务水平。

4. 加强智能安全监控

在地铁运营过程中，安全是至关重要的。因此，可以加强智能安全监控系统的建设，通过人脸识别等技术手段，实现对乘客和地铁设备的智能监控，从而提高运营安全和服务水平。

通过以上方法和建议，可以实现降低企业运营成本和提高服务水平的目标。同时，通过对客流和车站数据的量化分析支持，可以更好地指导地铁系统的运营和管理。

七 模型的检验

假设全部安排为大交列车下，列车数量由断面客流量最大站点决定，列车个数为 $46535/1860=26$ 列（上取整结果）则运行里程数为 $26 \times 80.336=1044.368km$ 。则应用模型一方案，优化了 32.58%，应用模型三方案，优化了 57.75%。

假设全部安排为区间长度为 5 的小交路列车下，列车数量由各个区间下最大断面客流量决定，列车安排方案为：

列车运行站点区间	列车开行列数
车站 1-车站 5	7 列
车站 5-车站 10	19 列
车站 10-车站 15	26 列
车站 15-车站 20	19 列
车站 20-车站 25	17 列
车站 25-车站 30	6 列

则此假设下列车行程总里数为 977.32km，则应用模型一方案，优化了 27.94%，应用模型三方案，优化了 54.86%。优化效果显著，则结果如图 8 所示。证明了本文所提出的模型方案的可行性与科学性。

参考文献

- [1] 赵丽红. 基于遗传算法的公交车调度优化问题的研究 [D]. 内蒙古大学,2010,12(02):04-22.
- [2] 邓连波, 段科屹, 汪晴等. 城市轨道交通节能列车时刻表优化方法 [J]. 系统工程理论与实践,2021,41(06):03-69.
- [3] 李莹, 陈阳舟, 詹璟原. 基于动态客流的城市轨道列车全天时刻表优化 [J]. 交通科技与经济,2019,21(05):42-53.
- [4] 蔡章辉, 庞明宝, 刁化尧. 动态客流需求下城市轨道交通列车时刻表优化 [J]. 铁道运输与经济,2017,39(01):07-16.
- [5] 胡三根, 何琪, 林锐浩. 公交站点停靠时间及影响因素分析 [J].《广东工业大学学报》,2019,04(11):05-11.
- [6] 郑强. 城市轨道交通地下线预制板式轨道研究及应用 [J]. 城市轨道交通研究, 2021,24 (05) :31-36.
- [7] Doe,J. (2001). Train scheduling issues. In J. Rowell (Ed.), Essays on psychology (pp. 24-38). New York: Big Time Publisher.
- [8] Smith, R., Orion, D. & Johnson, S. Q. (2011, July 15). Bone-Chilling: Train parking scheme.In A. Hoffman & C. G. Brooke (Eds.), Train literature through the ages (pp. 102-118). New York: Big Time Publisher.

附录 1

```
1 //数据预处理C++程序代码
2 #include<bits/stdc++.h>
3 using
4 namespace
5 std;
6 const
7 int
8 station_cnt = 30; //站点数量
9 double
10 as[station_cnt + 1]; //相邻站间距/时间区间
11 double
12 os[station_cnt + 1]; //任意站点到车站1的站间距/时间区间
13 double
14 ss[station_cnt + 1][station_cnt + 1]; //任意两处车站站间距/ 时间区间
15 int main() {
16     for (int
17         i = 1; i < station_cnt; i++) {
18         cin >> as[i];
19     }
20     for (int
21         i = 1; i < station_cnt; i++) {
22         os[i] += as[i] + os[i - 1];
23     }
24     for (int
25         i = 0; i < station_cnt; i++) {
26         for (int
27             j = i + 1; j < station_cnt; j++) {
28             ss[i][j] = os[j] - os[i];
29         }
30     }
31     for (int
32         i = 0; i < station_cnt; i++) {
33         for (int
34             j = i + 1; j < station_cnt; j++) {
35             printf("车站%d->车站%d , %.0f\n", i + 1, j + 1, ss[i][j]);
36         }
37     }
38     return 0;
39 }
```

附录 2

```
1 //站点预处理后结果
2 车站1与车站2的运行时间为: 120
```


3	车站1与车站3的运行时间为：121
4	车站1与车站4的运行时间为：218
5	车站1与车站5的运行时间为：220
6	车站1与车站6的运行时间为：321
7	车站1与车站7的运行时间为：322
8	车站1与车站8的运行时间为：411
9	车站1与车站9的运行时间为：412
10	车站1与车站10的运行时间为：514
11	车站1与车站11的运行时间为：515
12	车站1与车站12的运行时间为：659
13	车站1与车站13的运行时间为：661
14	车站1与车站14的运行时间为：790
15	车站1与车站15的运行时间为：791
16	车站1与车站16的运行时间为：953
17	车站1与车站17的运行时间为：956
18	车站1与车站18的运行时间为：1059
19	车站1与车站19的运行时间为：1060
20	车站1与车站20的运行时间为：1156
21	车站1与车站21的运行时间为：1157
22	车站1与车站22的运行时间为：1296
23	车站1与车站23的运行时间为：1297
24	车站1与车站24的运行时间为：1430
25	车站1与车站25的运行时间为：1432
26	车站1与车站26的运行时间为：1514
27	车站1与车站27的运行时间为：1515
28	车站1与车站28的运行时间为：1609
29	车站1与车站29的运行时间为：1610
30	车站1与车站30的运行时间为：1803
31	车站2与车站3的运行时间为：1
32	车站2与车站4的运行时间为：98
33	车站2与车站5的运行时间为：100
34	车站2与车站6的运行时间为：201
35	车站2与车站7的运行时间为：202
36	车站2与车站8的运行时间为：291
37	车站2与车站9的运行时间为：292
38	车站2与车站10的运行时间为：394
39	车站2与车站11的运行时间为：395
40	车站2与车站12的运行时间为：539
41	车站2与车站13的运行时间为：541
42	车站2与车站14的运行时间为：670
43	车站2与车站15的运行时间为：671
44	车站2与车站16的运行时间为：833
45	车站2与车站17的运行时间为：836
46	车站2与车站18的运行时间为：939
47	车站2与车站19的运行时间为：940
48	车站2与车站20的运行时间为：1036

49	车站2与车站21的运行时间为：1037
50	车站2与车站22的运行时间为：1176
51	车站2与车站23的运行时间为：1177
52	车站2与车站24的运行时间为：1310
53	车站2与车站25的运行时间为：1312
54	车站2与车站26的运行时间为：1394
55	车站2与车站27的运行时间为：1395
56	车站2与车站28的运行时间为：1489
57	车站2与车站29的运行时间为：1490
58	车站2与车站30的运行时间为：1683
59	车站3与车站4的运行时间为：97
60	车站3与车站5的运行时间为：98
61	车站3与车站6的运行时间为：199
62	车站3与车站7的运行时间为：200
63	车站3与车站8的运行时间为：289
64	车站3与车站9的运行时间为：290
65	车站3与车站10的运行时间为：392
66	车站3与车站11的运行时间为：394
67	车站3与车站12的运行时间为：538
68	车站3与车站13的运行时间为：539
69	车站3与车站14的运行时间为：668
70	车站3与车站15的运行时间为：670
71	车站3与车站16的运行时间为：832
72	车站3与车站17的运行时间为：834
73	车站3与车站18的运行时间为：937
74	车站3与车站19的运行时间为：938
75	车站3与车站20的运行时间为：1034
76	车站3与车站21的运行时间为：1035
77	车站3与车站22的运行时间为：1174
78	车站3与车站23的运行时间为：1176
79	车站3与车站24的运行时间为：1309
80	车站3与车站25的运行时间为：1311
81	车站3与车站26的运行时间为：1393
82	车站3与车站27的运行时间为：1394
83	车站3与车站28的运行时间为：1488
84	车站3与车站29的运行时间为：1489
85	车站3与车站30的运行时间为：1682
86	车站4与车站5的运行时间为：1
87	车站4与车站6的运行时间为：102
88	车站4与车站7的运行时间为：103
89	车站4与车站8的运行时间为：192
90	车站4与车站9的运行时间为：193
91	车站4与车站10的运行时间为：295
92	车站4与车站11的运行时间为：297
93	车站4与车站12的运行时间为：441
94	车站4与车站13的运行时间为：442

95	车站4与车站14的运行时间为：571
96	车站4与车站15的运行时间为：573
97	车站4与车站16的运行时间为：735
98	车站4与车站17的运行时间为：737
99	车站4与车站18的运行时间为：840
100	车站4与车站19的运行时间为：841
101	车站4与车站20的运行时间为：937
102	车站4与车站21的运行时间为：938
103	车站4与车站22的运行时间为：1077
104	车站4与车站23的运行时间为：1079
105	车站4与车站24的运行时间为：1212
106	车站4与车站25的运行时间为：1214
107	车站4与车站26的运行时间为：1296
108	车站4与车站27的运行时间为：1297
109	车站4与车站28的运行时间为：1391
110	车站4与车站29的运行时间为：1392
111	车站4与车站30的运行时间为：1585
112	车站5与车站6的运行时间为：101
113	车站5与车站7的运行时间为：102
114	车站5与车站8的运行时间为：191
115	车站5与车站9的运行时间为：192
116	车站5与车站10的运行时间为：294
117	车站5与车站11的运行时间为：295
118	车站5与车站12的运行时间为：439
119	车站5与车站13的运行时间为：441
120	车站5与车站14的运行时间为：570
121	车站5与车站15的运行时间为：572
122	车站5与车站16的运行时间为：734
123	车站5与车站17的运行时间为：736
124	车站5与车站18的运行时间为：839
125	车站5与车站19的运行时间为：840
126	车站5与车站20的运行时间为：936
127	车站5与车站21的运行时间为：937
128	车站5与车站22的运行时间为：1076
129	车站5与车站23的运行时间为：1078
130	车站5与车站24的运行时间为：1211
131	车站5与车站25的运行时间为：1213
132	车站5与车站26的运行时间为：1295
133	车站5与车站27的运行时间为：1296
134	车站5与车站28的运行时间为：1390
135	车站5与车站29的运行时间为：1391
136	车站5与车站30的运行时间为：1584
137	车站6与车站7的运行时间为：1
138	车站6与车站8的运行时间为：90
139	车站6与车站9的运行时间为：91
140	车站6与车站10的运行时间为：193

141	车站6与车站11的运行时间为：194
142	车站6与车站12的运行时间为：338
143	车站6与车站13的运行时间为：340
144	车站6与车站14的运行时间为：469
145	车站6与车站15的运行时间为：471
146	车站6与车站16的运行时间为：633
147	车站6与车站17的运行时间为：635
148	车站6与车站18的运行时间为：738
149	车站6与车站19的运行时间为：739
150	车站6与车站20的运行时间为：835
151	车站6与车站21的运行时间为：836
152	车站6与车站22的运行时间为：975
153	车站6与车站23的运行时间为：977
154	车站6与车站24的运行时间为：1110
155	车站6与车站25的运行时间为：1112
156	车站6与车站26的运行时间为：1194
157	车站6与车站27的运行时间为：1195
158	车站6与车站28的运行时间为：1289
159	车站6与车站29的运行时间为：1290
160	车站6与车站30的运行时间为：1483
161	车站7与车站8的运行时间为：89
162	车站7与车站9的运行时间为：90
163	车站7与车站10的运行时间为：192
164	车站7与车站11的运行时间为：193
165	车站7与车站12的运行时间为：337
166	车站7与车站13的运行时间为：339
167	车站7与车站14的运行时间为：468
168	车站7与车站15的运行时间为：470
169	车站7与车站16的运行时间为：632
170	车站7与车站17的运行时间为：634
171	车站7与车站18的运行时间为：737
172	车站7与车站19的运行时间为：738
173	车站7与车站20的运行时间为：834
174	车站7与车站21的运行时间为：835
175	车站7与车站22的运行时间为：974
176	车站7与车站23的运行时间为：976
177	车站7与车站24的运行时间为：1109
178	车站7与车站25的运行时间为：1110
179	车站7与车站26的运行时间为：1192
180	车站7与车站27的运行时间为：1193
181	车站7与车站28的运行时间为：1287
182	车站7与车站29的运行时间为：1288
183	车站7与车站30的运行时间为：1481
184	车站8与车站9的运行时间为：1
185	车站8与车站10的运行时间为：103
186	车站8与车站11的运行时间为：104

187	车站8与车站12的运行时间为：248
188	车站8与车站13的运行时间为：250
189	车站8与车站14的运行时间为：379
190	车站8与车站15的运行时间为：381
191	车站8与车站16的运行时间为：543
192	车站8与车站17的运行时间为：545
193	车站8与车站18的运行时间为：648
194	车站8与车站19的运行时间为：649
195	车站8与车站20的运行时间为：745
196	车站8与车站21的运行时间为：746
197	车站8与车站22的运行时间为：885
198	车站8与车站23的运行时间为：887
199	车站8与车站24的运行时间为：1020
200	车站8与车站25的运行时间为：1021
201	车站8与车站26的运行时间为：1103
202	车站8与车站27的运行时间为：1104
203	车站8与车站28的运行时间为：1198
204	车站8与车站29的运行时间为：1199
205	车站8与车站30的运行时间为：1392
206	车站9与车站10的运行时间为：102
207	车站9与车站11的运行时间为：103
208	车站9与车站12的运行时间为：247
209	车站9与车站13的运行时间为：249
210	车站9与车站14的运行时间为：378
211	车站9与车站15的运行时间为：380
212	车站9与车站16的运行时间为：542
213	车站9与车站17的运行时间为：544
214	车站9与车站18的运行时间为：647
215	车站9与车站19的运行时间为：648
216	车站9与车站20的运行时间为：744
217	车站9与车站21的运行时间为：745
218	车站9与车站22的运行时间为：884
219	车站9与车站23的运行时间为：886
220	车站9与车站24的运行时间为：1019
221	车站9与车站25的运行时间为：1021
222	车站9与车站26的运行时间为：1103
223	车站9与车站27的运行时间为：1103
224	车站9与车站28的运行时间为：1197
225	车站9与车站29的运行时间为：1199
226	车站9与车站30的运行时间为：1392
227	车站10与车站11的运行时间为：1
228	车站10与车站12的运行时间为：145
229	车站10与车站13的运行时间为：147
230	车站10与车站14的运行时间为：276
231	车站10与车站15的运行时间为：278
232	车站10与车站16的运行时间为：440

233	车站10与车站17的运行时间为：442
234	车站10与车站18的运行时间为：545
235	车站10与车站19的运行时间为：546
236	车站10与车站20的运行时间为：642
237	车站10与车站21的运行时间为：643
238	车站10与车站22的运行时间为：782
239	车站10与车站23的运行时间为：784
240	车站10与车站24的运行时间为：917
241	车站10与车站25的运行时间为：919
242	车站10与车站26的运行时间为：1001
243	车站10与车站27的运行时间为：1001
244	车站10与车站28的运行时间为：1095
245	车站10与车站29的运行时间为：1097
246	车站10与车站30的运行时间为：1290
247	车站11与车站12的运行时间为：144
248	车站11与车站13的运行时间为：146
249	车站11与车站14的运行时间为：275
250	车站11与车站15的运行时间为：276
251	车站11与车站16的运行时间为：438
252	车站11与车站17的运行时间为：441
253	车站11与车站18的运行时间为：544
254	车站11与车站19的运行时间为：545
255	车站11与车站20的运行时间为：641
256	车站11与车站21的运行时间为：642
257	车站11与车站22的运行时间为：781
258	车站11与车站23的运行时间为：783
259	车站11与车站24的运行时间为：916
260	车站11与车站25的运行时间为：917
261	车站11与车站26的运行时间为：999
262	车站11与车站27的运行时间为：1000
263	车站11与车站28的运行时间为：1094
264	车站11与车站29的运行时间为：1095
265	车站11与车站30的运行时间为：1288
266	车站12与车站13的运行时间为：2
267	车站12与车站14的运行时间为：131
268	车站12与车站15的运行时间为：132
269	车站12与车站16的运行时间为：294
270	车站12与车站17的运行时间为：297
271	车站12与车站18的运行时间为：400
272	车站12与车站19的运行时间为：401
273	车站12与车站20的运行时间为：497
274	车站12与车站21的运行时间为：498
275	车站12与车站22的运行时间为：637
276	车站12与车站23的运行时间为：639
277	车站12与车站24的运行时间为：772
278	车站12与车站25的运行时间为：773

279	车站12与车站26的运行时间为：855
280	车站12与车站27的运行时间为：856
281	车站12与车站28的运行时间为：950
282	车站12与车站29的运行时间为：951
283	车站12与车站30的运行时间为：1144
284	车站13与车站14的运行时间为：129
285	车站13与车站15的运行时间为：131
286	车站13与车站16的运行时间为：293
287	车站13与车站17的运行时间为：295
288	车站13与车站18的运行时间为：398
289	车站13与车站19的运行时间为：399
290	车站13与车站20的运行时间为：495
291	车站13与车站21的运行时间为：496
292	车站13与车站22的运行时间为：635
293	车站13与车站23的运行时间为：637
294	车站13与车站24的运行时间为：770
295	车站13与车站25的运行时间为：771
296	车站13与车站26的运行时间为：853
297	车站13与车站27的运行时间为：854
298	车站13与车站28的运行时间为：948
299	车站13与车站29的运行时间为：949
300	车站13与车站30的运行时间为：1142
301	车站14与车站15的运行时间为：2
302	车站14与车站16的运行时间为：164
303	车站14与车站17的运行时间为：166
304	车站14与车站18的运行时间为：269
305	车站14与车站19的运行时间为：270
306	车站14与车站20的运行时间为：366
307	车站14与车站21的运行时间为：367
308	车站14与车站22的运行时间为：506
309	车站14与车站23的运行时间为：508
310	车站14与车站24的运行时间为：641
311	车站14与车站25的运行时间为：642
312	车站14与车站26的运行时间为：724
313	车站14与车站27的运行时间为：725
314	车站14与车站28的运行时间为：819
315	车站14与车站29的运行时间为：820
316	车站14与车站30的运行时间为：1013
317	车站15与车站16的运行时间为：162
318	车站15与车站17的运行时间为：164
319	车站15与车站18的运行时间为：267
320	车站15与车站19的运行时间为：269
321	车站15与车站20的运行时间为：365
322	车站15与车站21的运行时间为：366
323	车站15与车站22的运行时间为：505
324	车站15与车站23的运行时间为：506

325	车站15与车站24的运行时间为：639
326	车站15与车站25的运行时间为：641
327	车站15与车站26的运行时间为：723
328	车站15与车站27的运行时间为：724
329	车站15与车站28的运行时间为：818
330	车站15与车站29的运行时间为：819
331	车站15与车站30的运行时间为：1012
332	车站16与车站17的运行时间为：2
333	车站16与车站18的运行时间为：105
334	车站16与车站19的运行时间为：107
335	车站16与车站20的运行时间为：203
336	车站16与车站21的运行时间为：204
337	车站16与车站22的运行时间为：343
338	车站16与车站23的运行时间为：344
339	车站16与车站24的运行时间为：477
340	车站16与车站25的运行时间为：479
341	车站16与车站26的运行时间为：561
342	车站16与车站27的运行时间为：562
343	车站16与车站28的运行时间为：656
344	车站16与车站29的运行时间为：657
345	车站16与车站30的运行时间为：850
346	车站17与车站18的运行时间为：103
347	车站17与车站19的运行时间为：104
348	车站17与车站20的运行时间为：200
349	车站17与车站21的运行时间为：201
350	车站17与车站22的运行时间为：340
351	车站17与车站23的运行时间为：342
352	车站17与车站24的运行时间为：475
353	车站17与车站25的运行时间为：477
354	车站17与车站26的运行时间为：559
355	车站17与车站27的运行时间为：559
356	车站17与车站28的运行时间为：653
357	车站17与车站29的运行时间为：655
358	车站17与车站30的运行时间为：848
359	车站18与车站19的运行时间为：1
360	车站18与车站20的运行时间为：97
361	车站18与车站21的运行时间为：98
362	车站18与车站22的运行时间为：237
363	车站18与车站23的运行时间为：239
364	车站18与车站24的运行时间为：372
365	车站18与车站25的运行时间为：374
366	车站18与车站26的运行时间为：456
367	车站18与车站27的运行时间为：456
368	车站18与车站28的运行时间为：550
369	车站18与车站29的运行时间为：552
370	车站18与车站30的运行时间为：745

371	车站19与车站20的运行时间为：96
372	车站19与车站21的运行时间为：97
373	车站19与车站22的运行时间为：236
374	车站19与车站23的运行时间为：238
375	车站19与车站24的运行时间为：371
376	车站19与车站25的运行时间为：372
377	车站19与车站26的运行时间为：454
378	车站19与车站27的运行时间为：455
379	车站19与车站28的运行时间为：549
380	车站19与车站29的运行时间为：550
381	车站19与车站30的运行时间为：743
382	车站20与车站21的运行时间为：1
383	车站20与车站22的运行时间为：140
384	车站20与车站23的运行时间为：142
385	车站20与车站24的运行时间为：275
386	车站20与车站25的运行时间为：276
387	车站20与车站26的运行时间为：358
388	车站20与车站27的运行时间为：359
389	车站20与车站28的运行时间为：453
390	车站20与车站29的运行时间为：454
391	车站20与车站30的运行时间为：647
392	车站21与车站22的运行时间为：139
393	车站21与车站23的运行时间为：141
394	车站21与车站24的运行时间为：274
395	车站21与车站25的运行时间为：275
396	车站21与车站26的运行时间为：357
397	车站21与车站27的运行时间为：358
398	车站21与车站28的运行时间为：452
399	车站21与车站29的运行时间为：453
400	车站21与车站30的运行时间为：646
401	车站22与车站23的运行时间为：2
402	车站22与车站24的运行时间为：135
403	车站22与车站25的运行时间为：136
404	车站22与车站26的运行时间为：218
405	车站22与车站27的运行时间为：219
406	车站22与车站28的运行时间为：313
407	车站22与车站29的运行时间为：314
408	车站22与车站30的运行时间为：507
409	车站23与车站24的运行时间为：133
410	车站23与车站25的运行时间为：135
411	车站23与车站26的运行时间为：217
412	车站23与车站27的运行时间为：218
413	车站23与车站28的运行时间为：312
414	车站23与车站29的运行时间为：313
415	车站23与车站30的运行时间为：506
416	车站24与车站25的运行时间为：2

```

417  车站24与车站26的运行时间为：84
418  车站24与车站27的运行时间为：85
419  车站24与车站28的运行时间为：179
420  车站24与车站29的运行时间为：180
421  车站24与车站30的运行时间为：373
422  车站25与车站26的运行时间为：82
423  车站25与车站27的运行时间为：83
424  车站25与车站28的运行时间为：177
425  车站25与车站29的运行时间为：178
426  车站25与车站30的运行时间为：371
427  车站26与车站27的运行时间为：1
428  车站26与车站28的运行时间为：95
429  车站26与车站29的运行时间为：96
430  车站26与车站30的运行时间为：289
431  车站27与车站28的运行时间为：94
432  车站27与车站29的运行时间为：95
433  车站27与车站30的运行时间为：288
434  车站28与车站29的运行时间为：1
435  车站28与车站30的运行时间为：194
436  车站29与车站30的运行时间为：193

```

附录 3

```

1  //check 判断函数 C++代码
2  pair<int,int> check(int cnt,int len){
3
4      int cur=0;//将客流量以列车数量为单位，得到在列车数目为 cnt，区间长度为 len 下所承载的客流量的值
5
6      unordered_map<int,int>vis;//已覆盖小交路列车个数for(int i = 1 ; i < 30 ; i++ ){
7          if(is_not[i]&&((i+len<30&&is_not[i+len])||i+len>=30))continue;//不能作为小交路列车的起始终止点将被除
            去
8
9          if(cnt==0)return {0,cur};//原计划的车辆列车数为 0 时，返回当前所承载的用户量（用列车数量为单位）
10
11         if(vis[i]<small_cnt[i]){//小于要求最小小交路列车要求数量，则在此处增加小交路
            列车数量
12
13
14             cnt--;//第一次二分得出的个数，此处增加小交路列车数量，则原计划小交路列车数量减一（逆向思维）
15
16             for(int j=i;j<=i+len&&j<30;j++){ cur+=max(small_cnt[j]-vis[j],0);//计算承载客流量
17                 vis[j]=max(small_cnt[j],vis[j]);//更新站点的车辆数目
18
19             }
20         }
21     }

```

附录 4

```
1 //数据预处理C++程序代码
2 #include<bits/stdc++.h>
3 using
4 namespace
5 std;
6 const
7 int
8 big_maxx = 2 * 1860;//大交路载客量
9 const
10 int
11 big_capacity = 1860;//列车的最大载客量
12 int
13 nums[50];//断面客流量
14 int
15 st[50];//更新站点客流量
16 unordered_map<int, bool>is_not;//没有折返能力的车站
17 int
18 is_not_cnt;//没有折返能力的车站总数目
19 int
20 small_cnt[50];//每一站点所需小交路列车的数量
21 int
22 _num, _len;//小交路列车的数量, 区间长度
23 int
24 must_cnt;//以列车数目为单位的总客流量
25 pair<int, int>
26 check(int
27 cnt, int
28 len) {
29     int
30     cur = 0;//将客流量以列车数量为单位, 得到在列车数目为cnt, 区间长度为
31     len下所承载的客流量的值
32     unordered_map<int, int>vis;//已覆盖小交路列车个数
33     for (int
34         i
35         =
36         1
37         ;
38         i
39         <
40         30
41         ;
42         i++)
43     {
44         if (is_not[i] && ((i + len < 30 && is_not[i + len]) || i + len >= 30))continue; /
```

```

45     /不能作为小交路列车的起始终止点将被除去
46     if (cnt == 0)return
47     { 0,cur };//原计划的车辆列数为0时，返回当前所承载的
48     用户量（用列车数量为单位）
49     if (vis[i] < small_cnt[i]) {//小于要求最小小交路列车要求数量，则在此处增
50     加小交路列车数量
51     cnt--;//第一次二分得出的个数，此处增加小交路列车数量，则原计划小交路
52     列车数量减一（逆向思维）
53     for (int
54     j = i; j <= i + len && j < 30; j++) {
55         cur += max(small_cnt[j] - vis[j], 0);//计算承载客流量
56         vis[j] = max(small_cnt[j], vis[j]);//更新站点的车辆数目
57     }
58     }
59     }
60     return
61     { cnt,cur };
62 }
63 void
64 print(int
65 cnt, int
66 len) {
67     map<int, int>vis;//已覆盖小交路列车个数
68     vector<int>vec;//列车开行方案
69     for (int
70     i = 1; i < 30; i++) {
71         if (is_not[i] && ((i + len < 30 && is_not[i + len]) || i + len >= 30))continue; /
72         /不能作为小交路列车的起始终止点将被除去
73         if (vis[i] < small_cnt[i]) {//小于要求最小小交路列车要求数量，则在此处增
74         加小交路列车数量
75         vec.push_back(i);//记录车站站点
76         for (int
77         j = i; j <= i + len && j < 30; j++) {
78             vis[j] = max(small_cnt[j], vis[j]);//更新站点的车辆数目
79         }
80     }
81     }
82     for (int
83     i = 0; i < vec.size(); i++) {
84         for (int
85         j = vec[i]; j <= vec[i] + len && j < 30; j++) {
86             vis[vec[i]] = max(vis[j], vis[vec[i]]);//更新每一开行区间中列车的数
87             量
88         }
89     }
90     cout << vec.size() << "个小交路公交站和区间长度为" << len + 1 << endl;

```

```

91     for (auto
92     it : vec) {
93         cout << it << "站: " << vis[it] << endl;
94     }
95     puts("");
96     return
97     ;
98     int
99     binary_two(int
100     cnt) {
101         int
102         l = 2, r = 4; //限定开行区间长度范围
103         while (l < r) {
104             int
105             mid = l + r >> 1;
106             if (check(cnt, mid).first >= 0) {
107                 l = mid + 1;
108             }
109             else
110                 r = mid;
111         }
112         _len = l;
113         return
114         check(cnt, l).second;
115     }
116     void
117     binary_one() {
118         int
119         l = 3, r = 24; //限定列车数量范围
120         while (l < r) {
121             int
122             mid = l + r >> 1;
123             if (binary_two(mid) >= must_cnt) { //满足
124                 r = mid;
125             }
126             else
127                 l = mid + 1;
128         }
129         _num = l;
130     }
131     int
132     main() {
133         for (int
134         i = 1; i < 30; i++) {
135             scanf("%d", &nums[i]);
136         }

```

```

137         cin >> is_not_cnt;
138         for (int
139             i = 1; i <= is_not_cnt; i++) {
140             int
141                 node;
142             cin >> node;
143             is_not[node] = true; //将不能作为小交路列车的起点终点坐标标记
144         }
145         for (int
146             i = 1; i < 30; i++) {
147             st[i] = max(nums[i] - big_maxx, 0);
148         }
149         for (int
150             i = 1; i < 30; i++) {
151             small_cnt[i] = (st[i] + big_capacity - 1) / big_capacity; //每一站点小交
152             路列车向上取整时最小数量
153             must_cnt += small_cnt[i]; //将总客流量换算为列车数目为单位
154         }
155         binary_one();
156         print(_num, _len);
157         return 0;
158     }

```

附录 5

```

1 %MATLAB问题一
2 clc
3 clear
4 [~, ~, G] = xlsread('附件1: 车站数据.xlsx');
5 G = G(2:end, 2); % 去除车站名称列
6 T = xlsread('附件2: 区间运行时间.xlsx');
7 C = xlsread('附件3: OD客流数据.xlsx');
8 D = xlsread('附件4: 断面客流数据.xlsx');
9 min_max_headway = [120, 360];
10 min_spacing = 108;
11 min_max_stop_time = [20, 120];
12 train_capacity = 1860;
13 min_max_station_num = [3, 24];
14 avg_boarding_time = 0.04;
15
16 n = size(T, 1); % 大交路数量
17 m = size(T, 2); % 小交路数量
18
19 f = [C(:); ones(m, 1)]; % 目标函数系数
20
21 Aeq = zeros(m, n*m + m); % 相等约束系数矩阵

```

```

22     beq = D(:); % 相等约束右侧常数
23
24     lb = zeros(n*m+m,1); % 变量下限
25
26     % 不等约束系数矩阵
27     A = zeros(m*n,n*m+m);
28     b = zeros(m*n,1);
29
30     % 相等约束
31     for j = 1:m
32         Aeq(j, (j-1)*n+1:j*n) = 1;
33         Aeq(j, n*m+j) = 1;
34     end
35
36     % 不等约束
37     for j = 1:m
38         for i = 1:n
39             A((j-1)*n+i, (i-1)*m+j) = 1;
40             A((j-1)*n+i, n*m+j) = T(i,j);
41         end
42         A((j-1)*n+1:j*n, n*m+j) = -1;
43         b((j-1)*n+1:j*n) = -min_max_headway(1) * D(j) / 3600; % 转换为小时
44     end
45
46     % 非负约束
47     for i = 1:n*m+m
48         lb(i) = 0;
49     end
50
51     % 上下界约束
52     ub = Inf(n*m+m,1);
53     for j = 1:m
54         ub(n*m+j) = min_max_headway(2);
55     end
56     for i = 1:n
57         for j = 1:m
58             ub((i-1)*m+j) = min(D(j), train_capacity) / train_capacity;
59         end
60     end
61
62     for j = 1:m
63         station_num = 0; % 初始化车站数量为0
64         for i = 1:n
65             if G{i,j} == 1 % 统计车站数量
66                 station_num = station_num + 1;
67             end

```

```

68     end
69     A((j-1)*n+1:j*n, (j-1)*n+1:j*n) = -station_num*avg_boarding_time*ones(n);
70     b((j-1)*n+1:j*n) = -min_max_stop_time(1)*station_num*ones(n);
71     end
72
73     [x, ~, ~, ~] = linprog(f, A, b, Aeq, beq, lb, ub);
74
75     % 提取解向量中的决策变量
76     X = reshape(x(1:nm), n, m);
77     Y = x(nm+1:n*m+m);
78
79     % 打印结果
80     fprintf('Objective Value: %f\n', fval);
81     fprintf('X:\n');
82     disp(X);
83     fprintf('Y:\n');
84     disp(Y);

```

附录 6

```

1     %MATLAB问题二
2     % 列名重命名过
3     % 车站名称 (station_name) 是否可作为交路起点/终点(if_origin_destination)
4     % 区间 (interval) 区间运行时间/s (running_time) 站间距/km (station_spacing)
5     % 车站1-30 改为station_1到station_30
6
7     % 读取车站数据
8     station_data = readtable('附件1: 车站数据.xlsx');
9     is_start_end = strcmp(station_data.if_origin_destination, '是'); % 判断是否为起点或终点
10    start_end_stations = station_data(is_start_end, :); % 起点和终点车站
11    num_start_end = size(start_end_stations, 1); % 起点和终点车站数目
12
13    % 读取区间运行时间
14    interval_data = readtable('附件2: 区间运行时间.xlsx');
15    num_interval = size(interval_data, 1); % 区间数目
16
17    % 读取OD客流数据
18    od_data = readtable('附件3: OD客流数据.xlsx');
19    num_station = size(od_data, 1); % 车站数目
20    od_matrix = table2array(od_data(:, 2:end)); % 提取OD客流矩阵
21
22    % 计算交路运行时间和运行里程
23    route_time = zeros(num_start_end, num_start_end); % 交路运行时间矩阵
24    route_distance = zeros(num_start_end, num_start_end); % 交路运行里程矩阵
25    for i = 1:num_start_end
26    for j = 1:num_start_end

```



```

27     if i == j
28         continue;
29     end
30     start_station = start_end_stations(i, :).station_name{1};
31     end_station = start_end_stations(j, :).station_name{1};
32     interval_index = strcmp(interval_data.interval, [start_station '->' end_station]);
33     route_time(i, j) = interval_data.running_time(interval_index);
34     route_distance(i, j) = interval_data.station_spacing(interval_index) * 1000;
35     end
36     end
37
38     % 计算交路客流量
39     route_flow = zeros(num_start_end, num_start_end); % 交路客流量矩阵
40     for i = 1:num_start_end
41         for j = 1:num_start_end
42             if i == j
43                 continue;
44             end
45             start_station = start_end_stations(i, :).station_name{1};
46             end_station = start_end_stations(j, :).station_name{1};
47             start_index = find(strcmp(od_data.station_name, start_station));
48             end_index = find(strcmp(od_data.station_name, end_station));
49             route_flow(i, j) = sum(sum(od_matrix(start_index:end_index-1, start_index+1:end_index)));
50         end
51     end
52
53     % 计算交路运行速度和加权平均速度
54     route_speed = route_distance ./ route_time; % 交路运行速度矩阵
55     weighted_speed = sum(sum(route_flow .* route_speed)) / sum(sum(route_flow)); % 加权平均速度
56
57     % 输出结果
58     disp("交路运行时间（单位：秒）：")
59     disp(route_time)
60     disp("交路运行里程（单位：米）：")
61     disp(route_distance)
62     disp("交路客流量（单位：人次）：")
63     disp(route_flow)
64     disp("交路运行速度（单位：米/秒）：")
65     disp(route_speed)
66     disp("加权平均速度（单位：米/秒）：" + weighted_speed)

```