考虑时空效率的任务匹配定价模型与算法

摘要

本文运用 FCM 聚类方法和多阶段轮盘赌算法，解决了“拍照赚钱”APP 平台日常业务中不同任务和会员分布条件下的任务定价问题。

对于问题一，首先将平台数据导入 MapInfo Pro15.0 软件，观察得出平台定价以城市中心区为重心，以距离大小向周围辐射梯度递增的规律。建立 FCM 地理中心聚类模型，确定了 6 个定价中心；建立中心梯度线性定价模型并通过一元线性回归分别得到 6 个城市的定价模型参数。

对于问题二，从会员客户关系定价的时间、空间和效率三维影响因素着手，首先对会员信誉度进行分级，并对各任务整合时空高效和时空可抵两类距离构建了备选会员集，建立了时空效率定价模型和任务匹配规划模型；考虑任务预订开始时间和任务配额等约束，设计二维多阶段轮盘赌多项式时间算法，运用 Matlab2015b 编程求解模型，结果说明提出的时空效率定价模型较原定价方案优化完成了 20.2%的未完成任务。

对于问题三，考虑区域任务打包策略，构建了供需平衡模型，获得最优打包任务数规模为 3；并以打包规模区间[2,5]为参数，构建了会员信誉等级定价模型，增大具有高信誉度的会员获得打包任务的概率；求解结果较不打包模式下成功完成的任务规模优化了 6.5%的比例。

对于问题四，首先将新项目数据导入 MapInfo Pro15.0 软件，观察得任务匹配具有“供大于求，高聚集态”等特征，构建了歧视激励定价模型；通过提升定价激励高信誉度会员高质量选择并完成任务，通过歧视定价为低信誉度会员设置保留价格和准入门槛；将新项目通过会员信誉等级定价和提出的歧视激励定价模型分别求解，求得 34.7%和 46.9%的完成率，验证了歧视激励定价的有效性。

本文通过查阅、引用、自主推导、编程模拟等多种方式，深入分析了“拍照赚钱”APP 平台定价机制，并取得一定程度的优化效果，具备相应参考借鉴价值。

关键字： FCM 聚类；中心梯度定价；时空效率定价；匹配；多阶段轮盘赌算法

第1章 问题重述

1.1 问题背景

目前，基于移动互联网的自助式劳务众包平台已经成为一种调查数据的新方式，这种模式大大节省了调查成本，保证了调查数据的真实性，缩短了调查的周期。例如：“拍照赚钱”的运营模式就是会员从 APP 上领取需要拍照的任务，赚取 APP 对任务所标定的酬金。在这种模式中，APP 成为该平台运行的核心，而 APP 中的任务定价又是其最为关键的要素。

1.2 问题提出

1. 研究“拍照赚钱”APP 一个已结束项目的任务数据，包括每个任务的位置、定价和完成情况，来分析“拍照”APP 的任务定价规律和任务未完成的原因。
2. 设计一种新的定价方案，通过对已结束项目数据的仿真运算，与原定价方案进行对比。
3. 考虑多个任务可能因为位置比较集中的情况，提出将这些任务联合在一起打包发布的方法。在这种考虑下修改问题二的定价方案，并对其结果进行分析评价。
4. 结合附件三中给出的新项目，给出一种新方案，并检验其实施效果。

第2章 模型假设及符号说明

2.1 模型假设

1. 假设任务的定价在匹配过程中始终不变。

2. 假设每个任务被匹配后仅有“成功执行”和“未执行”两种状态，任务匹配后因“未执行”不再重新返回平台进行再匹配。

3. 假设会员选择任务时，不受外界因素干扰，只受其信誉度、开始预订时间和配额确定的选择概率影响。

2.2 符号说明

序号	符号	符号说明
1	x_i	数据集，$i=1,2,...,n$
2	u_{ij}	U 中的元素，$u_{ij}\in(0,1)$
3	c_i	模糊组 I 聚类中心，$i=1,2,...,c$
4	J	FCM 的价值函数
5	d_{ij}	第 I 个聚类中心与第 j 个数据点间的欧几里德距离 $d_{ij}=\|c_i-x_j\|$

6	m	加权指数，$m \in [1,\infty)$
7	λ_j	是 $\sum_{i=1}^{c} u_{ij} = 1, \forall j = 1,...,n$ 式的 n 个约束式的拉格朗日乘子
1	A	任务集合，任务 j 包含任务编号 a_j 、地理位置经纬度坐标 (x_j^A, y_j^A) 和任务定价 P_j， $j=1,2,\cdots,\|A\|$
2	B	会员集合，会员 i 包含会员编号 b_i 、经纬度坐标 (x_i^B, y_i^B)、任务配额 c_i^B 和任务预订开始时间 t_i^B 和信誉度 q_i^B， $i=1,2,\cdots,\|B\|$
3	G_j	任务 j 的备选会员集，G_j^{up} 为可抵备选集，G_j^{low} 为高效备选集
4	C	聚类中心集合，聚类中心编号为 p， $p=1,2,3...,\|C\|$
5	z_{ij}	0-1 变量，当任务 j 被会员 i 执行时取值为 1，否则为 0
6	F_j	任务 j 被匹配的会员编号，$F_j \in \{b_i\}$， $j=1,2,3...,\|A\|$
7	P_{pj}	聚类中心 p 内任务 j 的定价， $p=1,2,3...,\|C\|$， $j=1,2,3...,\|A\|$
8	Po_j	附件 1 中任务 j 在 APP 中的初始定价， $j=1,2,3...,\|A\|$
9	K_p	聚类中心 p 的经济发展系数， $p=1,2,3...,\|C\|$
10	D_{ij}	会员 i 与任务 j 间的欧氏距离，$D_{ij} = \sqrt{(x_i^B - x_j^A)^2 + (y_i^B - y_j^A)^2}$
11	$r_1(j)$	任务 j 的时空高效距离，H_1 为其会员信息系数， $j=1,2,3...,\|A\|$
12	$r_2(j)$	任务 j 的时空可抵距离，H_2 为其会员信息系数， $j=1,2,3...,\|A\|$
13	$T(m)$	批次 m 的匹配开始时间，T 为时间的升序序列， $m=1,2,3...,\|T\|$

第3章 模型的建立与求解

3.1 问题一模型的建立及求解

3.1.1 问题分析

问题一要求有效处理附件 1 中的相关数据，找出 APP 平台现有定价规律。

为使数据更直观，需将附件 1 中任务地理特征在 MapInfo Pro15.0 中进行处理，以根据各任务经

纬度生成具体坐标点位属性，绘制专题统计图后，观察定价和任务执行情况的空间地理分布特征。

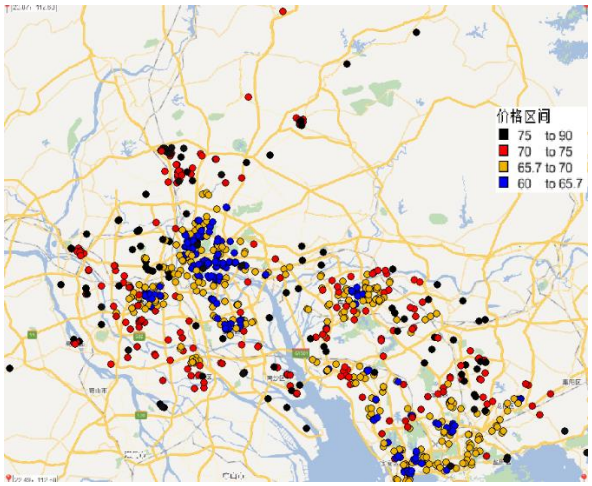


图 3.1　任务价格空间分布

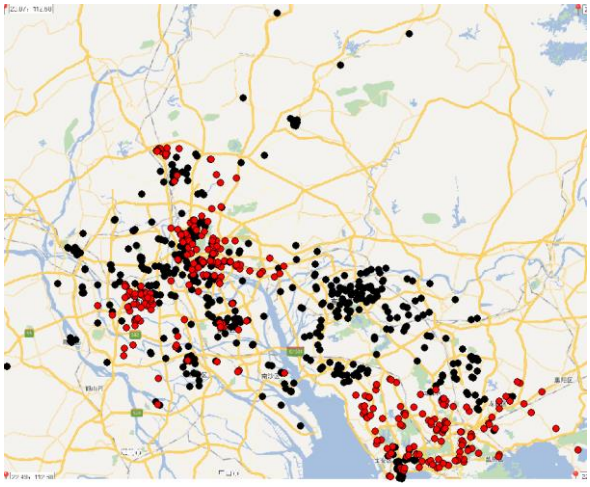


图 3.2 任务完成情况空间分布

分析图 3.1 可知，市中心附近价格项目定价主要集中在（65-67.5）区间；在城市非中心地带和城乡地区，项目定价以（67.5-72.5）为主；而高价区（>72.5）主要分布在距离城市较远的地区。分析图 3.2 可知，未完成任务大多分布于城区。综上所述，我们可以推测 APP 平台的原始定价规律如下：

（1）定价高低与任务位置到市中心的距离有关，并随距离的增加发生一定梯度的变化。

（2）未完成任务集中于市中心并与中低价区高度重合，推测其未完成可能与会员行为选择有关。

为将上述定价规律进行数学量化，可通过 FCM 聚类法针对空间任务聚集特性进行定价中心坐标求解，并按照其周围任务距离分布和定价值进行线性拟合，获得地理中心梯度定价数学模型。

3.1.2　FCM 地理中心聚类模型

首先，对本问题的项目位点进行聚类，需要考虑以下两个方面：1.坐标位点没有明确的类别标记，其界限并不分明；2.对整个地图范围内项目位点进行聚类以求得最优解。因此，我们采用模糊聚类中的模糊 C-均值（FCM）进行聚类，相比于传统聚类方法只能在局部最好最优解，该方法能够通过极小化目标函数求得全局最优解。

聚类就是按照某个特定标准，如距离准则，把一个数据集分割成不同的类或簇，使得同一个簇内的数据对象的相似性尽可能大，同时不在同一个簇中的数据对象的差异性也尽可能地大。目前，主要的聚类算法可以硬聚类和模糊聚类两种。其中在硬聚类方法：如划分方法、层次方法等，每一个数据只能被归为一类。而模糊聚类通过隶属函数来确定每个数据隶属于各个簇的程度，而不是将一个数据对象硬性地归类到某一簇中。相比于前面的”硬聚类“，模糊聚类算法计算每个样本对所有类的隶属度，能够将一个没有类别标记的样本按照某种规则划分成若干个子集，使相似样本尽可能归为一类。

在本题中，由于没有给出明确的分类指标，所以我们采用模糊聚类法中的 FCM 地理中心聚类模型，其一般化形式为:

$$J(U,c_1,...,c_c)=\sum_{i=1}^{c}J_i=\sum_{i=1}^{c}\sum_{j}^{n}u_{ij}^{m}d_{ij}^{2} \tag{3.1}$$

这里 u_{ij} 介于 0，1 间；c_i 为模糊组 I 的聚类中心，$d_{ij}=\left\|c_i-x_j\right\|$ 为第 I 个聚类中心与第 j 个数据点间的欧几里得距离；且 $m\in[1,\infty)$ 是一个加权指数。

构造如下新的目标函数，可求得使（3.1）式达到最小值的必要条件：

$$\begin{aligned}\overline{J}(U,c_1,...,c_c,\lambda_1,...,\lambda_n)&=J(U,c_1,...,c_c)+\sum_{j=1}^{n}\lambda_j(\sum_{i=1}^{c}u_{ij}-1)\\&=\sum_{i=1}^{c}\sum_{j}^{n}u_{ij}^{m}d_{ij}^{2}+\sum_{j=1}^{n}\lambda_j(\sum_{i=1}^{c}u_{ij}-1)\end{aligned}\tag{3.2}$$

3.1.3 地理中心梯度定价数学模型

采用一次线性拟合，将二维平面内的点拟合在一条直线上，其公式为：

$$f(x)=ax+b\tag{3.3}$$

3.1.4 FCM 地理中心聚类模型的求解

1.聚类中心数确定

通过附件一所给的任务经纬度坐标数据来看，任务热点区域与人们聚集密度有很大关系，即人口密度大的地方往往任务分布较为集中。因此，城市中心附近任务聚集度较高，从图中反映从西到东依次为佛山市、广州市、番禺区、东莞市、深圳市宝安区和龙华区。由此，我们确定聚类中心数 c 为 6。

2.FCM 聚类

通过调用 MATLAB 函数中[center, U, obj_fcn] = FCM(data, cluster_n, options)语句。其中 data 为附件一中各项目坐标经纬度数据，cluster_n 为聚类中心心数，这里把它设为 6。Options 中，隶属度矩阵 U 的指数设为 3.0；最大迭代次数设为 50 次，迭代待终止条件为隶属度最小变化量小于 1e-5。聚类结果如下图 3.3 所示：

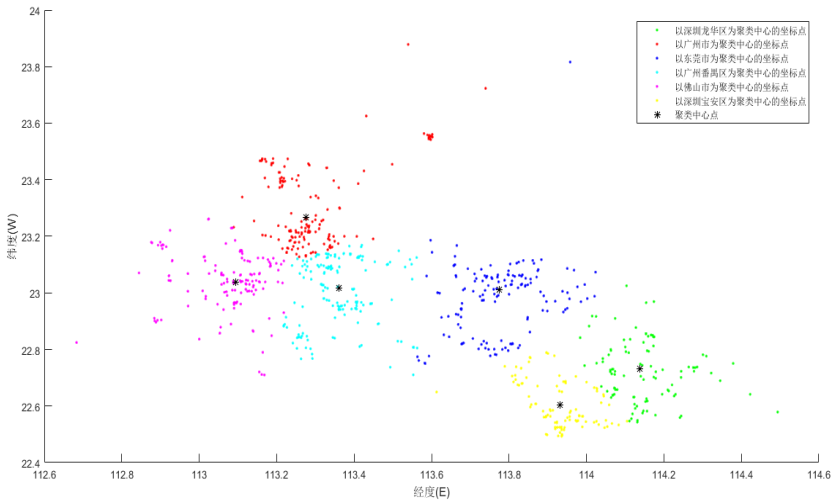


图 3.3 FCM 聚类位点图

3.聚类有效性分析

为了验证上述聚类数的合理性，保证算法得到有效的分类结果，我们对在 6 个聚类中心情况下的 FCM 算法进行算法有效性分析。

聚类有效性分析标准有两种：一是外部标准，通过测量聚类结果和参考标准的一致性来评价聚类结果的优良；另一种是内部指标，用于评价同一聚类算法在不同聚类数条件下聚类结果的优良程度，通常用来确定数据集的最佳聚类数。 我们采用内部指标汇中的 Calinski-Harabasz(CH)指标，Davies-Bouldin(DB)指标进行有效应分析。

（1） CH 指标

CH 指标通过类内离差矩阵描述紧密度，类间离差矩阵描述分离度，指标定义为

$$CH(k)=\frac{trB(k)/(k-1)}{trW(k)/(n-k)} \tag{3.4}$$

其中，n 表示聚类的数目，k 表示当前的类，$trB(k)$ 表示类间离差矩阵的迹，$trW(k)$ 表示类内离差矩阵的迹。CH 越大代表着类自身越紧密，类与类之间越分散，即更优的聚类结果。

（2） DB 指标

DB 指标通过描述样本的类内散度与各聚类中心的间距，定义为

$$DB(k)=\frac{1}{K}\sum_{i=1}^{K}\max_{j=1\sim k,j=i}\left(\frac{W_i+W_j}{C_{ij}}\right) \tag{3.5}$$

其中，K 是聚类数目，W_i 表示类 C_i 中的所有样本到其聚类中心的平均距离，W_j 表示类 C_i 中的所有样本到类 C_j 中心的平均距离，C_{ij} 表示类 C_i 和 C_j 中心之间的距离。可以看出，DB 越小表示类与类之间的相似度越低，从而对应越佳的聚类结果。

通过对本题 FCM 算法的有效性检验，得到 CH=0.863,DB=124.1。聚类有效性分析结果较好，中心数为 6 的聚类分类有效。

4.结果分析

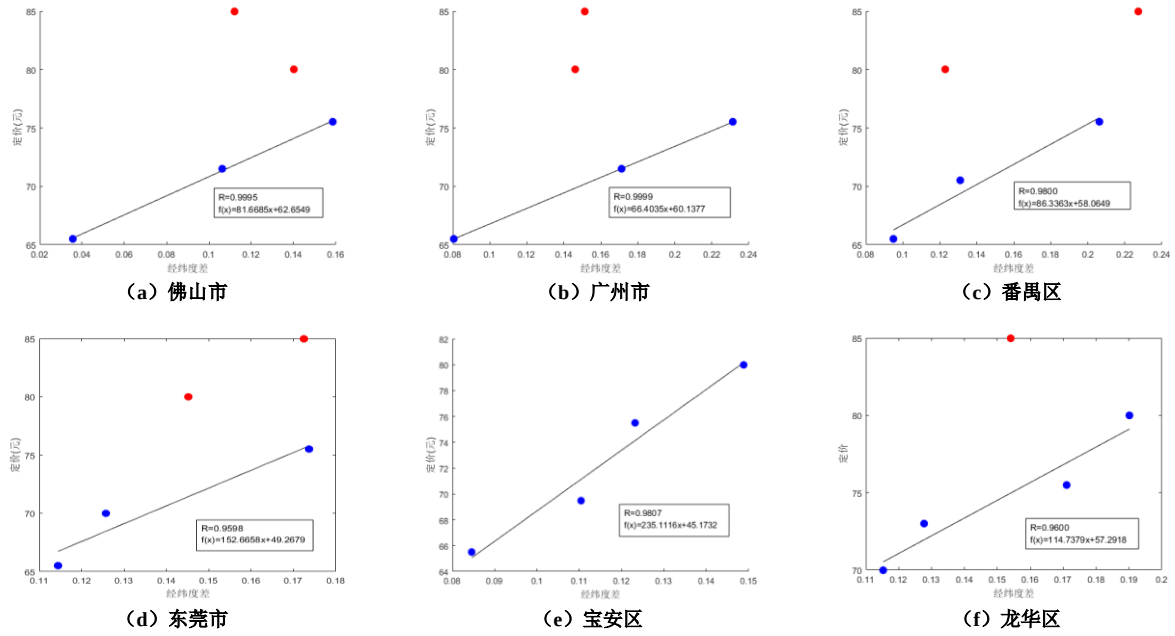
通过 FCM 聚类模型，我们对项目经纬坐标进行有效聚类，并形成聚类中心数为 6 的集群分布。从聚类中心地理坐标看，六个聚类中心分别位于佛山市、广州市、番禺区、东莞市、深圳市宝安区和龙华区。与之前我们之前的判断结果一致。

为了更好的描述项目定价标准和项目距离聚类中心点之间的关系，排除不必要的干扰。我们对同一类别下不同定价标准的位点进行期望处理:

$$EX=\sum_{i=1}^{n}x_i p_i \tag{3.6}$$

这里，x_i 为位点距离聚类中心的距离。 p_i 为 $1/n$ 。

以距离期望为横坐标，定价为纵坐标进行一元线性回归，得到距离与价格的关系如下图所示:



（a）佛山市　　（b）广州市　　（c）番禺区

（d）东莞市　　（e）宝安区　　（f）龙华区

图 3.4 六个聚类中心距离和价格的一次线性回归

从 MATLAB 运行的结果图中可以看出，定价标准与任务位置到中心点的距离呈一次线性关系:

（1） 以佛山市为中心的聚类，距离 x 与定价 $f(x)$ 的关系为： $f(x)=81.6685x+62.6549$；

（2） 以广州市为中心的聚类，距离 x 与定价 $f(x)$ 的关系为： $f(x)=66.4035x+60.1377$；

（3） 以番禺区为中心的聚类，距离 x 与定价 $f(x)$ 的关系为： $f(x)=86.3363x+58.0649$；

（4） 以东莞市为中心的聚类，距离 x 与定价 $f(x)$ 的关系为：$f(x)=152.6658x+49.2679$；

（5） 以宝安区为中心的聚类，距离 x 与定价 $f(x)$ 的关系为：$f(x)=235.1116x+45.1732$；

（6） 以龙华区为中心的聚类，距离 x 与定价 $f(x)$ 的关系为：$f(x)=114.7379x+57.2918$；

从上述式子来看，拟合后的一次线性函数线性度 R 均在 0.90 以上，拟合效果较好。在六个聚类中心，定价标准基本分为 5 档：65±2.5、70±2.5、75±2.5、80±2.5、85±2.5（单位/元）。±2.5 偏差的出现，体现了平台价格浮动策略，根据实际任务的难易在一定区间内上下浮动，来刺激任务接收与执行。

图中与直线存在较大偏差的红点的的出现，红点处定价高，取在 80±2.5 和 85±2.5 这两档，而这两档的定价标准有着奖励的性质，主要由于任务少有人接，平台为了提高任务的被执行率，作为奖励而提高价格，所以其在图中与线的位置距离呈现不规则性。

3.2 问题二模型的建立及求解

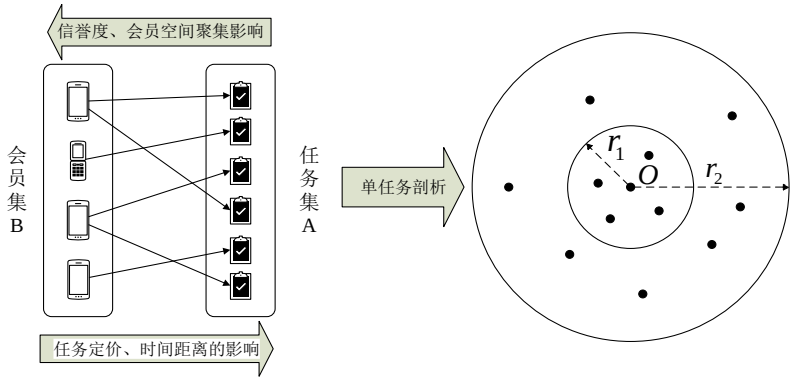
3.2.1 问题分析

“拍照赚钱”商业模式本质为会员和任务间的一种匹配过程。从宏观条件下分析，这种匹配呈现“多对一”的特征，将影响因素按匹配主体双方分类，进行 3.5 (a)所示的两个过程分析：

（1） 会员选择任务：任务定价越高，距离越近，会员越乐意选择；

（2） 任务匹配会员：会员信誉度越高，一定空间内会员聚集数量越多，任务被执行概率越高。

分析单个任务点，如图 3.5 (b)，周围会员分布情况很大程度上影响了该任务的执行效率。聚集的高信誉度、高配额会员数量越多，任务被执行的概率越大；反之则概率越小。参照匹配时空规范化距离概念[1]，为了使任务能够尽可能的被会员执行，平台在进行任务匹配时，会先匹配距离较近的部分会员，在其没有接单后，再考虑匹配较远会员。因此，匹配时需对高效距离和可抵距离分类讨论。



(a). 匹配模式　　(b). 时空距离

图 3.5 “拍照赚钱”模式解析

3.2.2 时空效率定价模型

设计新的定价方案，建立时空效率定价模型，主要从以下三个维度，全面考虑了定价影响因素：

（1）时间维度：会员与任务间的距离影响，距离影响会员抵达任务点的时间，从而影响完成率。

（2）空间维度：任务附近会员聚集情况的影响，聚集的高信誉度会员越多，完成率越大。

（3）效率维度：模型充分考虑信誉度对定价与分配的影响，算法中以任务开始时间和配额取代。

首先，对距离 D_{ij} 和信誉度 q_i^B 进行数据规范化处理，以规避影响定价的各类指标量纲间复杂的相关性和极端数据对定价模型计算带来的不良影响。

D_{ij} 为在时空可抵范围内会员 i 与任务 j 的距离，$D_i^{'}$ 为的 D_{ij} 均值方差处理[2]值：

$$D_{ij}^{'}=\frac{D_{ij}-\overline{D_{ij}}}{\sigma_{ij}},\quad i\in G_j, j\in A \tag{3.7}$$

$$\overline{D_{ij}}=\frac{1}{|G_j|}\sum_{i=1}^{i=|G_j|}D_{ij},\sigma_{ij}=\sqrt{\frac{1}{|G_j|-1}\sum_{i=1}^{i=|G_j|}(D_{ij}-\overline{D_{ij}})},\quad j\in A \tag{3.8}$$

q_i^B 为会员 i 的信誉值，依据在线支付信誉评价标准[3],将附件二中各会员按其信誉度值划分为高信誉、中信誉和低信誉三个等级，得图 3.6 所示信誉趋势，并量化得会员信誉分段函数 Q_i^B：

$$Q_i^B=\begin{cases}1.1, & q_i^B>19.9231\\ 1, & q_i^B=19.9231\\ 0.9, & q_i^B<19.9231\end{cases} \tag{3.9}$$

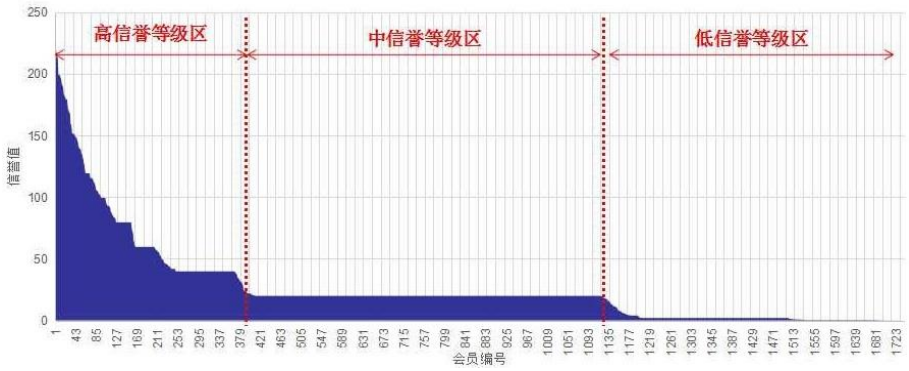


图 3.6 会员信誉度分级

由此，构建包含任务会员时间距离、空间备选会员集和会员信誉度的时空效率定价模型：

$$P_{pj}=\frac{1}{2}K_pPo_j(H_1\sum_{m=1}^{m=\left|G_j^{low}\right|}\frac{D_{mj}^{'}}{Q_m^B}+H_2\sum_{n=1}^{n=\left|G_j^{up}\right|}\frac{D_{nj}^{'}}{Q_n^B}) \tag{3.10}$$

式中，p 为聚类中心编号，$p=1,2,3...,\left|C\right|$；$j$ 为任务编号，$j=1,2,3...,\left|B\right|$；$H_1$ 和 H_2 分别为时空高效范围 $r_1(j)$ 及时空可抵范围 $r_2(j)$ 的会员信息系数值，分别取值 1.1 和 0.9。执行时空效率定价时，任务 j 先匹配 $r_1(j)$ 范围内的会员，再匹配 $r_1(j)$ 和 $r_2(j)$ 范围间的会员，故 $H_1>H_2$。

根据问题一任务未完成情况归因可知，经济发展水平较高的地区（如广州、深圳）任务未完成的概率远大于经济发展水平较低的地区（如东莞）。因此式（3.10）中设置经济发展系数 K_p 量化宏观经济因素。由此，可以通过调节定价 K_p 提高经济发达地区任务基准定价，刺激当地会员的积极性，促使其更好地完成任务。据《2016 年广东省各市 GDP 与人均 GDP 报告》赋予表 4.1 所示系数值。

表 4.1 各聚类中心经济发展系数赋值

聚类中心点	佛山市	广州市	番禺区	东莞市	宝安区	龙华区
K_p	1.02	1.13	1.08	0.97	1.15	1.19

3.2.3 任务匹配规划模型

基于时空效率定价模型重新定价后，参考竞争成功选择问题（WBDP）相关规律[5]，构建以平台任务总定价提升比例最小和任务成功执行数最大为目标的会员任务匹配规划模型。

$$\min\left(\sum_{i=1}^{i=|B|}\sum_{j=1}^{j=|A|} z_{ij}\frac{P_{pj}}{P_{Oj}}-\sum_{j=1}^{m}\sum_{i=1}^{n} z_{ij}\right) \tag{3.11}$$

$$s.t. \quad \sum_{i=1}^{i=|B|} z_{ij}\le 1, \quad j=1,2,...,|A| \tag{3.12}$$

$$\sum_{j=1}^{j=|A|} z_{ij}\le c_i^B \quad i=1,2,...,|B| \tag{3.13}$$

$$\Pr\left(F_j=b_i\right)\ge \Pr\left(z_{ij}=1\right), \quad i=1,2,...,|B|, j=1,2,...,|A| \tag{3.14}$$

目标函数（3.11）分为两部分，前部为在 APP 平台原始定价基础上成功执行任务的平台总定价，后部为任务被成功执行次数和；约束（2）为一个任务至多被一个会员成功执行；约束（3）为被同一会员成功执行的任务数不得超过该会员的任务配额；约束（4）表示成功匹配的任务不一定被执行。

3.2.4 二维多阶段轮盘赌求解算法

针对时空效率定价模型的任务匹配问题特征，我们设计了基于时空高效距离 r_1 和时空可抵距离 r_2 的“二维多阶段轮盘赌”任务会员匹配算法。对于模型中涉及会员信誉度 q_i^B 的约束，算法中可通过题设条件“参考其信誉给出的任务开始预订时间和预订配额”中的任务开始预订时间 t_i^B 和预订配额 c_i^B 作归约处理。由于距离约束受 r_1 和 r_2 限制，每个任务 j 备选会员集 G_j 的规模将有效减小，并划分为可抵会员集 G_j^{up} 与高效会员集 G_j^{low}，其中 $G_j^{up}\bigcap G_j^{low}=\varnothing$ 且 $G_j^{up}\bigcup G_j^{low}=G_j$，与全局遍历算法相比，显著降低了计算的时间复杂性。由于 G_j 元素从所有会员中遍历获得，在多阶段轮盘赌过程中，末段决策各会员被选概率之和为 1，因此算例规模下，“二维多阶段轮盘赌”算法属于精确解法。

（1）主体算法执行步骤

对包含任务编号、地理位置、任务定价的任务集 $A\left\{a_j,(x_j^A,y_j^A),P_j\right\}$ 和包涵会员编号、地理位置、配额、任务开始时间、信誉度的会员集 $B\left\{b_i,(x_i^B,y_i^B),c_i^B,t_i^B,q_i^B\right\}$ 进行匹配，获得每个任务对应的匹配解 $F_j\left\{a_j,b_i\right\}$。如某任务下 b_i=NaN，则决策变量 z_{ji}=0；否则 z_{ji}=1。

为使得会员备选集中至少有一个高信誉度会员，计算时空高效距离：

$$r_1(j)=\min_{j\in A}\left\{D_{ij}\right\}, \quad i=1,2,...,|B| \; and \; q_i^B>19.9231 \tag{3.15}$$

为保证每个任务至少有一个备选会员，以悲观准则计算时空可抵距离：

$$r_2(j)=\max_{j\in A}\min_{i\in B}\{D_{ij}\} \tag{3.16}$$

算法具体步骤如下：

Step1：初始化 $j=1$；

Step2：选中 a_j，初始化 $i=1$；

Step3：选中 b_i，若 $D_{ij}\le\min\{r_1(j),r_2(j)\}$，则将 b_i 存入 G_j^{low}，$i=i+1$；否则，$i=i+1$；

Step4：判断 $i>|B|$？若是，进入 Step5；否则，返回 Step3；

Step5：调用 BET 得 F_j^{low}，若 $z_{ji}=1$ 则 $F_j=F_j^{low}$，$c_i^B=c_i^B-1$，$j=j+1$，进入 Step9；否则 Step6；

Step6：选中 b_i，若 $D_{ij}\le\max\{r_1(j),r_2(j)\}$，则将 b_i 存入 G_j^{up}，$i=i+1$，否则，$i=i+1$；

Step7：判断 $i>|B|$？若是，进入 Step8；否则，返回 Step8；

Step8：调用 BET 得解 F_j^{up}，$F_j=F_j^{up}$，$j=j+1$；

Step9：判断 $j>|A|$？若是，结束匹配并输出所有 F_j；否则，返回 Step2。

（2）多阶段轮盘赌规则

本规则在充分考虑“具有同一任务开始预订时间的会员属于相同的优先批次”和“同批次内各会员选中任务概率依预订配额占比决定”两种情况下，设计批次内完全信息共享任务匹配规则。

分析 G_j 各元素 $g_{jn}\left(b_{jn},x_{b_{jn}}^B,y_{b_{jn}}^B,c_{b_{jn}}^B,t_{b_{jn}}^B,q_{b_{jn}}^B\right)$，$n=1,2,\cdots,|G_j|$，$b_{jn}\in B$。将会员划分为 $G_j^{T(1)}$，$G_j^{T(2)},\ldots,G_j^{T(m)}$，$\bigcup_{i=1}^{i=m}G_j^{T(i)}=G_j$。其中相同集合内的会员具有相同的 t_i^B，即 $\forall i\in G_j^{T(m)}$，$T(m)=t_i^B$。

基于提出的新定价标准，每个会员选中任务的概率与新标准价格 P_{pj} 和平台原有价格 Po_j 的比值成正比。通过式（3.17）可计算会员 b_{jn} 在集合 G_j 中选中任务 j 的概率 $\Pr(b_{jn})$：

$$\Pr(b_{jn})=\frac{c_{b_{jn}}^B}{\sum_{n=1}^{n=|G_j|}c_{b_{jn}}^B}\times\frac{P_{pj}}{Po_j} \tag{3.17}$$

由此，针对每一批次 $T(m)$，均有式（3.18）所示的未选中概率 $\Delta\Pr(m)$：

$$\Delta\Pr(m)=1-\sum_{k=1}^{k=m}\sum_{n=1}^{n=|T_m|}\Pr(b_{jn}^k) \tag{3.18}$$

多阶段轮盘赌规则的算法流程和转盘概率分布示意如图 3.7 所示。

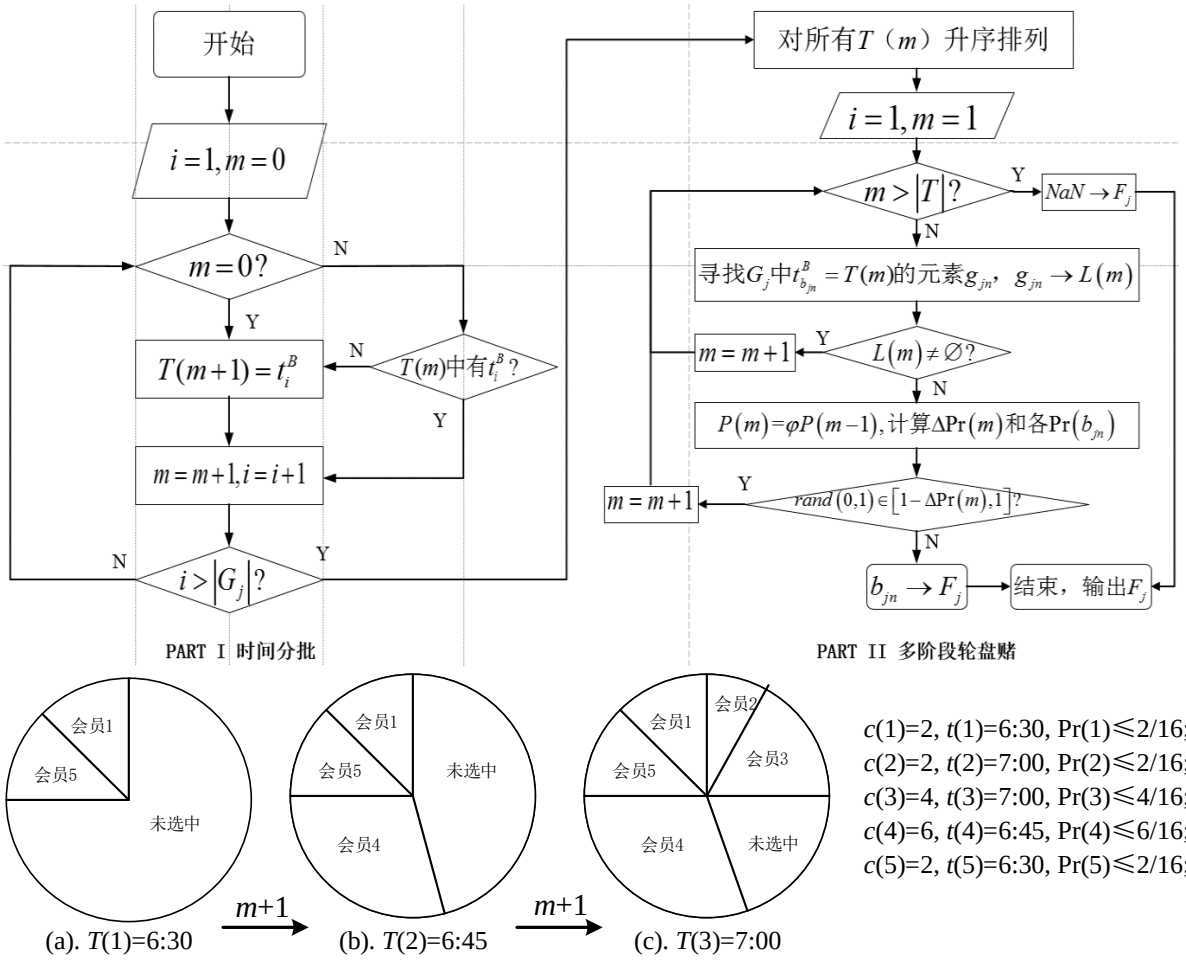


图 3.7 . 匹配流程和轮盘赌规则示意

3.2.5 求解结果分析

运用 Matlab2015b 编写二维多阶段轮盘赌求解算法（详见附件五），求得原题附件 1 中 835 个任务点的时空效率定价及成功执行的匹配会员编号，详细求解结果见附录 1。

在任务完成情况方面，任务未执行数量为 208 个，小于平台 APP 原始定价情况下的 313 个；新策略下，任务完成成功率为 75.1%，高于原方案的 62.5%。说明时空效率定价模型相比原定价策略，具有 20.2%的优化效果，一定程度上能提高匹配成功率。

在平台总定价方面，原方案所有任务总估价 57707.5 元，新方案为 58420.8 元，增长了 1.24%；原方案为已完成的任务花费 36446.0 元，新方案为 44166.4 元，增长了 21.2%，新方案定价稍有偏高。

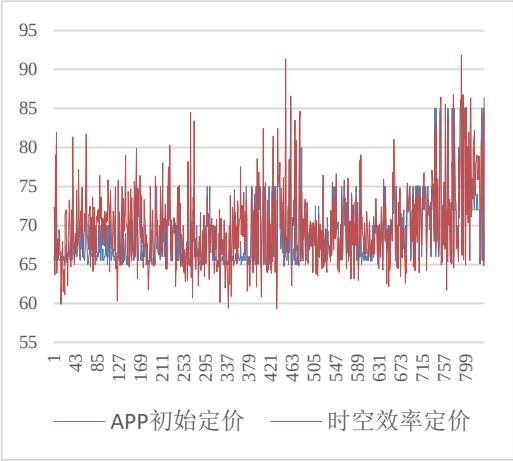


图 3.8　. 定价对比图

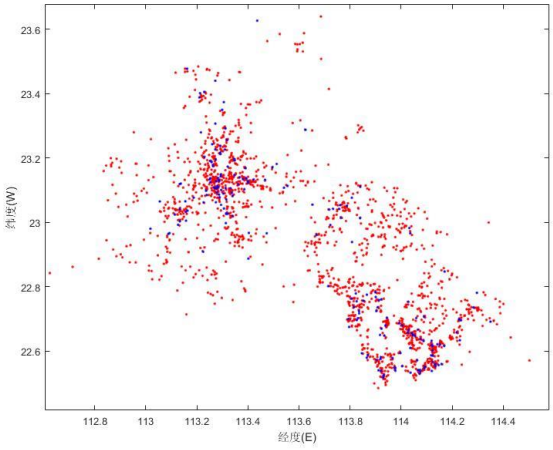


图 3.9　. 任务完成图

此外，分析图 3.8 中的全局定价折线图，可发现时空效率定价波动性更大。而将求解所得的任务完成地理分布情况（图 3.9）与 4.1 节图（蓝点为未完成任务，红点为已完成任务）进行对比，可发现广州市区与深圳市区的未完成任务数显著减少，可能由新策略根据地区经济情况提升基准定价导致；佛山市原方案中未完成任务几乎都被新方案完成；东莞市新出现零星未完成任务，属合理概率事件。

综上所述，新设计的时空效率定价模型能在附件 2 会员规模的情况下，有效提高附件 1 中任务的成功执行概率，但需要付出小幅上升的总体定价，用于激励经济较发达的大城市会员执行任务。

3.3 问题三模型的建立及求解

3.3.1　　问题分析

本题将多个位置比较集中的任务联合在一起打包发布。一方面，打包发布能够让平台效率更高，成本更低；但另一方面，如果打包发布的任务未被执行，则平台受到的损失也就更大。因此，平台在打包发布任务时，为了提高任务被执行的概率，会采取两方面的措施：1.会更加关注“守信用”的会员，即信誉度高的会员。2.分析打包规模与仍有预留任务限额的有效会员数的关系，寻找最佳的打包规模。

3.3.2　　会员信誉等级定价模型

1.数据预处理

首先，为了消除不同单位和量纲对结果的影响，我们将数据进行预处理。

（1）d_p 为同一类中的所有点到聚类中心 p 的平均距离，$d_p^{'}$ 为 d_p 均值方差处理后的距离数据

$$d_p = \frac{\sum_{j=1}^{n} d_{pj}}{n}, p = 1,2,3\ldots,|C| \tag{3.19}$$

其中

$$d_p^{'}=\frac{d_p-\overline{d_p}}{\sigma_i},p\in1,2,3\ldots,|C| \tag{3.20}$$

$$\overline{d_p}=\frac{1}{c}\sum_{i=1}^{c}d_p,\sigma_i=\sqrt{\frac{1}{c-1}\sum_{i=1}^{c}(d_p-\overline{d_p})},p\in1,2,3\ldots,|C| \tag{3.21}$$

（2） Dc_i 为时空可抵距离范围内会员 i 与聚类中心点之间的距离， $Dc_i^{'}$ 为 Dc_i 均值方差处理后的距离数据

$$Dc_i^{'}=\frac{Dc_i-\overline{Dc_i}}{\sigma_i},i\in1,2,3\ldots,|B|,p=1,2,3\ldots,|C| \tag{3.22}$$

其中

$$\overline{Dc_i}=\frac{1}{w}\sum_{i=1}^{w}Dc_i,\sigma_i=\sqrt{\frac{1}{w-1}\sum_{i=1}^{w}(Dc_i-\overline{Dc_i})},i\in1,2,3\ldots,|B| \tag{3.23}$$

2.新增变量分析

在问题二中，影响定价规律的因素主要有： 1.时空可抵距离范围内会员 i 与聚类中心点之间的距离 Dc_i；2.会员信誉等级值 Q_i^B；3.会员信息系数值 H_1， H_2；4.聚类中心的经济发展系数 K_p。

问题三在问题二基础上，考虑打包对原有定价体系的影响，主要体现在以下几个方面：

（1）同一类中的所有点到聚类中心 p 的距离总和 d_p。d_p 用于描述任务位置的离散程度。在评价规则中， d_p 越大，表明任务位置的离散程度越高，对会员来说，完成任务所需要付出的时间和空间成本越高，因此定价越高。

（2）分段函数 Q_i^B。Q_i^B 用于衡量会员信誉值对定价规则的影响。在平台打包发布任务时，会更加关注会员的信誉值，从而降低任务未被执行的风险。因此，相比于第二问中的静态等级划分，在第三问中，采用分段函数的方式，增强不同信誉等级用户的区分度。同时，信誉等级值的变化与打包规模 n 有关，打包规模 n 越大，定价规律对高信誉值的会员更有利。

$$Q_i^B=\begin{cases}S+k_1n,q_i^B>19.9231\\S+k_2n,q_i^B=19.9231\\S+k_3n,q_i^B<19.9231\end{cases} \tag{3.24}$$

式中， q_i^B 为会员 i 的信誉值， Q_i^B 为会员 i 的信誉等级值。$S=1$，为基准信誉等级值；k 为不同信誉等级的系数，$k_1=-0.1$，$k_2=-0.06$，$k_3=-0.02$；$n\in[2,6]$ 为打包规模。信誉度高的会员给予较高的报酬，而信誉度较低的会员报酬较少，从而保证当所有会员在平台竞争时，信誉度低的会员选中任务的概率远小于信誉度高的会员。

（3）打包规模n。在平台打包发布任务的情况下，打包规模与可参与执行任务的有效会员数之间存在如下关系：当打包规模增加时，打包后每个任务所含的任务数变多，对会员预定任务配额提出更高的要求。在这种情况下，有足够任务配额参与任务执行的会员就会减少。当有资格的会员数减少到一定程度，未被执行的任务数就会大大增加，从而促使平台缩小打包规模，让一些拥有较低任务配额的会员加入进来。从以上打包规模和会员数量的变化来看，存在一个让配置最为合理的供需平衡点。

在这个动态平衡的机制中，打包规模对会员数量的变化反应往往是滞后的，即第n次打包数Q_n^S取决于$n-1$会员数P_{n-1}；有效会员数对打包规模变动的反应是瞬时的，即第n期有效会员数P_n取决于本期打包规模Q_n。假设每次任务的发布最终都能被完成，即$Q_n^D = Q_n^S$。根据上述条件，可得如下供需平衡数学模型：

$$\begin{cases} Q_n^D = a - bP_n & (a>0, b>0) \\ Q_n^D = -c + dP_{n-1} & (c>0, d>0) \\ Q_n^D = Q_n^S \end{cases} \tag{3.25}$$

在式中，由于$b>d$，$\lim\limits_{n\to\infty} P_n = \dfrac{a+c}{b+d}$，即打包规模$P_n$是收敛的，故打包规模和有效会员数供求关系趋向于平衡点。

3.定价方案

根据各变量对定价规律的影响作用，可得在平台打包发布任务的情况下，新的定价方案如下所示：

$$P_{pj} = \frac{ne^{-n}}{2d_p^{'}} K_p Po_p (H_1 \sum_{i=1}^{i\in G_p^{low}} \frac{Dc_i^{'}}{Q_i^B} + H_2 \sum_{i=t+1}^{i\in G_p^{low} \cup i\in G_p^{up}} \frac{Dc_i^{'}}{Q_i^B}) \tag{3.26}$$

3.3.3 模型的求解

在第三问的求解中，求解步骤如下所示：

（1） 确定最佳打包规模n，并给出打包规模的上界。通过供需平衡数学模型计算$n\to\infty$时，P_n收敛于3，说明当打包规模分布在3时，是最为合理的，并将上界定为5，$n\in[2,5]$。

（2） 确定同一类中的所有点到聚类中心p的距离总和d_p。在会员执行任务时，过大的距离往往会超出会员的承受能力，造成任务执行率偏低。因此，我们将d_p的上界定为2km。

（3） 通过n和d_p的值确定打包方案，并通过MATLAB 2015b运行后（程序详见附录九）得到下（图3.10）结果：

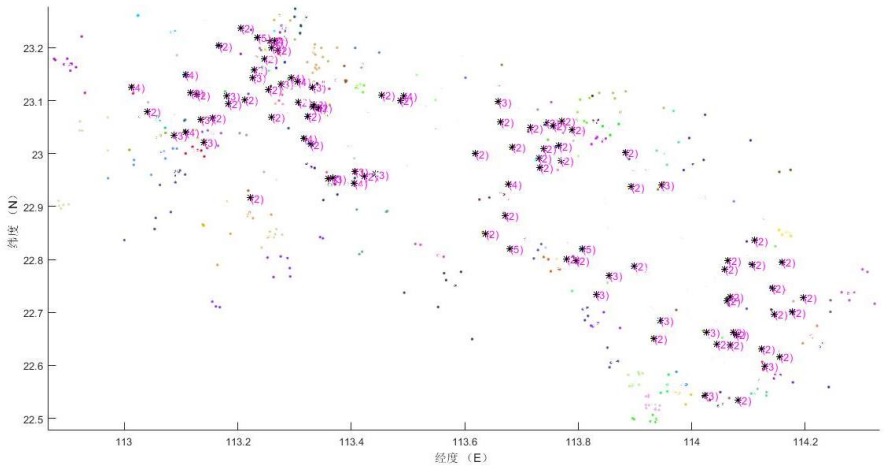


图 3.10 打包方案图

通过对打包方案图的分析，可以得到以下结论：1.打包后的任务点分布有明显的集群效应；2.未参与打包的任务点主要分布在偏远地区；3.打包后的任务规模以 2 个任务和 3 个任务为主.

（4）通过“二维多阶段轮盘赌”算法进行匹配（详见附件八），具体结果如下：

一共 692 个任务中，匹配成功的共有 564 个，配对成功率为 81.5%。比问题二中 74.9%的匹配成功率高出 6.5%。说明新的定价方案优于问题二中的定价方案。

在平台总定价方面，问题二中的定价方案所有任务为 58420.8 元，新方案为 60456.7 增长了 3.48%；原方案为已完成的任务花费 44166.4 元，新方案为 40345.9 元，降低了 8.6%，新方案收益较原方案更好。

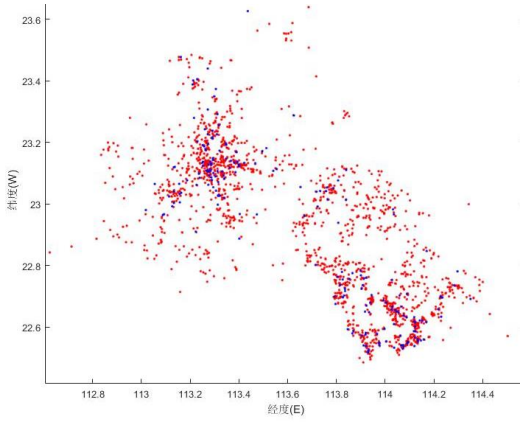


图 3.11 问题二匹配方案图

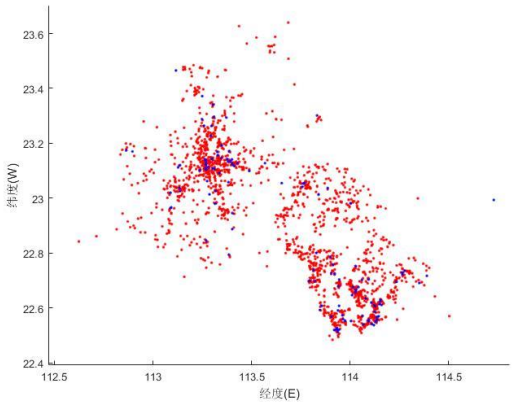


图 3.12 问题三匹配方案图

通过对匹配方案图分析可知：通过问题三的定价方案，热点区域未匹配的数量明显减少，匹配率明显上升。

3.4 问题四模型的建立及求解

问题四需要针对原文附件 3 中新的任务分布，给出与其相适应的定价方案。将新任务数据导入 Mapinfo Pro15.0, 得到其对应的空间分布。分析图 3.13 （黑点为新任务，黄点为会员）可得如下特点：

（1）任务分布呈现高度聚集性，主要集中在广州和深圳两个城市。

（2）任务数量众多，远超任务周边时空可抵距离范围内的会员数量。

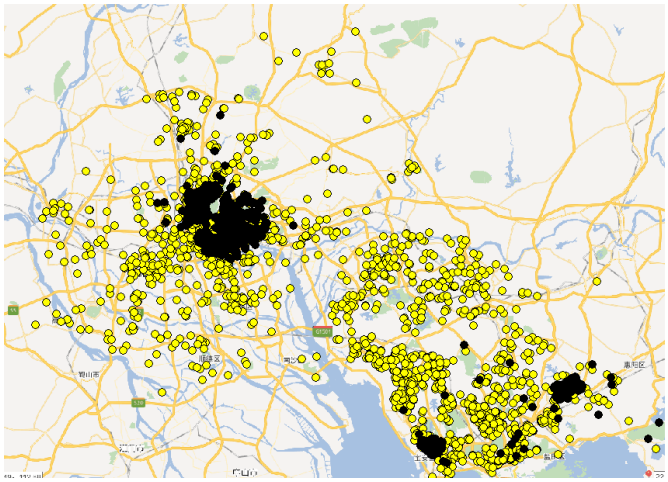


图 3.13 附件三项目位置分布图

因此，任务和会员间的匹配呈现“供大于求”的状况，此时会员可选择的任务数较多，其违约率往往较高，并且定价的主动性逐步由平台主导转向会员主导。针对大规模聚集态任务分布，任务打包定价发布仍是一种可供选择的定价方式。但这种方式在前节给出的打包定价方式下，往往找不到最优打包任务数，这可能由于存在如下的悖反规律：

（1）单次打包任务数上升时，由于会员配额限制，选择任务的准入门槛不断提高，由于任务周边时空可抵距离内有效会员样本缺乏，且具有高信誉度高配额的会员自身数量有限，任务难以匹配。

（2）单次打包任务数下降甚至不打包时，任务数量剧增，会员数量更少，更加难以匹配执行。

为保证项目被执行率，需要建立新的特殊定价模型，以适用于供大于求的高聚态极端任务情况。

3.4.1 歧视激励定价模型

极端环境下，考虑采取“价格歧视、高信誉度会员提价和低信誉度会员保留定价”的差异化定价策略，建立歧视激励定价模型，以保障具有高信誉高配额的会员所选任务的完成质量，并在宁愿损失部分任务完成率的基础上，逐渐淘汰低信誉度会员。定价模型由如下三部分组成：

（1）二级歧视定价函数

$$P_j=\begin{cases}\left(1+g\left(q_i^B\right)\right)P_{pj}, & q_i^B>19.9231\\ P_{pj}, & q_i^B=19.9231\\ P_0\left(q_i^B\right), & q_i^B<19.9231\end{cases} \tag{3.27}$$

（2）高信誉激励函数

$$g\left(q_i^B\right)=\frac{q_i^B}{\max\limits_{i=B}\left(q_i^B\right)} \tag{3.28}$$

（3）保留定价函数

$$P_0\left(q_i^B\right)=\min_{j\in A}\left(P_{pj}\right) \tag{3.29}$$

3.4.2 配额极值算法

依据高信誉会员最大化配额利用，低信誉会员概率淘汰，对二维多阶段轮盘赌参数做如下改进：

$$\Pr\left(b_{jm}\right)=\begin{cases}\dfrac{c_{b_{jm}}^{B}}{\sum_{m=1}^{m=|G_j|} c_{b_{jm}}^{B}}, & c_{b_{jm}}^{B}\geq n\\ 0, & c_{b_{jm}}^{B}<n\end{cases} \tag{3.30}$$

由此，将二维多阶段轮盘赌算法加入配额极值修正规则，使得配额有剩余的高信誉度会员选中任务的概率显著增大，并自然淘汰低信誉度或配额不足的会员，求得高聚态极端分布匹配结果。

3.4.3 求解结果分析

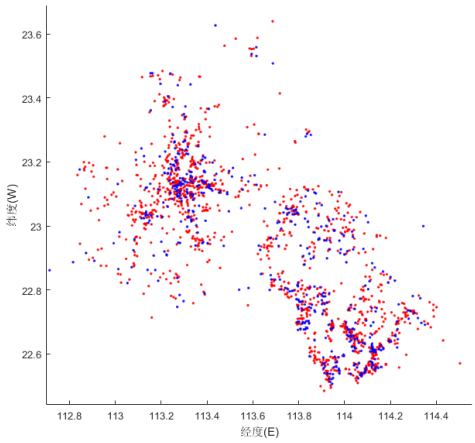


图 3.14 基于问题三模型求解结果图

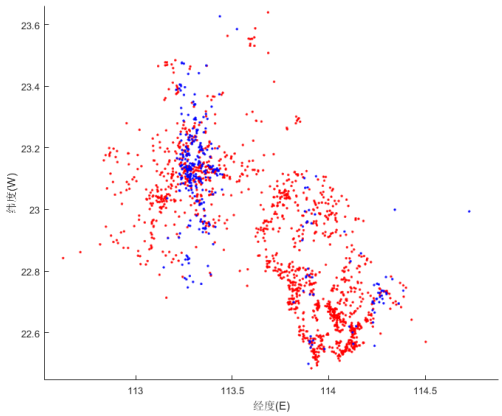


图 3.15 基于问题四模型求解结果图

运用 Matlab2015b 编写二维多阶段轮盘赌求解算法，求得原题附件 3 中 2066 个任务点的时空效率定价及成功执行的匹配会员编号。

在问题三定价模型求解结果中，2066 个任务点，其中有 717 个项目匹配成功，完成率为 34.7%；在问题四定价模型的求解结果中，有 969 个任务匹配完成，完成率为 46.9%，高于原方案的 12.2%。说明歧视激励定价方案相比原定价策略，具有 12.2%的优化效果，一定程度上能提高匹配成功率。

此外，在任务高聚集度的情况下，单纯的会员等级信誉定价模型并不能很好的提升任务的完成率，考虑采取“价格歧视、高信誉度会员提价和低信誉度会员保留定价”的差异化定价方案，不仅没有损失完成率，反而使得完成率上升了 12.2 个百分点。

第4章 模型的改进

在问题二中，我们构建了包括任务会员时间距离、空间备选会员集和会员信誉度的时空效率定价模型。该模型中的 D_{ij} 为一个不变量，但在实际情况中，在时空可抵范围内，会员 i 与任务 j 的距离是一个实时动态过程。即 $D_{ij}=F(t)$，是一个有关时间的函数，因此面对一个动态过程，要想模型具有前瞻性，需要加入实时动态预测，对于一个随机过程，我们可以使用平稳时间序列法作为预测

模型：

$$E(y_t) = E(y_{t+m})$$

$$\mathrm{cov}(y_t, y_{t+k}) = \mathrm{cov}(y_{t+m}, y_{t+m+k})$$

采用平稳时间序列法能够让平台不仅实时检测会员动态，更能对未来进行预测，使平台的做出更加合理的决策和匹配方案。

第5章 模型的评价与推广

5.1 模型的优点

1、本文创新的根据题目提出了“时空效率定价模型”，充分全面地考虑了定价影响因素考虑全面，而且定价稳定，适用于模型；

2、“依概率接受”，那么意味着每次都是按照分值的高低排序来选择，这样可能容易陷入局部最优，而无法获得全局最优解。使用了轮盘赌则可以更加修正陷入局部最优这个问题

2、利用 EXCEL 软件对数据进行处理作出了各种图表，使结论简便、直观、快捷；

3、运用了多种数学软件，取长补短，计算结果更加准确，清晰，可信度高；

4、本文建立的模型能结合实际情况对问题进行求解，实用性强，具有很好的推广性。

5.2 模型的缺点

1、“时空效率定价模型”在大规模下的应用有一定限制；

2、任务匹配模型是非线性模型，属于 NP-hard 问题，大型实例不能用精确算法求解，必须寻求这类问题的有效的近似算法；

5.3 模型的推广

本文建立的定价模型有效的提高了在原有定价规律下，任务被成功执行的概率。并创新性的“提出二维多阶段轮盘赌”算法，该算法运行效率高，准确性高，能够有效的解决移动互联网匹配效率平台，大大的降低了平台的实时性和运营成本。

参考文献

[1]乐琦，樊治平．基于累积前景理论的双边匹配决策方法[J]．系统工程学报, 2013, 28(1):38-46.

[2]陈鲤江，景程，吴姚鑫,等．数学表达式的归一化方法研究[J]．浙江工业大学学报, 2012, 40(2):229-232.

[3]王宝丽，杨琦峰．在线支付下个人信用等级评价指标体系[J]．中国水运:理论版, 2006, 4(4):170-171.

[4]《2016 年广东省各市 GDP 与人均 GDP 报告》

http://www.360doc.com/content/17/0419/14/502486_646832037.shtml

[5]许志凯，张宏莉，余翔湛,等．基于组合双向拍卖的物联网搜索任务分配机制[J]．通信学报, 2015, 36(12):47-56.

附录

附录 1：问题 2 的定价及匹配结果

任务编号	匹配会员	新定价	任务编号	匹配会员	新定价	任务编号	匹配会员	新定价
1	408	72.3	281	NaN	62.3	561	NaN	66.8
2	NaN	71.9	282	1642	64.9	562	545	72.1
3	NaN	63.7	283	8	68.6	563	1220	72.6
4	NaN	79.1	284	86	66.0	564	1222	75.9
5	550	71.0	285	89	71.7	565	294	71.2
6	1383	81.9	286	89	64.3	566	200	71.8
7	1064	63.9	287	32	65.0	567	1200	69.2
8	12	68.0	288	13	66.2	568	NaN	64.5
9	943	68.3	289	1273	68.0	569	NaN	70.9
10	1582	65.8	290	1196	69.4	570	111	67.1
11	1142	69.5	291	86	65.1	571	1253	76.1
12	1110	68.6	292	974	71.3	572	505	65.4
13	762	66.2	293	317	63.5	573	248	67.5
14	441	66.6	294	75	67.8	574	NaN	68.6
15	NaN	59.9	295	91	68.8	575	1174	70.6
16	NaN	63.5	296	NaN	65.1	576	259	68.8
17	1866	66.3	297	841	69.4	577	NaN	63.1
18	1859	67.9	298	1097	73.6	578	1427	74.2
19	NaN	61.5	299	588	69.9	579	808	69.1
20	NaN	62.9	300	586	71.5	580	259	67.6
21	NaN	64.3	301	602	68.7	581	1255	72.6
22	NaN	61.1	302	NaN	63.1	582	137	66.3
23	283	71.5	303	NaN	70.1	583	NaN	63.1
24	NaN	71.5	304	586	68.3	584	200	70.2
25	290	72.0	305	1421	67.8	585	NaN	69.9
26	436	65.3	306	NaN	69.0	586	505	67.4
27	1004	66.1	307	588	70.9	587	1727	73.6
28	NaN	62.3	308	257	69.4	588	180	67.5
29	NaN	68.6	309	305	66.2	589	1635	67.3
30	1613	68.7	310	632	66.3	590	36	71.4
31	204	73.3	311	586	68.7	591	NaN	63.0
32	199	69.9	312	665	63.9	592	96	65.9
33	12	66.2	313	NaN	69.9	593	1168	66.7
34	1410	64.7	314	29	67.3	594	1222	78.3
35	1145	72.0	315	1418	66.9	595	1174	67.8
36	74	65.7	316	304	70.8	596	1174	79.0
37	1163	74.3	317	882	64.1	597	NaN	69.6
38	375	81.3	318	1838	65.4	598	1234	71.2
39	142	69.7	319	109	64.7	599	NaN	68.9

40	106	67.4	320	677	65.3	600	1174	68.8
41	1116	65.2	321	NaN	64.7	601	1174	68.2
42	314	67.5	322	NaN	72.1	602	533	66.2
43	97	69.0	323	NaN	60.2	603	1166	70.3
44	NaN	66.4	324	584	66.6	604	1204	68.8
45	NaN	74.5	325	989	70.9	605	1166	68.1
46	1093	67.3	326	126	66.2	606	1193	66.4
47	977	69.5	327	NaN	67.1	607	261	71.8
48	NaN	68.1	328	109	68.2	608	1240	71.3
49	NaN	77.2	329	763	65.6	609	1240	68.8
50	135	70.6	330	596	66.2	610	NaN	66.0
51	114	68.7	331	951	70.7	611	255	66.9
52	1116	70.1	332	849	66.3	612	NaN	67.5
53	375	71.9	333	NaN	62.0	613	178	70.5
54	1135	71.8	334	985	64.4	614	164	66.3
55	NaN	68.2	335	718	69.1	615	146	65.4
56	667	74.9	336	303	65.0	616	137	67.2
57	52	66.6	337	1169	65.4	617	180	68.1
58	NaN	65.9	338	NaN	62.3	618	486	66.4
59	NaN	67.3	339	NaN	59.4	619	2	68.6
60	NaN	71.5	340	NaN	61.6	620	NaN	65.8
61	NaN	69.2	341	718	69.9	621	1094	71.1
62	1613	71.0	342	NaN	62.5	622	34	66.4
63	33	79.6	343	NaN	73.8	623	12	69.2
64	NaN	81.7	344	NaN	65.4	624	215	70.0
65	NaN	71.2	345	NaN	60.9	625	1071	73.4
66	NaN	68.7	346	366	66.9	626	329	70.9
67	478	68.1	347	59	72.3	627	114	68.7
68	492	66.9	348	132	71.0	628	499	65.9
69	151	71.4	349	1169	64.9	629	917	69.9
70	43	71.3	350	NaN	69.9	630	95	65.4
71	233	72.5	351	NaN	74.6	631	57	64.5
72	NaN	64.7	352	1224	67.0	632	197	70.2
73	1089	71.1	353	1680	71.6	633	616	67.3
74	17	68.0	354	249	67.3	634	343	65.9
75	657	66.8	355	1014	69.9	635	1106	69.6
76	1109	73.6	356	21	65.1	636	15	72.4
77	750	64.1	357	1033	73.1	637	94	68.4
78	43	67.9	358	NaN	72.2	638	514	64.5
79	209	72.3	359	92	67.1	639	1	64.3
80	NaN	72.0	360	NaN	69.8	640	157	75.9
81	NaN	65.0	361	NaN	70.8	641	1144	70.2
82	NaN	68.6	362	323	70.9	642	178	74.5
83	290	66.5	363	NaN	77.5	643	546	68.3

84	NaN	71.2	364	865	68.9	644	NaN	72.5
85	NaN	70.8	365	412	71.0	645	1396	68.1
86	1065	71.9	366	NaN	72.6	646	NaN	62.6
87	766	70.4	367	NaN	68.9	647	842	69.0
88	NaN	72.1	368	394	68.6	648	1148	66.9
89	386	68.0	369	NaN	74.2	649	546	69.5
90	1064	76.4	370	508	67.4	650	NaN	62.2
91	103	70.9	371	633	66.3	651	563	68.7
92	1388	72.6	372	132	71.7	652	NaN	64.2
93	426	73.6	373	16	68.6	653	NaN	65.2
94	NaN	69.4	374	16	73.3	654	294	70.9
95	1161	71.0	375	1116	70.9	655	545	70.6
96	42	67.5	376	NaN	61.6	656	1178	69.1
97	668	70.9	377	53	65.7	657	578	68.1
98	128	70.1	378	531	66.0	658	508	76.8
99	1118	67.4	379	637	65.7	659	NaN	73.3
100	1027	71.8	380	1267	65.4	660	1254	81.0
101	1014	68.1	381	NaN	62.2	661	261	76.8
102	795	71.4	382	214	66.7	662	77	76.7
103	512	70.3	383	NaN	68.9	663	96	66.3
104	851	71.7	384	586	73.5	664	1422	72.1
105	489	70.6	385	562	74.7	665	696	74.3
106	NaN	75.8	386	73	71.4	666	NaN	64.7
107	1129	72.7	387	54	64.0	667	24	70.0
108	79	72.0	388	312	71.8	668	NaN	64.5
109	52	64.1	389	280	65.8	669	1208	70.7
110	1142	72.5	390	344	72.8	670	957	73.0
111	1065	74.5	391	18	69.7	671	180	70.0
112	1093	69.7	392	29	66.5	672	261	74.8
113	485	67.9	393	1195	74.9	673	NaN	63.5
114	NaN	67.0	394	NaN	62.1	674	957	71.5
115	95	68.2	395	1522	78.6	675	NaN	67.1
116	1106	71.8	396	1422	65.3	676	727	65.8
117	NaN	71.2	397	NaN	65.0	677	1220	69.6
118	NaN	66.7	398	727	76.9	678	666	68.4
119	121	67.5	399	54	74.4	679	NaN	69.7
120	NaN	75.0	400	971	74.3	680	1223	67.9
121	100	64.5	401	905	74.1	681	1225	66.7
122	495	65.2	402	347	66.2	682	NaN	62.6
123	52	66.0	403	NaN	60.9	683	NaN	66.1
124	NaN	60.4	404	NaN	63.7	684	NaN	63.8
125	NaN	65.0	405	834	71.0	685	NaN	67.3
126	930	75.8	406	941	71.3	686	774	75.4
127	952	65.9	407	1208	82.4	687	1076	72.9

128	875	71.8	408	4	65.4	688	111	71.7
129	NaN	64.9	409	651	73.3	689	549	72.8
130	NaN	65.0	410	912	74.0	690	501	74.8
131	996	69.1	411	912	75.5	691	200	75.1
132	952	71.3	412	59	65.4	692	1193	67.5
133	NaN	72.5	413	86	65.2	693	NaN	71.0
134	NaN	74.1	414	297	64.0	694	378	68.2
135	1624	65.2	415	359	73.3	695	NaN	69.8
136	628	73.5	416	32	66.8	696	1055	68.1
137	86	69.2	417	32	64.3	697	850	73.9
138	8	71.3	418	NaN	71.1	698	NaN	64.7
139	841	73.7	419	32	65.9	699	509	69.9
140	NaN	79.0	420	32	70.2	700	930	72.8
141	40	70.5	421	770	64.3	701	486	64.2
142	32	70.0	422	378	67.5	702	546	73.5
143	393	69.6	423	8	63.9	703	NaN	68.9
144	952	65.4	424	136	74.5	704	1072	74.4
145	NaN	74.4	425	863	65.4	705	1026	67.9
146	517	72.5	426	183	81.5	706	1257	75.1
147	NaN	64.7	427	24	67.3	707	128	72.9
148	689	70.6	428	NaN	73.8	708	57	73.5
149	1267	71.9	429	NaN	73.6	709	38	64.0
150	967	74.7	430	NaN	64.0	710	1724	72.4
151	40	65.5	431	1279	73.2	711	1585	73.3
152	NaN	73.6	432	960	66.0	712	185	73.6
153	NaN	68.5	433	NaN	59.3	713	16	68.9
154	841	67.8	434	NaN	63.7	714	1	67.5
155	10	66.0	435	1216	82.4	715	NaN	71.1
156	554	65.9	436	773	75.4	716	345	74.0
157	651	75.7	437	1178	76.4	717	197	72.6
158	266	74.6	438	505	75.0	718	546	76.8
159	270	66.2	439	568	78.1	719	1603	72.7
160	996	74.9	440	NaN	69.6	720	1017	74.1
161	3	79.9	441	1158	70.0	721	150	74.4
162	NaN	68.7	442	977	74.8	722	543	73.5
163	NaN	63.2	443	NaN	72.4	723	46	64.2
164	NaN	74.8	444	1108	67.9	724	NaN	67.2
165	249	71.1	445	NaN	70.4	725	NaN	71.6
166	NaN	74.0	446	175	76.2	726	285	73.3
167	232	74.6	447	NaN	64.8	727	380	73.2
168	86	76.4	448	101	69.0	728	607	71.3
169	538	75.4	449	208	68.1	729	NaN	72.4
170	28	68.4	450	33	91.3	730	1225	72.0
171	232	73.8	451	NaN	68.5	731	NaN	72.8

172	249	67.7	452	1118	66.6	732	304	67.2
173	1421	69.7	453	62	68.6	733	658	73.2
174	NaN	64.6	454	NaN	66.6	734	1634	77.1
175	849	65.9	455	241	72.5	735	133	75.6
176	508	73.3	456	90	69.5	736	24	73.0
177	1037	71.5	457	568	78.4	737	200	75.3
178	91	70.5	458	99	68.2	738	14	69.1
179	957	67.1	459	830	77.5	739	NaN	82.0
180	NaN	68.8	460	510	86.6	740	NaN	74.4
181	1832	65.2	461	NaN	65.4	741	1	67.0
182	905	67.7	462	NaN	62.3	742	1112	83.2
183	1624	65.3	463	NaN	69.9	743	NaN	68.0
184	NaN	61.8	464	65	71.0	744	1029	72.0
185	NaN	63.2	465	NaN	70.0	745	210	76.0
186	714	65.1	466	345	73.3	746	1091	67.6
187	NaN	68.7	467	62	71.7	747	1	66.0
188	NaN	75.0	468	322	83.5	748	16	68.0
189	277	70.0	469	617	67.2	749	21	84.5
190	192	65.3	470	NaN	72.3	750	209	84.4
191	1320	65.5	471	108	80.9	751	671	86.5
192	NaN	72.0	472	18	66.8	752	386	66.3
193	3	70.6	473	771	67.5	753	1	65.2
194	905	66.3	474	1749	65.4	754	1039	65.6
195	NaN	64.8	475	NaN	65.4	755	299	72.6
196	13	66.6	476	1324	64.8	756	1026	67.9
197	181	69.0	477	1529	83.4	757	974	75.6
198	344	76.3	478	404	84.6	758	200	70.6
199	1794	74.2	479	435	65.4	759	1141	67.0
200	NaN	71.0	480	70	65.5	760	948	85.6
201	483	68.6	481	NaN	74.3	761	43	71.3
202	47	70.7	482	1240	71.7	762	NaN	61.7
203	NaN	66.6	483	241	65.7	763	148	71.7
204	1317	66.2	484	NaN	70.9	764	386	65.4
205	830	66.5	485	122	68.1	765	1331	83.9
206	902	69.8	486	182	69.1	766	NaN	70.6
207	NaN	65.6	487	1605	72.0	767	99	70.1
208	NaN	74.7	488	546	73.4	768	78	73.4
209	91	68.5	489	61	65.0	769	493	65.4
210	10	66.7	490	1240	70.3	770	514	65.3
211	508	71.6	491	1026	70.8	771	1102	72.9
212	NaN	78.1	492	1149	73.0	772	299	65.1
213	1874	69.2	493	NaN	65.2	773	NaN	78.1
214	1634	70.8	494	548	65.2	774	550	73.5
215	508	70.9	495	38	67.8	775	168	86.8

216	1208	69.7	496	930	66.9	776	139	64.6
217	133	72.8	497	1137	69.6	777	1258	83.0
218	109	64.5	498	382	70.5	778	1194	78.8
219	479	70.7	499	NaN	67.3	779	175	73.4
220	957	64.4	500	489	68.9	780	15	72.9
221	444	70.1	501	144	70.0	781	NaN	67.0
222	1477	74.1	502	1144	70.3	782	1105	67.3
223	NaN	77.5	503	NaN	64.0	783	210	68.9
224	1229	70.8	504	197	70.4	784	505	66.9
225	168	74.8	505	158	69.0	785	144	65.4
226	303	80.3	506	252	69.8	786	NaN	79.1
227	249	70.2	507	1123	67.4	787	1074	71.8
228	47	66.4	508	NaN	70.7	788	NaN	78.0
229	957	70.7	509	547	63.8	789	1217	86.1
230	108	65.4	510	769	70.6	790	501	84.6
231	NaN	68.3	511	241	66.8	791	1174	91.8
232	1181	69.2	512	NaN	63.5	792	1223	66.2
233	109	69.2	513	15	65.4	793	NaN	76.9
234	1181	71.0	514	294	71.5	794	507	86.8
235	1181	69.0	515	38	69.1	795	NaN	65.6
236	1181	70.3	516	336	65.8	796	533	85.0
237	NaN	62.2	517	107	70.2	797	NaN	81.2
238	NaN	64.7	518	298	67.0	798	1190	84.3
239	940	64.4	519	1058	69.4	799	NaN	65.0
240	282	71.7	520	NaN	65.3	800	1175	85.2
241	NaN	74.9	521	14	64.4	801	1174	71.4
242	505	71.2	522	NaN	66.3	802	62	80.2
243	428	69.7	523	841	76.5	803	1037	72.5
244	NaN	75.2	524	499	64.7	804	NaN	70.4
245	1181	68.3	525	1787	68.2	805	573	81.9
246	854	72.8	526	1097	65.3	806	1028	73.0
247	59	64.7	527	128	69.6	807	298	65.6
248	49	68.9	528	236	65.8	808	NaN	81.0
249	633	68.2	529	NaN	64.9	809	229	86.3
250	714	67.1	530	14	64.0	810	1166	71.1
251	NaN	64.7	531	NaN	67.7	811	1168	76.0
252	13	70.0	532	1036	69.2	812	NaN	75.0
253	1655	64.0	533	521	67.6	813	501	73.4
254	1169	67.0	534	1612	67.0	814	1227	80.2
255	73	64.0	535	547	66.0	815	1185	81.2
256	NaN	62.8	536	167	66.3	816	NaN	82.2
257	286	66.3	537	1111	65.3	817	1234	73.8
258	59	67.0	538	249	71.9	818	261	74.1
259	NaN	65.7	539	95	66.1	819	1637	77.9

260	NaN	62.9	540	NaN	66.7	820	96	78.4
261	NaN	78.2	541	1174	75.5	821	133	79.0
262	286	65.6	542	36	70.1	822	96	75.9
263	234	65.5	543	NaN	67.0	823	1193	77.2
264	53	65.8	544	200	73.4	824	111	75.9
265	526	64.8	545	1427	70.6	825	545	78.9
266	NaN	84.5	546	NaN	68.3	826	501	73.6
267	NaN	63.4	547	NaN	69.7	827	537	65.1
268	996	69.3	548	1295	76.6	828	15	67.0
269	317	68.9	549	232	68.8	829	36	80.9
270	NaN	60.8	550	77	68.6	830	NaN	80.9
271	277	73.6	551	NaN	64.3	831	1185	72.0
272	13	68.4	552	NaN	66.3	832	525	72.6
273	NaN	83.4	553	232	68.3	833	74	84.4
274	NaN	66.0	554	NaN	64.0	834	85	64.8
275	181	68.3	555	168	66.6	835	131	86.4
276	32	64.4	556	1178	65.5			
277	13	67.2	557	1194	70.2			
278	NaN	66.6	558	219	65.6			
279	59	66.2	559	164	73.7			
280	NaN	66.1	560	111	65.6			

附录二：问题一 一元函数线性回归程序

求出平均距离和定价的关系代码

```
%求出平均距离和定价关系矩阵
function av=aver(data,price)
price=sort(price);
averge=zeros(5,2);
m=size(data);
av=zeros(5,2);
for i=1:m(1)
    if(data(i,2)<=price(1))
        averge(1,1)=averge(1,1)+data(i,1);%累加长度
        averge(1,2)=averge(1,2)+1;%计数
    elseif(data(i,2)<=price(2))
        averge(2,1)=averge(2,1)+data(i,1);
        averge(2,2)=averge(2,2)+1;
    elseif(data(i,2)<=price(3))
        averge(3,1)=averge(3,1)+data(i,1);
        averge(3,2)=averge(3,2)+1;
```

```
        elseif(data(i,2)<=price(4))
            averge(4,1)=averge(4,1)+data(i,1);
            averge(4,2)=averge(4,2)+1;
        else
            averge(5,1)=averge(5,1)+data(i,1);
            averge(5,2)=averge(5,2)+1;
        end
        av(1,:)=[price(1),averge(1,1)/averge(1,2)];%求平均
        av(2,:)=[price(2),averge(2,1)/averge(2,2)];
        av(3,:)=[price(3),averge(3,1)/averge(3,2)];
        av(4,:)=[price(4),averge(4,1)/averge(4,2)];
        av(5,:)=[85,averge(5,1)/averge(5,2)];
    end
```

附录三：

按定价分类显示代码

```
%按定价分类显示
function classify(data)
hold on;
for i=1:833
    if(data(i,3)<=60)
        plot(data(i,2),data(i,1),'g.')
    elseif(data(i,3)<=65.5)
        plot(data(i,2),data(i,1),'b.')
    elseif(data(i,3)<=70)
        plot(data(i,2),data(i,1),'y.')
    elseif(data(i,3)<=75)
        plot(data(i,2),data(i,1),'r.')
    else
        plot(data(i,2),data(i,1),'k.')
    end
end
end
```

附录四：

画出拟合直线代码

```
%画出拟合直线
function R=compare(av,n)
hold on;
x=av(1:n,2);%自变量?
x=x';
y=av(1:n,1);%因变量
y=y';
R=corrcoef(x,y);%求出相关系数
b=polyfit(x,y,1);%线性拟合
z=polyval(b,x);
plot(x,z,'k-');
plot(av(:,2),av(:,1),'.b','MarkerSize',30);
plot(av(n+1:end,2),av(n+1:end,1),'.r','MarkerSize',30);
end
```

求出距离中心点长度代码

```
%求出距离中心点长度
function length=lens(data,center,index)
m=size(index);
length=zeros(m(2),3);
for i=1:m(2)
   length(i,1)=sqrt((data(index(i),1)-center(1))^2+(data(index(i),2)-center(2))^2);%求出长度并保存
      length(i,2)=data(index(i),3);%保存该点定价信息
      length(i,3)=data(index(i),4);%保存该点完成状态
end
%调用：
%length1=lens(oldtask,center(1,:),index(1,:))
%length2=lens(oldtask,center(2,:),index(2,:))
%length3=lens(oldtask,center(3,:),index(3,:))
%length4=lens(oldtask,center(4,:),index(4,:))
%length5=lens(oldtask,center(5,:),index(5,:))
%length6=lens(oldtask,center(6,:),index(6,:))
```

附录五：

```
%问题 2 代码%
%将附件 1 导入 matlab 中并取名为 task
```

使用 findcustomer 将每个 task 分配会员

使用 doublesystem 为确定每个 task 由哪个会员执行

对会员分组代码%

```
%findcustomer 用于在问题 3 中对会员分组%
function group=findcustomer(task,cust)
[a,~]=size(cust);
index=1:a;
index=index';
cust=[index,cust];%会员矩阵加编号
customer=sortrows(cust,2);%按经度排列
[n,~]=size(task);
group=zeros(n,6);
for i=1:n
    p1=max(find(customer(:,2)<task(i,2)));
    p2=min(find(customer(:,2)>task(i,2)));
    for j=1:3
        group(i,4-j)=customer(p1+1-j,1);
        group(i,3+j)=customer(p2-1+j,1);
    end
end
end
```

附录六

会员匹配任务代码

```
%doublesystem 用于在问题 3 中在每组会员中匹配一个任务%
function man=doublesystem(group,customer)
[~,b]=size(group);
sum=0;
g=zeros();
range=zeros(2,1);
if(b==1)
    man=group(1);
else
    for i=1:b
        if(customer(group(i),3)<=405)
            g(1,i)=group(i);
```

```
        elseif(customer(group(i),3)<=420)
            g(2,i)=group(i);
        elseif(customer(group(i),3)<=435)
            g(3,i)=group(i);
        elseif(customer(group(i),3)<=450)
            g(4,i)=group(i);
        elseif(customer(group(i),3)<=465)
            g(5,i)=group(i);
        else
            g(6,i)=group(i);
        end
        sum=customer(group(i),5)+sum;%求和
    end %分组
    [c,~]=size(g);
    for i=1:c%分组进盘
        temp=g(i,:);%第 i 组
        temp((temp==0))=[];%去 0；
        [~,m]=size(temp);
        for j=1:m%i 组进盘
            if(range(1,1)==0)
                range(1,1)=customer(temp(j),5)/sum;
                range(2,1)=temp(j);
            else
                endnum=range(1,end);
                per=(customer(temp(j),5)/sum);
                nu=per+endnum;
                range=[range,[nu;temp(j)]];
            end
        end
        ra=rand(1,1);
        if(ra<range(1,end))
            [~,index]=size(range);

                while(index>=1&&ra<range(1,index))
                    index=index-1;
```

```
                end

                man=range(2,index+1);
                return;%找到并退出
            end
        end
    end
    end
```

附录七

开始匹配代码

```
%开始匹配%
[a,~]=size(group);
mans=zeros();
for i=1:a
    mans(i)=doublesystem(group(i,:),customer);
end
```

运行代码

```
groups2=findcustomer(task,customer);
findtest;
xx=isnan(mans2);
aa=mans2(find(xx==0));
hold on;
plot(customer(:,1),customer(:,2),'r.')
plot(customer(aa,1),customer(aa,2),'b.')
mans22=mans2';
[n,m]=size(mans22);
dd=1:n;
dd=dd';
dd=[dd,mans22];%分配结果
```

附录八

%问题 3 代码%

对会员分组代码

```
%findcustomer 用于在问题 3 中对会员分组%
```

```
function group=findcustomer(task,cust)
[a,~]=size(cust);
index=1:a;
index=index';
cust=[index,cust];%会员矩阵加编号
customer=sortrows(cust,2);%按经度排列
[n,~]=size(task);
group=zeros(n,6);
for i=1:n
    p1=max(find(customer(:,2)<task(i,2)));
    p2=min(find(customer(:,2)>task(i,2)));
    for j=1:3
        group(i,4-j)=customer(p1+1-j,1);
        group(i,3+j)=customer(p2-1+j,1);
    end
end
end
```

会员匹配任务代码

```
%doublesystem 用于在问题 3 中在每组会员中匹配一个任务%
function man=doublesystem(group,customer)
[~,b]=size(group);
sum=0;
g=zeros();
range=zeros(2,1);
if(b==1)
    man=group(1);
else
    for i=1:b
        if(customer(group(i),3)<=405)
            g(1,i)=group(i);
        elseif(customer(group(i),3)<=420)
            g(2,i)=group(i);
        elseif(customer(group(i),3)<=435)
            g(3,i)=group(i);
        elseif(customer(group(i),3)<=450)
```

```
        g(4,i)=group(i);
    elseif(customer(group(i),3)<=465)
        g(5,i)=group(i);
    else
        g(6,i)=group(i);
    end
    sum=customer(group(i),5)+sum;%求和
end %分组
[c,~]=size(g);
for i=1:c%分组进盘
    temp=g(i,:);%第 i 组
    temp((temp==0))=[];%去 0；
    [~,m]=size(temp);
    for j=1:m%i 组进盘
        if(range(1,1)==0)
            range(1,1)=customer(temp(j),5)/sum;
            range(2,1)=temp(j);
        else
            endnum=range(1,end);
            per=(customer(temp(j),5)/sum);
            nu=per+endnum;
            range=[range,[nu;temp(j)]];
        end
    end
    ra=rand(1,1);
    if(ra<range(1,end))
        [~,index]=size(range);

            while(index>=1&&ra<range(1,index))
                index=index-1;
            end

        man=range(2,index+1);
        return;%找到并退出
    end
```

```
    end
end
end
```

开始匹配代码

```
%开始匹配%
[a,~]=size(group);
mans=zeros();
for i=1:a
    mans(i)=doublesystem(group(i,:),customer);
end
```

附录九

任务点打包代码

```
%将任务点打包
%result 任务结果
%sum 同一包内的距中心的距离和
%count 同一包内的任务个数
%center 任务中心

function [result,sum,count,index,center]=sumtopoint(newtask,n)
[index,center]=fcmanddraw(newtask,n);
sum=zeros(1,n);
count=zeros(1,n);
result=[];
for i=1:n;
    temp=tasklens(newtask,center(i,:),index(i,:));
    [~,b]=size(temp);
    count(i)=b;%个数
    if(b>1)
        for j=1:b
            sum(i)=sum(i)+temp(j);
        end
        if(sum(i)>0.022)%还原点
                plot(center(i,2),center(i,1),'*','color','w');
                dd1=index(i,:);
                dd1(dd1==0)=[];
```

```
                dd2=newtask(dd1,1);
                dd3=newtask(dd1,2);
                result=[result,[dd2';dd3']];
            else%取中心
                result=[result,[center(i,1);center(i,2)]];
                plot(center(i,2),center(i,1),'*','color','k');
                text(center(i,2)+0.0005,center(i,1)+0.0005,['(',num2str(b),')'],'color','m');
                plot(newtask(index(i,:)~=0,2),newtask(index(i,:)~=0,1),'.w');
            end
        else
            plot(center(i,2),center(i,1),'*','color','w');
            result=[result,[center(i,1);center(i,2)]];
        end
    end
end
```

求距离代码

```
%求出任务到任务中心点的距离
function length=tasklens(data,center,index)
index(index==0)=[];
m=size(index);
length=zeros(1,m(2));
for i=1:m(2)
    length(i)=sqrt((data(index(i),1)-center(1))^2+(data(index(i),2)-center(2))^2);
end
```