湖南省人民政府学位委员会办公室
湖南省教育厅学位管理与研究生教育处

2017 湖南省研究生数学建模竞赛参赛承诺书

我们仔细阅读了湖南省研究生数学建模竞赛的竞赛规则.

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的，如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们将受到严肃处理。

我们授权湖南省研究生数学建模竞赛组委会，可将我们的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等）。

我们参赛选择的题号是（从组委会提供的试题中选择一项填写）：A 题

我们的队号为（填写完整的队号）：201718001005

所属学校（请填写完整的全名）：国防科学技术大学

参赛队员（打印并签名）：

1. 纪后继
2. 蒋纬中
3. 肖昌成

指导教师或指导教师组负责人(打印并签名)：王丹

日期：　　年　　月　　日

评阅编号（由组委会评阅前进行编号）：

2017 湖南省研究生数学建模竞赛

编 号 专 用 页

评阅编号（由组委会评阅前进行编号）：

评阅记录（可供评阅时使用）：

湖南省第三届研究生数学建模竞赛

题 目 出租车合乘业务系统设计

摘 要：

本文提出了“有座即可合乘”的合乘模式，在不超过出租车承载量的情况下，快速准确地为乘客派车。将合乘调度问题转化为带约束路径规划问题，基于遗传算法提出优化方案，通过选取距离乘客最近的五辆车作为预优化选择，降低了问题的复杂度，提出一种嵌套型自然数编码方式，以总距离的倒数为优化函数，对乘客等待时间及出租车使用数目均有优化，并对不满足约束的个体施加惩罚因子，加快遗传算法的收敛，保证了结果的最优性。

针对合乘模式，提出“分人分段”的收费模式，根据合乘人数，对合乘部分里程按比例收费。采用比例因子可调的方式，实现因地制宜。实现了乘客、出租车“多赢”，有利于缓解城市拥堵。

关键字：遗传算法 预优化 嵌套型自然数编码 惩罚因子 分人分段

1. 问题重述

1.1 问题背景

出租车合乘业务是指路线相同或相近的两位或多位乘客共同乘坐同一辆出租车出行，系统根据合乘人数、乘车时间、实际路线等因素，分别计算出每位乘客的车费（通常低于各自独乘时的车费）。司机收入则为所有乘客支付的车费总和。该业务可以在不增加运营车辆总数的情况下提高运力，有助于缓解打车难，而且能够降低乘客出行成本，同时提高司机收入。因此，相当一部分乘客、司机愿意接受该业务，特别是在打车的高峰时段。

1.2 问题提出

某出租车公司拟开展合乘业务。通过调研发现，其他城市或公司的合乘业务主要有两种模式。其中一种模式是相同起点模式，这种模式要求合乘乘客必须在同一地点上车，顺路去相同或不同的目的地。这种模式的计费规则简单，但缺点是合乘的条件较严格，合乘比例偏低。另一种模式是“一口价”模式，这种模式是基于历史数据预估车费。优点是合成条件低，方案灵活，缺点是这种模式可能出现绕行，引起乘客不满;若全程无法构成合乘，影响司机收入，引起司机不满。

因此，需要设计新的合乘模式，充分调动乘客、司机等各方参与合乘的积极性。

1.3 待解决问题

为了充分调动乘客、出租车司机参与合乘的积极性，需要完成以下任务：

1、设计高效的合乘模式及其相应的算法，使乘客等待时间尽量短，所需出租车数量尽量少；
2、设计与合乘方案相应的合理的车费计算方法；
3、根据题目附件中打车需求数据以及空驰出租车位置信息，按照设计方案给出具体结果。

2. 问题分析

2.1 对问题 1 的分析

问题 1 要求设计高效的合乘方案以及算法，使得乘客等待的时间尽量短，所需出租车数量尽可能少。这个问题本质上是合乘出租车车辆的的路径优化问题，这是以乘客为匹配目标，对确定的乘客拼车组合进行优化。车辆路径优化问题属于 NP 完全难题，即：问题规模较大时，很难得到全局最优解或满意解，而且算

法的计算时间随着问题规模的增大呈指数增长。对这类问题采用如遗传算法、模拟退火算法等启发式算法来解决。

2.2 对问题 2 的分析

问题 2 要求设计与合乘方案相应的车费计算方法。车费的计算方法应本着“多赢”的原则，即所设计的方案要能让乘客、出租车司机都有所收益，这样才有利于新的合成方案的推广。

2.3 对问题 3 的分析

问题 3 要求按照所设计的合乘方案以及计费规则，利用附件中所给数据给出具体的计算结果并按照指定的格式输出结果。需要注意的是与费用相关的计算结果需要自己设计输出文件格式。

3. 模型假设

1、假设不需要考虑对公司现有调度平台和手机打车软件的改造难度。
2、出租车在当前合乘情况下，需将所有乘客送达目的地后，才会进行下一次合乘安排。
3、合乘的承载量应该小于出租车的承载量，最大载客容量为 3。
4、合乘过程中每位乘客享受到的服务没有差异性。
5、出租车在行驶时均为匀速，不考虑转弯、红绿灯、堵车的情况。
6、假设两点之间无障碍物阻隔，使用曼哈顿距离作为两点距离。由于车道由边长为 0.5km 的正方形网格构成，两点被方格挡住时，在最坏的情况下两点路程比两点曼哈顿距离大 0.5km，可忽略不予考虑。

4. 符号说明

符号	符号说明
$pickup_i\_x$	第 i 个乘客当前 x 坐标
$pickup_i\_y$	第 i 个乘客当前 y 坐标
$dropoff_i\_x$	第 i 个乘客目的地的 x 坐标
$dropoff_i\_x$	第 i 个乘客目的地的 y 坐标
tax_i_x	第 i 辆出租车的 x 坐标
tax_i_y	第 i 辆出租车的 y 坐标
Dij	第 i 个乘客和第 j 辆车之间的距离
D	合乘方案的总距离 D
Pb_i	个体的适应度占种群中所有个体适应度总和的比例
A_i%	出租车上有 i 个合乘乘客时，按单乘收费的 A_i%收取
P_s	单人乘车费用
C_b	出租车起步价
C_v	乘车里程超过起步距离后的累进费率

d	出租车的行驶里程
R_n	多人合乘时每位乘客应付费用
i	合乘状态过程
d_{0i}	在起步公里范围内（3 千米）的行驶距离
d_i	在 i 状态下，超出起步范围的行驶距离
P_I	第 i 名乘客上车位置
D_i	第 i 名乘客下车位置

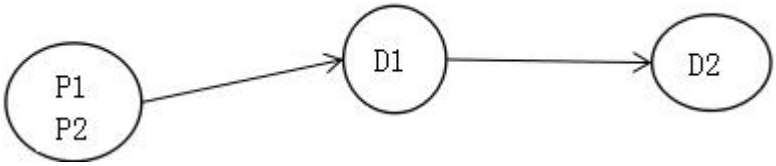
5. 问题 1 的模型建立与求解

5.1 出租车合乘出行的匹配算法

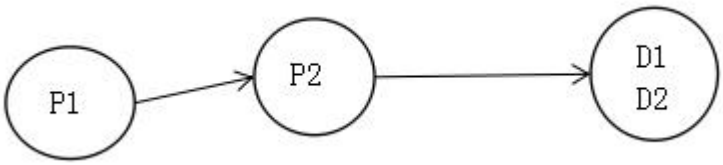
5.1.1 合乘路线分类

按照出发地及目的地的情况，以 2 人合乘为例，居民合乘出租车出行的类型主要有以下几种[1]，见图 5.1.

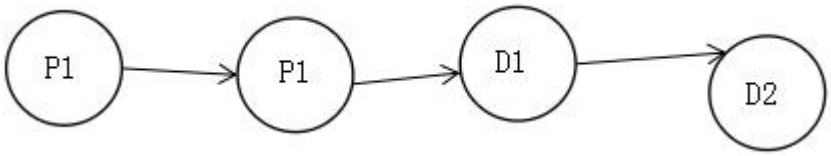
（a）乘客具有相同的起点、终点



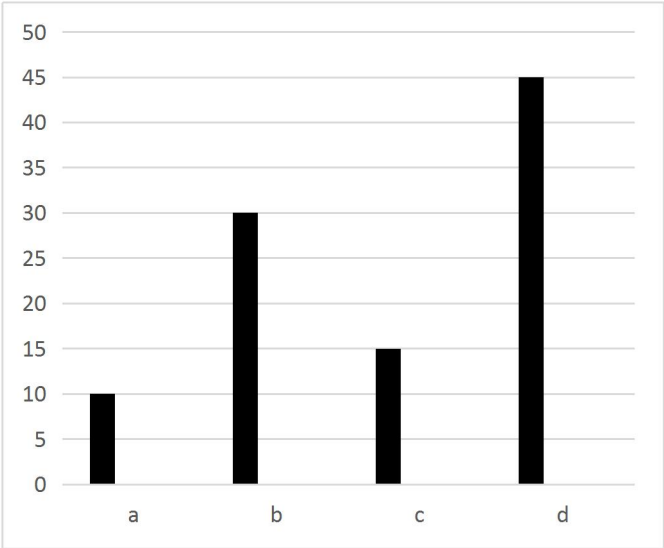
（b）乘客的起点相同、终点不同



（c）乘客的起点不同，终点相同



（d）乘客起点、终点都不同



（e） 合成路线类型比例

图 5.1 出租车合成路线类型及比例

通过分析路线类型及其比例，发现具有不同起点、终点的乘客所占比例最大，故为了更好地服务乘客应该不只是考虑具有“同一起点”和“相同终点”的路线匹配问题。

5.1.2 合乘方案

为了实现乘客等待的时间尽量短并且所用出租车尽量少，本文制定的合乘方案不要求乘客具有相同的起点或终点，以乘客的等待时间和所需出租车数量为优化目标。

乘客通过手机软件提交打车请求（起始位置信息等），系统根据当前所有乘客的位置信息以及出租车位置信息，在不超过出租车承载量的情况下，遵循乘客等待时间最短的原则（即出租车到乘客的距离最近），为乘客派车。将这种方案称为“有座即可合乘”。

5.2 合乘出行匹配算法的实现

合乘出行匹配问题属于 NP 问题，本文选择遗传算法来实现。遗传算法是一种“物竞天择，适者生存”的随机、自适应的优化算法。在对乘客和出租车位置

分布进行预处理后，采用遗传算法来实现[3]。

本文给出了具体的编码方法、适应度函数、选择策略、交叉变异策略。

算法流程图如图 5.2 所示。

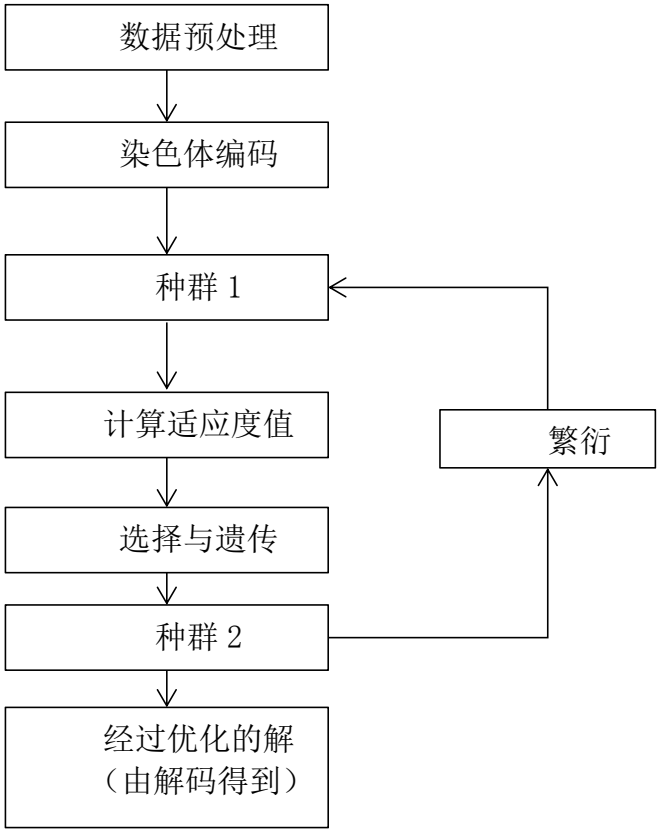


图 5.2 遗传算法总体框图

5.2.1 乘客位置和出租车位置信息预处理

由于附件中是以坐标的形式给出乘客和出租车的位置信息，需要进行预处理以便于染色体的编码。乘客和每辆出租车之间的距离计算公式见（5.1）式。

$$D_{ij} = |tax_{i\_}x - pick_i\_x| + |tax_{i\_}y - pick_i\_y| \qquad (5.1)$$

每组被选出的五辆车按照距离由小到大进行排序编号，分别为 1、2、3、4、5。所有被选中的出租车组成车库矩阵 carf，如 carf(9,3)表示 P9 乘客备选五辆车中的第三辆。

对于 1001 位乘客、667 辆出租车，每位乘客对出租车只有五个选择方案，将问题的复杂度由 667^{1001} 降为 5^{1001}，且需保证所有出租车上乘客数不能大于 3，那么问题变为带约束的规划问题。

以下采用遗传算法对该问题进行求解，在满足约束条件的情况下，最差的解也必须为从乘客周围五辆车选择接送方案，保证了结果的优越性，降低了问题的复杂度，加速了遗传算法求解速度。

5.2.2 嵌套型自然数编码

本文采用的编码方法在自然数编码方法(或称整数编码方式)上进行了改进。自然数编码方法用一个向量（$s_1, s_2, \ldots, s_N$）来表示染色体，其中基因 si 为[0,N]之间的一个互不重复的自然数。在数据预处理时选出了距离每个乘客最近的五辆车，编号为 1、2、3、4、5，用基因的位数 i 代表第 i 个乘客。这种编码可以把染色体表示为:

（$s_1, s_2, \ldots, s_i, \ldots$），　其中 $s_1, s_2, \ldots, s_i \in \{1,2,3,4,5\}$

以染色体（1,3,5,4）为例，它表示接乘客 P1 的车是距离其最近的五辆车中编号为 1 的车 carf(1,1)，接乘客 2 的车是距离其最近的五辆车中编号为 3 的车 carff(2,3)，接乘客 3 的车是距离其最近的五辆车中编号为 5 的车 carf(3,5)，接乘客 4 的车是距离其最近的五辆车中编号为 4 的车 carf(4,4)。

这种编码方法最大化地利用了染色向量，将乘客和出租车联系了起来,优点是编码简单。

5.2.3 适应度函数

适应度函数是用来评价种群中个体好坏的，这是进化的驱动力。设某个个体对应的合乘方案的总距离为 D，这是目标函数，本文将该个体的适应度函数 F 表示为目标函数值的倒数，即：

$$F = 1/D \tag{5.2}$$

F 的值越大就表示对应个体的性能越好，所对应的解越好，即乘客等待时间越短。

本题存在每辆车最多可乘坐三人的约束，采用在适应度函数中增加惩罚因子的方法，使不满足约束的个体被遗传算法淘汰。

5.2.4 选择策略

选择策略是用来确定如何从父代群体中按照某种方法选取个体遗传到下一代群体中。本文选用轮盘赌策略，这种策略以个体的适应度占种群中所有个体适应度总和的比例作为选择的概率，即对于 n 个个体，若 F 为适应度函数，个体 x_i 的选择概率为

$$Pb_i = \frac{F(x_i)}{\sum_{j=1}^{n} F(x_j)} \tag{5.3}$$

5.2.5 交叉策略

交叉策略是指两个相互配对的染色体按某种方式相互交换其部分基因，从而形成新的个体。交叉操作是在一定概率下发生的。本文采用单点定长交叉策略，即在个体串中随机选定一个交叉点，将父代个体在该点以后的十位子码进行互换，生成两个新个体。

5.2.6 变异策略

变异策略是指个体中某一个或某几个基因以一定的概率随机发生变动，变异操作是产生新个体的辅助方法，决定了遗传算法的局部搜索能力。本文采用基因位变异，即以某一概率独立地改变个体中的一个基因的值。

5.3 适应度函数模型

本文将该个体的适应度 F 表示为出租车运行距离总和的倒数。个体编码对应出租车的接乘方案，如编码为{2,3,3,4,5,1,2…},carf 为预优化后的备选车库，

第一位乘客对应车号为 carf(1,2)，第三位乘客对应车号为 carf(3,3)，若车号 carf(1,2)与 carf(3,3)相同，即表示第一位乘客与第三位乘客合乘。然而在合乘方案确定后，出租车接人顺序及行走路线有多种可能，本文以出租车为对象建立基于穷举法寻找最佳线路的模型，使得一个编码对应一个最优的接送方案。

5.3.1 最优化行车距离

根据编码的特点，解码后得到每辆出租车所要接送的乘客{Px1、Px2...}，设第 i 辆出租车送完乘客行驶的距离为 ds_i,则所有出租车行驶的总距离为 $D=\sum D_i$ 。所要优化目标如式（5.4）所示。

$$\min D=\sum ds_i \qquad i=1,2,3..... \tag{5.4}$$

D_i 相互独立，故优化任务转变为： $\min\ ds_i \qquad i=1,2,3.....$

根据乘客上下车位置排列出租车到各点的顺序{$x_{i1}, x_{i2}, x_{i3}...$}，出租车的位置为 C_i，则

$$D_i=d(C_i,x_{1i})+d(x_{1i},x_{2i})+d(x_{2i},x_{3i})\cdots \quad i=1,2,3\cdots \tag{5.5}$$

其中 $d(p1,p2)$ 为两点的曼哈顿距离。

根据排列组合得出第 i 辆出租车所有乘车人上下车位置的排序，并剔除所有乘车人先下车后上车的可能。求解所有可行排序方案中出租车的行驶距离，通过比较选取最小值作为优化结果。

5.3.2 惩罚因子

对于不满足约束，即存在四人以上搭乘同一辆车 C_j 的情况，对其设定该车远大于正常值的行车距离作为惩罚，平均一个人的行驶距离约为 10km，设定惩罚因子为 D_J=1000. 即该车的行驶路程设为 1000km，可以保证不满足约束的个体适应度最差。

5.4 算法实现

算法利用 MATLAB 编写，具体程序见附录 1。

6. 问题 2 的模型建立与求解

6.1 问题研究背景

目前各大中城市从事出租车运营的主导车型主要有捷达、桑塔纳、夏利等，据测算，2012 年大连每台出租车全天（包括两班）平均行驶里程 400 公里，载客有效里程平均为 260 公里，空载里程 140 公里，日运营次数平均 41 次，其中高峰期运营次数 4 次，平峰期运营次数 37 次。日运营收入 703.15 元，日支出成本 438.94 元，日纯收入 264.21 元。单车日耗油量平均 38 升。据测算，出租车司机单班平均月收入为 4137.4 元[（日纯收入 264.21 元+日油补 11.62 元）×30 天÷2 人=4137.45 元。需要说明的是，出租汽车驾驶员每天的出车时间平均为 11 小时，每周出车时间平均 77 为小时，劳动时间比国家规定的每天 8 小时、每周 40 小时的工作时间多近一倍[4]。

另一方面，在全国 36 个大中城市对于空驶率的调查研究显示，除少数城市出租车空驶率低于 30%外，大多数城市空驶率均高于 30%，这使得出租车运营市场面临考验[5]。

因此，出租车费率的制定关系到我国城市交通建设和广大群众切身利益，现在 36 个大中城市对出租车价格都实行了政府定价，具体的计价办法和标准由城市政府价格主管部门制定，报本级人民政府审批，并且会按照规定召开听证会。

6.2 现行出租车计价方法

6.2.1 我国现行出租车计价办法

我国现行的出租车计价办法，总的来看包括起步价、累进运价，低速或等候收费、空驶费、夜间加班费等。起步价即规定一个起步里程和最低收费标准,未超过起步里程仍按最低标准计费；低速或等候收费,绝大多数城市都允许出租车额外收取等候或低速行驶费；空驶费为鼓励出租车开展市区与郊区之间运营,减少拒载,各地普遍规定出租车载客超过一定里程后,加收一定的空驶费,以补偿车辆返空时的成本支出；夜间附加费,不少城市都规定夜间出租车运价适当上浮。

6.2.2 出租车合乘计费的特殊性

合乘出租车的出行方式和单乘出租车的出行方式相比会有一些区别：

（1）复杂性

单乘出租车的计价方式简单，计费公式为 2.1（假设起步里数为 3 公里）：

总价=起步价 + 累运费率 ×（行驶距离—3）+ 其他费用　　（6.1）

出租车的合乘出行，涉及到的乘客数量为两人或三人，每个人的乘车距离、上车的次序均不同，所以在起步价、累乘运价的分担方法就会更复杂。

（2）公平性

出租车合乘是基于多个乘客的需求而组成的集合体，出租车司机与乘客、乘客与乘客之间的利益关系要在费用的支付上尽量体现出公平，实现共赢，这样才有助于推进合乘模式的推广。

6.3 基于行驶距离的出租车合乘计费模型

6.3.1 基于行驶距离的出租车合乘计费办法。

目前，已经有不少城市对出租车拼车计费做出了规定,都是在单乘基础上简单的比例运算，比如北京市规定,在发生合乘的情况下“合乘里程部分”由每位乘客按单乘情况下应付金额的 60%支付,重庆市以 70%支付。以这种思路进行模型化即为基于行驶距离的出租车拼车费率计算。

采用基于行驶距离的出租车拼车收费方式,出租车内部合乘的人数为分类对象,即以合乘车辆当前的行驶里程为基准,按照普通出租车的标准收费,在三种：单人、双人，三人的状态下对每个乘客进行独立的费用折减[6]。

计算思路如下：

（1）单人、双人、三人乘坐部分,合计收费总额分别按照普通收费标准的 $A_1\%$、$A_2\%$、$A_3\%$计算。

（2）费用计算公式：

①单人乘坐的计费公式：

$$P_s=\begin{cases}C_b\times A_1\% & d\le 3\\ [C_b+C_v\times(d-3)]\times A_1\% & d>3\end{cases} \qquad (6.2)$$

式中，Ps 为单人费用，Cb 为起步价，Cv 为累进费率，d 为行驶距离（单位：千米）。

②多人合乘的计费公式：

$$R_n=\begin{cases}\sum_{i=1}^{n}\frac{C_b}{3}\times d_{0i}\times A_i\%, \sum_{i=1}^{n}d_{0i}\le 3km\\ \sum_{i=1}^{n}(\frac{C_b}{3}\times d_{0i}+C_v\times d_i)\times A_i\%, \sum_{i=1}^{n}d_{0i}>3km\end{cases} \quad i=1,..m,\quad n=2or3 \qquad (6.3)$$

在（6.3）式中，n 代表合乘的人数，i 代表合乘的分段，R_n 为多人合乘每位乘客应付费用；d_{0i} 表示在第 i 段下，在起步公里范围内（3 千米）的行驶距离；d_i 表示在第 i 段，超出起步范围的行驶距离。

（3）比例因子 A 的确定，可参考现在实行的收费情况以及城市实际路况，因子的确定具体来说可参照出租车运营价格的制定，通过价格听证会的举办，协调各方面利益，确定参数。

6.3.2 计算实例

假设在长沙市合乘的收费标准为：起步价 8.00 元/3 公里，3 公里以上续程

单价为每公里 2.0 元，A_1%=90%、A_2%=70%、A_3%=50%。现在有一辆合乘出租车，第一名乘客上车行驶了 2 公里后，第二名乘客上车；继续行驶了 2.5 公里，第三名乘客上车；又行驶了 2 公里后，第二名乘客下车；再行驶了 1.5 公里后，第一名乘客下车；又行驶了 1 公里后，第三名乘客下车；

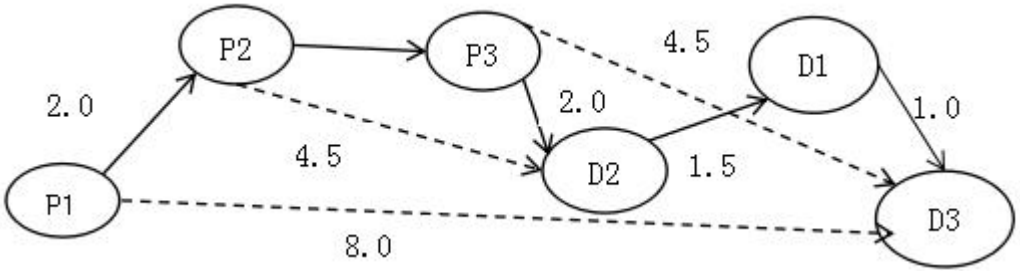


图 2.1 实例网络图及距离表示

注：图中虚线表示乘客单乘所行驶距离，单位（Km）

费用计算如下：
第一名乘客：R1=8÷3×2×90%+（8÷3×1+2×1.5×2×70%+2×2×50%=12.9 元
第二名乘客：R2=8÷3×2.5×70%+（8÷3×0.5+2×1.5）×50%=6.8 元
第三名乘客：R3=8÷3×2×50%+（8÷3×1+2×0.5）×70%+2×1.5×90%=8.0 元
合乘总费用：R=12.9+6.8+8.0=27.7 元

6.3.3 计算办法的经济效果分析

设计出租车合乘的目的是为了实现司机和乘客在经济上的“多赢”，只有实现了这个目标，出租车合乘才能更好地推广。

根据计算实例，将计算结果对比，见表 2.1，可以看出采用出租车合乘方案，在本例中，乘客所能够节省的费用平均可达 30%以上，而出租车司机也可以比单独服务多收益 50%以上（没有考虑多次起、停所造成的额外油耗），可见采用这种方案实现了多赢的目标。

表 6.1 三人合乘出租车与单乘出行的费用分析

方式 / 参与者	出租车拼车出行费用（单位：元）	单独乘出租车出行费用（单位：元）	收益率
乘客一	12.9	18	28.3%
乘客二	6.8	11	38.2%
乘客三	8.0	11	27.3%
出租车司机	27.7	18（合成总距离）	53.4%

7. 问题 3 的求解

7.1 问题 3 中要解决的问题

（1）根据 1 中的方案，执行算法，得到具体的合乘方案，并按指定的格式输出；
（2）计算每位乘客的乘车费用及每名出租车司机的收入；
（3）由程序运行结果对方案进行简要的分析。

7.2 问题 3 的解决思路

问题 3 的解决思路如图 7.1 所示。

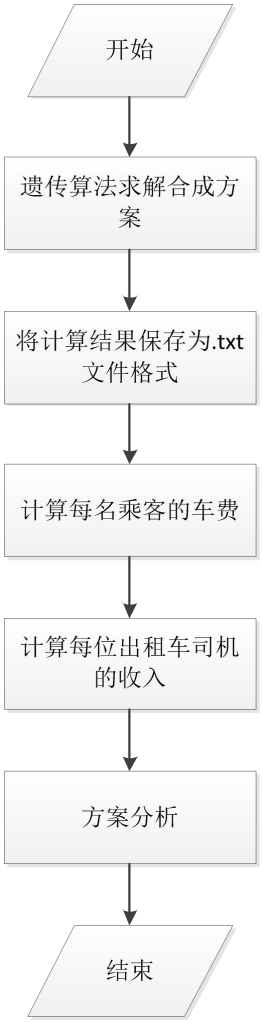


图 7.1 问题 3 处理流程图

编写的 MATLAB 程序见附件 output.m.

7.3 方案结果记录

（1）由 5 中的合乘方案，并利用遗传算法编写相应的处理程序。本方案中，为了节省程序运行时间与便于分析，从所给的乘客数据中选择前 100 位，从所给的出租车数据中选择前 60 辆。所选择的种群大小为 NP=50，最大进化代数 NG=1000，

交叉概率 P_c=0.75，变异概率 P_m=0.1 时，所求得的乘车的方案如下所示：

表 7.1 合乘方案一览表

出租车编号	乘客编号及状态						出租车编号	乘客编号及状态					
1	44	-44	8	-8	0	0	31	39	34	-34	5	-39	-5
2	59	-59	26	-26	0	0	32	12	-12	91	21	-21	-91
3	0	0	0	0	0	0	33	0	0	0	0	0	0
4	35	-35	0	0	0	0	34	0	0	0	0	0	0
5	41	-41	0	0	0	0	35	47	25	-25	96	-47	-96
6	73	11	-11	-73	31	-31	36	70	42	-70	-42	0	0
7	19	-19	17	-17	0	0	37	97	9	95	-95	-97	-9
8	36	-36	0	0	0	0	38	24	-24	0	0	0	0
9	61	-61	50	52	-50	-52	39	29	3	-3	-29	0	0
10	78	-78	60	-60	0	0	40	37	-37	0	0	0	0
11	80	10	-10	22	-80	-22	41	45	-45	0	0	0	0
12	0	0	0	0	0	0	42	2	-2	32	-32	0	0
13	14	-14	0	0	0	0	43	53	30	-30	82	-53	-82
14	92	93	-92	-93	0	0	44	0	0	0	0	0	0
15	57	-57	63	-63	0	0	45	94	48	-48	-94	0	0
16	84	-84	0	0	0	0	46	27	-27	0	0	0	0
17	40	-40	28	-28	0	0	47	0	0	0	0	0	0
18	0	0	0	0	0	0	48	90	33	-33	-90	0	0
19	67	-67	66	-66	0	0	49	1	38	-1	-38	0	0
20	23	77	-23	-77	0	0	50	46	-46	79	-79	81	-81
21	55	-55	0	0	0	0	51	65	58	-65	-58	0	0
22	74	7	4	-7	-74	-4	52	64	83	-64	-83	0	0
23	87	69	-87	-69	0	0	53	13	-13	68	-68	0	0
24	56	62	43	-56	-62	-43	54	75	54	-54	-75	98	-98
25	0	0	0	0	0	0	55	0	0	0	0	0	0
26	99	85	72	-99	-72	-85	56	51	18	-18	-51	0	0
27	6	15	86	-86	-6	-15	57	89	20	-20	-89	0	0
28	100	-100	0	0	0	0	58	0	0	0	0	0	0
29	76	-76	0	0	0	0	59	49	-49	0	0	0	0
30	71	-71	0	0	0	0	60	16	88	-88	-16	0	0

具体的方案见附录 2.

（2）利用（1）中得到的合乘方案，编写程序计算得到所选择的乘客的乘车花费及解中的各出租车的收入分别如下所示：

表 7.2 乘客乘车花费表

乘客	车费/元	乘客	车费/元	乘客	车费/元	乘客	车费/元
P0001	5.3816	P0026	8	P0051	2.289	P0076	9.0522
P0002	7.19472	P0027	3.4523	P0052	12.95722	P0077	66.6556
P0003	5.22624	P0028	12.31164	P0053	8.1178	P0078	8
P0004	6.23454	P0029	4.53421	P0054	3.5456	P0079	23.546
P0005	12.9897	P0030	5.56744	P0055	5.3264	P0080	4.58264
P0006	2.1292	P0031	8	P0056	1.2332	P0081	8.5654
P0007	8.23478	P0032	3.7232	P0057	8	P0082	9.6547
P0008	3.18904	P0033	25.83976	P0058	9.09936	P0083	10.0035
P0009	8.33616	P0034	0.84406	P0059	17.4595	P0084	4.6453
P0010	8	P0035	9.382133333	P0060	3.16952	P0085	51.45216
P0011	14.33782	P0036	8	P0061	4.5678	P0086	6.4547
P0012	8	P0037	7.01684	P0062	9.65682	P0087	6.6419
P0013	0.59472	P0038	4.51352	P0063	6.30666	P0088	7.8954
P0014	8	P0039	8.64548	P0064	4.096	P0089	5.65457
P0015	4.80152	P0040	5.43564	P0065	2.39442	P0090	9.00288
P0016	7.47414	P0041	7.3935	P0066	8	P0091	11.3256
P0017	2.7704	P0042	8.13204	P0067	8.45084	P0092	7.45478
P0018	10.58152	P0043	33.08688	P0068	14.44018	P0093	14.72256
P0019	1.8004	P0044	3.73828	P0069	8	P0094	27.07416
P0020	36	P0045	4.6575	P0070	2.345	P0095	0.82026
P0021	8	P0046	4.352533333	P0071	16.12026	P0096	10.00478
P0022	8	P0047	6.358666667	P0072	6.3456	P0097	19.47054
P0023	1.86318	P0048	24.831	P0073	8	P0098	20.44492
P0024	4.34231	P0049	8	P0074	4.31656	P0099	20.59056
P0025	0.828533333	P0050	15.41994	P0075	135	P0100	8

表 7.3 出租车收入一览表

出租车编号	收入	出租车编号	收入	出租车编号	收入
1	31.62068	21	86.59852	41	0
2	16	22	25.9794	42	27.99468
3	60.25946667	23	16	43	44.91124
4	0	24	16	44	0
5	14.787	25	31.51766667	45	0
6	16	26	16	46	66.17376
7	29.66776	27	59.85728	47	72.46116
8	16	28	16	48	32.24052
9	39.90168	29	29.44512	49	18.1044
10	28.56472	30	14.94828	50	20.007
11	34.42116	31	26.45564	51	16
12	0	32	19.31364	52	16
13	154.95672	33	16	53	0
14	16	34	49.662	54	50.84944
15	0	35	38.63177333	55	18.00576
16	16	36	16.26408	56	9.103466667
17	24.62328	37	0	57	16
18	22.6512	38	0	58	0
19	16	39	352.6528	59	0
20	145.26384	40	16	60	0

具体结果见附录文件夹中的“plan.txt”、“person_cost.xls”及“taxi_income.xls”文件。

7.4 方案分析

由以上结果可以得下表：

表 7.4 方案数据总结表

出租车总数量（辆）		667
乘客总人数（人）		1001
所选出租车数量（辆）		60
所选乘客总数（人）		100
方案实际用出租车数量（辆）	单人乘车（辆）	14
	双人合乘出租车数量（辆）	22
	三人合乘出租车数量（辆）	14
	合计（辆）	50

（1）由上表可以看出：执行程序时，所选择的乘客数量为 100 名，与原始数据的比值大致为 10:1，所选的出租车数量为 60，与原始数据的比值亦约为 10:1。最终得到的方案共用到 50 辆出租车，与所选的样本的比值为 6:5。

（2）方案中的合乘情况以双人合乘居多，虽然题目要求设计合乘方案，但是算法得到的结果仍无法避免单乘情况的发生。其原因一方面受算法迭代次数的影响，另一方面，还需考虑到，这种情况在实际中是容易发生的，因此接受算法得出的方案。

8. 模型评价

8.1 模型优点

本文提出了“有座即可合乘”的合乘模式，在不超过出租车承载量的情况下，快速准确地为乘客派车。将合乘调度问题转化为带约束路径规划问题，基于遗传算法提出优化方案，通过选取乘客距离最近五辆车作为预优化选择，降低了问题的复杂度，提出一种嵌套型自然数编码方式，以总距离的倒数为优化函数，对乘客等待时间及出租车使用数目均有优化，并对不满足约束的个体施加惩罚因子，加快遗传算法的收敛，保证了结果的最优性。

8.2 模型缺点

由于乘客数量达到 1001 位，这使得个体的编码达到 1001 位。遗传算法中的种群规模取得较大时，程序的复杂度较高，程序运行的时间较长，系统不能实时响应，这会影响乘客、出租车司机的体验。

参考文献

[1]卢川,吴群.城市居民出行合乘出租车问题的研究.道路交通与安全.2005,5(7)：52—55.
[2]孙中悦.车辆路径问题的仿真优化研究;（硕士学位论文），北京，北京交通大学，2011.
[3]基于遗传算法的动态出租车合乘模型研究[J]. 程杰,唐智慧,刘杰,钟流. 武汉理工大学学报(交通科学与工程版). 2013(01)
[4]陈漫漫.大连市出租车行业政府规划问题研究;（硕士学位论文），大连，东北财经大学，2012.
[5]张冬生.出租车行业现状及价格调查与分析.价格理论与实践—思考篇.2005(06):18-19.
[6]张瑾.出租车“拼车”问题研究及其服务系统设计实现;（硕士学位论文），兰州，兰州交通大学，2009.

附录 1

```
clear;
NP =50;    %    种群大小
NG=5000;     %  最大进化代数
Pc=0.75;     %  杂交（交叉）概率
Pm=0.1;      %   变异概率
          %   fv：目标函数的最小值
          %   xm：目标函数取最小值时的自变量值
        %   FA：最优化方案矩阵
        %   Result:标准输出格式结果

%%%%    读取数据
[request]=xlsread('E:\数学建模\A 题\requests.csv');
[taxi]=xlsread('E:\数学建模\A 题\taxi.csv');
pickup=request(1:500,1:2);dropoff=request(1:500,3:4);
now_taxi=taxi(1:300,1:2);
[p,~]=size(pickup);     %    乘客数量
[q,~]=size(now_taxi);   %    汽车数量
L=p;
[carf,mindis,bus]=mincar(now_taxi(:,1),now_taxi(:,2),pickup(:,1),
pickup(:,2));  %%%寻找最近的五辆车
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%设计遗传算法
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% p 编码长度

%%%   种群初始化   %%%
for i=1:NP
    x(i,:)=datasample(1:5,p);          %   种群初始化
[fx(i),fangan{i},wrongs]=zdistance(x(i,:),carf,now_taxi,pickup,dr
opoff); %%%   fx(i)个体适应值；初始化个体适应值
end

%%%     最优选择 / 代数运算   %%%
for k=1:NG
    [Bestfit(k),I]=max(fx);      %每代的最优适应值
    Meanfit(k)=min(fx);          %每代的最差适应值
    fv=Bestfit(k);               %取个体中最好值作为最终结果
    xv=x(I,:);                   %取最好的个体
    FA=fangan{I};                %对应的方案
```

```
    sumfx=sum(fx);                          %所有个体适应值之和
    Px=fx/sumfx;                            %所有个体适应值占总体的比重
    PPx=0;
     PPx(1)=Px(1);
     for i=2:NP                            %用于轮盘赌策略的概率累加
         PPx(i)=PPx(i-1)+Px(i);
     end
     for i=1:NP
         sita=rand();
         for n=1:NP;
             if sita<=PPx(n)
                 SelFather=n;              %根据轮盘赌策略决定父亲
                 break;
             end
         end
         Selmother=floor(rand()*(NP-1))+1;%随机选择母亲，floor 向下取整
         posCut1=floor(rand()*(p-2))+1;    %随机确定交叉点
         %posCut2=floor(rand()*(p-2))+1;
         %posCut3=floor(rand()*(p-2))+1;
         %posCut4=floor(rand()*(p-2))+1;
         %posCut=sort([posCut1 posCut2 posCut3 posCut4])
         r1=rand();
         if r1<=Pc                          %交叉
             nx(i,1:p)=x(SelFather,1:p);
        nx(i,(posCut1+1):min((posCut1+10),p))=x(Selmother,(posCut1
    +1):min((posCut1+10),p));
             r2=rand();
             if r2<=Pm                     %变异，随机
                 posMut=round(rand()*(p-1)+1);
                  nx(i,posMut)= randsample(5,1);
             end
         else
             nx(i,:)=x(SelFather,:);
         end
     end
     x=nx;                                  %繁衍
       for i=1:NP;
         [fx(i),fangan{i},wrongs]=zdistance(x(i,:),carf,now_taxi,pi
    ckup,dropoff);;%子代适应值
       end
end
Result=result_zhuanhuan(FA);%将最优方案转换为标准输出格式
```

```
function [carf,mindis,bus]=mincar(taxi_x,taxi_y,pickup_x,pickup_y)
%%%选最近的 5 辆车
%%% carf--5 辆车的编号
%%% mindis--五辆车到坐车人的距离
%%%1,2,3,4,5 距离依次减小
%%% bus--车为行，人为列，每辆车对应的选择人
n = size(taxi_x,1);
m = size(pickup_x,1);
mindis=1000*ones(m,5);
carf=zeros(m,5);
bus=zeros(n,10);
for i=1:m
    for j=1:n

distance=abs(taxi_x(j)-pickup_x(i))+abs(taxi_y(j)-pickup_y(i));
          if distance<mindis(i,1)
             mindis(i,5)=mindis(i,4);carf(i,5)=carf(i,4);
             mindis(i,4)=mindis(i,3);carf(i,4)=carf(i,3);
             mindis(i,3)=mindis(i,2);carf(i,3)=carf(i,2);
             mindis(i,2)=mindis(i,1);carf(i,2)=carf(i,1);
             mindis(i,1)=distance;   carf(i,1)=j;
          elseif distance<mindis(i,2)
             mindis(i,5)=mindis(i,4);carf(i,5)=carf(i,4);
             mindis(i,4)=mindis(i,3);carf(i,4)=carf(i,3);
             mindis(i,3)=mindis(i,2);carf(i,3)=carf(i,2);
             mindis(i,2)=distance;   carf(i,2)=j;
          elseif distance<mindis(i,3)
             mindis(i,5)=mindis(i,4);carf(i,5)=carf(i,4);
             mindis(i,4)=mindis(i,3);carf(i,4)=carf(i,3);
             mindis(i,3)=distance;   carf(i,3)=j;
          elseif distance<mindis(i,4)
             mindis(i,5)=mindis(i,4);carf(i,5)=carf(i,4);
             mindis(i,4)=distance;   carf(i,4)=j;
          elseif distance<mindis(i,5)
             mindis(i,5)=distance;carf(i,5)=j;
          else
          end
       end
    end

    for  i=1:n
         b=1;
         for j=1:m
```

```
            if carf(j,1)==i
                bus(i,b)=j;b=b+1;
            end
        end
    end

    function
[zdist,fangan,wrongs]=zdistance(mark,carf,pcar,pickup,dropoff)
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%% 函数作用：对一种编码方式并求最优顺序方案及最小距离总和
    %%% mark--样本编码 1 m1
    %%% carf--人对应的最优五辆车矩阵 nx5
    %%% zdist--出租车行驶距离之和 数
    %%% pcar--车的位置 矩阵 mx2
    %%% pickup--人的起点矩阵 nx2
    %%% dropoff--人的终点矩阵 nx2
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    [mcar,~]=size(pcar);
    [ ~,m1 ]=size(mark);
    %[~,~]=size(carf);
    carp=zeros(m1);
    for i=1:m1
        carp(i)=carf(i,mark(i));   %%%  解码
    end

    %%%%%%  读取车的载人信息
    wrongs=0;
    distance=10000*ones(1,mcar);
    fangan=zeros(mcar,7);
    for i=1:mcar

        P=find(carp==i);
        [s1,s2]=size(P);
        switch (s1+s2)
            case 2  %%%1 个人上车
    distance(i)=dista(pcar(i,:)',pickup(P(1),:)',dropoff(P(1),:)');
                fangan(i,:)=[i ,P(1) ,-P(1),0 ,0, 0, 0];
            case 3  %%%2 个人上车
                %%%% a[2,4]
                a=[pickup(P(1),:)' ,pickup(P(2),:)' ,dropoff(P(1),:)',
dropoff(P(2),:)'];
                b=[P(1) P(2) -P(1) -P(2)];
                perm=perms([1 2 3 4]);[p1 , ~]=size(perm);
                [pyperm1,~]=find(perm'==1);[pyperm2,~]=find(perm'==2);
```

```
            [pyperm3,~]=find(perm'==3);[pyperm4,~]=find(perm'==4);
            for j=1:p1

                if (pyperm1(j)>pyperm3(j))||(pyperm2(j)>pyperm4(j))

                else
   temp=dista(pcar(i,:)',a(:,perm(j,1)),a(:,perm(j,2)),a(:,perm(j,3))
,a(:,perm(j,4)));
                    if temp<distance(i)
                        distance(i)=temp;
   fangan(i,:)=[i ,b(perm(j,1)) ,b(perm(j,2)) ,b(perm(j,3)) ,b(perm(j,4)
),0,0];
                    else
                    end
                end
            end

         case 4  %%%3 个人上车
   a=[pickup(P(1),:)',pickup(P(2),:)',pickup(P(3),:)',dropoff(P(1),:)
' ,dropoff(P(2),:)' ,dropoff(P(3),:)'];
            b=[P(1) ,P(2) ,P(3) ,-P(1) ,-P(2),-P(3)];
            perm=perms([1 2 3 4 5 6]);[p1 , ~]=size(perm);
            [pyperm1,~]=find(perm'==1);[pyperm2,~]=find(perm'==2);
            [pyperm3,~]=find(perm'==3);[pyperm4,~]=find(perm'==4);
            [pyperm5,~]=find(perm'==5);[pyperm6,~]=find(perm'==6);
            for j=1:p1
                if
(pyperm1(j)>pyperm4(j))||(pyperm2(j)>pyperm5(j))||(pyperm3(j)>pyperm6
(j))
                else
   temp=dista(pcar(i,:)',a(:,perm(j,1)),a(:,perm(j,2)),a(:,perm(j,3))
,a(:,perm(j,4)),a(:,perm(j,5)),a(:,perm(j,6)));
                    if temp<distance(i)
                        distance(i)=temp;
   fangan(i,:)=[i ,b(perm(j,1)) ,b(perm(j,2)) ,b(perm(j,3)) ,b(perm(j,4)
),b(perm(j,5)),b(perm(j,6))];
                    else
                    end
                end
            end
         case 1 %%%无此车
            fangan(i,:)=[i 0 0 0 0 0 0];distance(i)=0;
         Otherwise  %%%    施加惩罚值
            wrongs=1;fangan(i,:)=[i 0 0 0 0 0 0];distance(i)=1000;
```

```
        end
    end

    %%%%%%%%%%  求距离总和
    zd=0;
    for i=1:mcar
        zd=zd+distance(i);
    end
    zdist=1/zd;%%% 适应值，越大越好

    function distances=dista(varargin)

    %%%%%%%%%%%%%%%%%%%%%%%%%
    %%%按顺序求算各点之间距离
    %%%%%%%%%%%%%%%%%%%%%%%%%
    doto=zeros(2,nargin);
    for i=1:nargin
        doto(:,i)=varargin{i};
    end
    distances=0;
    %%%判断两点是否在相对区域,并计算总距离
    for i=1:(nargin-1)
        x1=doto(1,i);y1=doto(2,i);x3=doto(1,i+1);y3=doto(2,i+1);

    t1=floor(x1/0.5);m1=floor(y1/0.5);t2=floor(x3/0.5);m2=floor(y3/0.
5);
        d=abs(x1-x3)+abs(y1-y3);
        if (d>=0.5)&&((t1==t2)||(m1==m2))
            if t1==t2

a=min(abs(round(x3/0.5)-x3/0.5),abs(round(x1/0.5)-x1/0.5));
            else

a=min(abs(round(y3/0.5)-y3/0.5),abs(round(y1/0.5)-y1/0.5));
            end
               temp=abs(x1-x3)+abs(y1-y3)+a;
        else
           temp=abs(x1-x3)+abs(y1-y3);
        end
        distances=distances+temp;
    end

    function a=result_zhuanhuan(x)
    %%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%% 作用：将数字结果转换字符标准格式
%%%%
[m,n]=size(x);
i=0;
for j=1:m
    if x(j,2)==0

    else
        i=i+1;
        temp=num2str(x(j,1));
        switch length(temp)
            case 1
                a{i,1}= ['t000',temp];
            case 2
              a{i,1}= ['t00',temp];
            case 3
             a{i,1}= ['t0',temp];
            otherwise
                fprintf('数据转化出错！');
        end
        for k=2:n

            if x(j,k)==0
                a{i,k}='';
            elseif x(j,k)<0
                temp=num2str(-x(j,k));
                switch length(temp)
                    case 1
                         a{i,k}=['p000',temp,'d'];
                    case 2
                        a{i,k}=['p00',temp,'d'];
                    case 3
                        a{i,k}=['p0',temp,'d'];
                    otherwise
                        fprintf('数据转化出错！');
                end
            elseif x(j,k)>0
                temp=num2str(x(j,k));
                switch length(temp)
                    case 1
                         a{i,k}=['p000',temp,'u'];
                    case 2
                        a{i,k}=['p00',temp,'u'];
                    case 3
```

```
                    a{i,k}=['p0',temp,'u'];
                otherwise
                    fprintf('数据转化出错！');
            end
        end
    end
end
end
```

附录 2

输出的文本文档内容如下：

0050

t0001 p0044u p0044d p0008u p0008d 0000 0000

t0002 p0059u p0059d p0026u p0026d 0000 0000

t0003 0000 0000 0000 0000 0000 0000

t0004 p0035u p0035d 0000 0000 0000 0000

t0005 p0041u p0041d 0000 0000 0000 0000

t0006 p0073u p0011u p0011d p0073d p0031u p0031d

t0007 p0019u p0019d p0017u p0017d 0000 0000

t0008 p0036u p0036d 0000 0000 0000 0000

t0009 p0061u p0061d p0050u p0052u p0050d p0052d

t0010 p0078u p0078d p0060u p0060d 0000 0000

t0011 p0080u p0010u p0010d p0022u p0080d p0022d

t0012 0000 0000 0000 0000 0000 0000

t0013 p0014u p0014d 0000 0000 0000 0000

t0014 p0092u p0093u p0092d p0093d 0000 0000

t0015 p0057u p0057d p0063u p0063d 0000 0000

t0016 p0084u p0084d 0000 0000 0000 0000

t0017 p0040u p0040d p0028u p0028d 0000 0000

t0018 0000 0000 0000 0000 0000 0000

t0019 p0067u p0067d p0066u p0066d 0000 0000

t0020 p0023u p0077u p0023d p0077d 0000 0000

t0021 p0055u p0055d 0000 0000 0000 0000

t0022 p0074u p0007u p0004u p0007d p0074d p0004d

t0023 p0087u p0069u p0087d p0069d 0000 0000

t0024 p0056u p0062u p0043u p0056d p0062d p0043d

t0025 0000 0000 0000 0000 0000 0000

t0026 p0099u p0085u p0072u p0099d p0072d p0085d

t0027 p0006u p0015u p0086u p0086d p0006d p0015d

t0028 p0100u p0100d 0000 0000 0000 0000

t0029 p0076u p0076d 0000 0000 0000 0000

t0030 p0071u p0071d 0000 0000 0000 0000

t0031 p0039u p0034u p0034d p0005u p0039d p0005d

t0032 p0012u p0012d p0091u p0021u p0021d p0091d

t0033 0000 0000 0000 0000 0000 0000

t0034 0000 0000 0000 0000 0000 0000

t0035 p0047u p0025u p0025d p0096u p0047d p0096d

t0036 p0070u p0042u p0070d p0042d 0000 0000

t0037 p0097u p0009u p0095u p0095d p0097d p0009d

t0038 p0024u p0024d 0000 0000 0000 0000

t0039 p0029u p0003u p0003d p0029d 0000 0000

t0040 p0037u p0037d 0000 0000 0000 0000

t0041 p0045u p0045d 0000 0000 0000 0000

t0042 p0002u p0002d p0032u p0032d 0000 0000

t0043 p0053u p0030u p0030d p0082u p0053d p0082d

t0044 0000 0000 0000 0000 0000 0000

t0045 p0094u p0048u p0048d p0094d 0000 0000

t0046 p0027u p0027d 0000 0000 0000 0000

t0047 0000 0000 0000 0000 0000 0000

t0048 p0090u p0033u p0033d p0090d 0000 0000

t0049 p0001u p0038u p0001d p0038d 0000 0000

t0050 p0046u p0046d p0079u p0079d p0081u p0081d

t0051 p0065u p0058u p0065d p0058d 0000 0000

t0052 p0064u p0083u p0064d p0083d 0000 0000

t0053 p0013u p0013d p0068u p0068d 0000 0000

t0054 p0075u p0054u p0054d p0075d p0098u p0098d

t0055 0000 0000 0000 0000 0000 0000

t0056	p0051u	p0018u	p0018d	p0051d	0000	0000
t0057	p0089u	p0020u	p0020d	p0089d	0000	0000
t0058	0000	0000	0000	0000	0000	0000
t0059	p0049u	p0049d	0000	0000	0000	0000
t0060	p0016u	p0088u	p0088d	p0016d	0000	0000