

# 基于吸引力均衡模型的“拍照赚钱”任务定价问题研究

## 摘要

“拍照赚钱”APP 是基于互联网的自助式劳务众包平台, 这种运营模式具有调查成本低、数据真实性高、调查周期短等众多优势. 本文通过建立数学模型, 就 APP 中的任务定价问题进行分析, 给出最优的任务定价方案.

针对问题一, 通过数据处理与相关性分析推定影响任务定价的因子, 利用已有的数据点进行多元线性回归分析, 确定原任务定价规律与任务完成难易程度和任务完成状况的关系, 其中任务完成难易程度包括会员密度和任务与会员聚点的距离. 会员密度越大, 任务与会员聚点的距离越小; 任务的完成率越高, 定价越低. 接着, 利用已完成任务的数据点进行多元线性回归分析, 与原定价规律对比分析, 发现任务未完成的主要原因是忽视会员信誉值等导致的定价过低. 最后, 通过计算原定价方案的吸引力“方差”发现吸引力的均衡性较差也是导致完成率较低的原因.

针对问题二, 将任务的定价与会员对该任务的期望的比值作为该任务对该会员的吸引力, 而为了避免任务不被接单, 本文考虑新的定价方案能够使得每一个任务对会员的吸引差别更小, 按照此原则建立了关于吸引力均衡性的优化模型. 将带有会员信誉权重的吸引力“方差”作为目标函数, 利用遗传算法搜索所有定价在  $[20, 120]$  之间的定价方案, 最后得出了“方差”为 110.46 的定价方案, 而在此方案下, 总体成本比原来有上升 32.05%.

针对问题三, 考虑到优质会员可以优先抢单的事实, 本文直接按照每个时间段能够参与抢单人数的比例来安排每个时间段发放的任务数量. 在任务发放过程中, 通过生成随机任务编号模拟随机选择任务发放, 会员通过吸引力最大值以及吸引力阈值来选择任务, 再通过生成  $0 - 1$  随机数并判定所在区间来模拟以会员信誉权重占比为概率派发任务. 考虑到会员是否按约完成任务受到多种因素影响, 本文则直接使用正态分布随机数来模拟会员所受到影响的大小. 最后通过 100 次模拟, 得出的平均成功率为 74.16%, 在原来的基础上提高了 11.65%.

最后, 再对随机因素, 吸引力阈值以及正态分布随机数的临界值进行灵敏度分析, 再对所建立模型的优缺点给出了客观的评价.

**关键字:** 拍照定价   多元回归   吸引力均衡   遗传算法   仿真模拟

# 目录

摘要	1
1. 问题重述	4
1.1 问题背景	4
1.2 问题提出	4
2. 问题分析	4
2.1 问题一	4
2.2 问题二	5
2.3 问题三	5
3. 假设与符号说明	5
3.1 假设	5
3.2 符号说明	6
4. 模型的建立与求解	6
4.1 问题一——原任务定价规律与任务未完成的原因分析	6
4.1.1 数据解读与统计分析	6
4.1.2 回归分析	12
4.1.3 原定价方案下任务对会员吸引力的均衡性分析	15
4.2 问题二——均衡性定价方案的建立与求解	15
4.2.1 模型二的建立	15
4.3 基于遗传算法求解	16
4.4 问题三——新定价方案任务完成率的预测与评价	19
4.4.1 仿真模型的建立	19
4.4.2 仿真模型求解	20
附录 A 模型一	21
附录 B 模型二中的代码	24
2.1 主程序	24
2.2 适应度函数	25
2.3 计算吸引力矩阵	26



# 1. 问题重述

## 1.1 问题背景

“拍照赚钱”这种服务外包的营业模式在借助智能手机和移动互联网的方式帮助消费者实现了自助服务. 一般情况下, 自助服务平台都会要求会员在下载指定的 APP, 并要求会员注册, 平台按照一定的规则向会员开放任务, 会员完成任务后可以领取一定的佣金. 这样的自助式劳务众平台能够帮助企业进行商务调查和信息搜索. 相比较传统的调查方式, 这类方式能够更好地保证数据的真实性. 但劳务众包平台存在的问题是, 任务的完成率不够高, 导致平台的运营成本远远大于企业的收益, 因此如何对平台上发放的任务进行合理定价以使得任务可以被尽可能地完成受到广泛的关注.

## 1.2 问题提出

重读“拍照赚钱”这个问题, 完成以下任务:

问题一: 研究附件一中的项目的任务定价规律, 得到已有的任务定价与任务完成状况和任务完成难易程度的关系, 并且要求计算在原来的定价方案下的任务完成率并且讨论任务对会员的吸引力的均衡性, 分析任务未完成的原因.

问题二: 在任务定价规律的基础上, 为项目设计一个新的定价方案, 定价方案应该满足在不提高平台的运营成本的前提下, 尽量提高任务的完成率, 考虑到定价方案对所有的会员具有一定的吸引力, 可考虑均衡性是最优的定价方案.

问题三: 在新的定价方案的情况下, 考虑通过随机仿真的方式, 模拟会员抢单, 在设定模拟规则时需注意: 1. 抢单既需要考虑优质会员的利益, 同时也需要对新会员保留一定的机会. 2. 任务是否完成还会受到店铺拒访、道路施工、天气不好、会员个人状态等随机因素的因素.

# 2. 问题分析

## 2.1 问题一

问题一要求研究附件一项目的任务定价规律, 并计算原任务定价方案对会员吸引力的均衡性, 分析任务未完成的原因. 首先对数据进行可视化处理, 将会员信息和任务信息分别标注在地图上. 对主要的指标进行基本统计分析和相关性分析, 对可能对任务定价产生影响的因子进行标准化处理, 并根据会员的位置分布进行 K-Means 聚类分析. 如何判断任务定价与这些因子的相关性是问题一的关键. 对此, 可利用所有数据点进行多元回归分析建立任务定价与影响因子的关系, 确定原任务定价规律. 探究任务未完成的原因, 可利用已完成任务的数据点进行回归分析, 与原定价规律对比, 分析二者的区别

以得到更为合理的定价策略. 最后, 利用吸引力均衡模型分析原定价方案对会员吸引力的均衡性.

## 2.2 问题二

问题二要求给出新的定价方案, 为了保证有较高的完成率, 则首先需要减少出现任务没有人接的情况, 即要使得每一个任务对会员都有一定的吸引力. 为此, 可以建立吸引力均衡模型, 通过每个任务对会员的吸引力的带会员信誉权重的总体“方差”来评判定价方案的均衡性, 再通过遗传算法来搜索可能的定价方案, 使“方差”达到最小, 从而可以得出最优的定价方案. 最后, 计算新方案所需要的成本, 再与原来的方案进行对比.

## 2.3 问题三

问题三要求预测新模型的任务完成率, 这需要首先制定派单以及抢单的规则. 考虑按不同时间段可以参与抢单的人数的比例随机派发任务, 再考虑会员每次抢当前派发任务中吸引力最高并且超过吸引力阈值的任务, 对于多个会员抢一个任务, 则按照信誉权重随机派发给某一位会员. 最后考虑被接单的任务能否完成所收到的各种影响. 其中随机选择任务派发以及随机选择会员派发的过程可以利用均匀分布随机数进行模拟, 被接单任务是否能完成受到的影响则可以利用正态分布随机数进行模拟. 最后, 计算出任务完成的比例, 并与附件一中的数据进行对比.

# 3. 假设与符号说明

## 3.1 假设

1. 假设会员位置的整体分布不发生显著变化.
2. 不考虑 APP 中任务描述等对用户是否接受任务的影响.
3. 每一个时间段内会员仅能抢单一次, 并且在开始派单时所有可以抢单的会员同时抢单.

为了保证给新会员留有一定的机会, 同时能够合理减少计算量. 所以我们假设会员 3 分钟内仅能抢单一次, 并且能抢单的会员可以同时抢单.

4. 假设会员在预定任务时, 优先预定对其吸引力最大的任务.

### 3.2 符号说明

符号	含义
$\rho_i$	第 $i$ 个任务的会员密度
$Den_i$	第 $i$ 个任务的相对会员密度
$\mu'_j$	第 $j$ 个会员的相对信誉值
$Rep_i$	第 $i$ 个任务的相对信誉值
$Dis_i$	第 $i$ 个任务点到最近会员聚类中心的距离
$X_i$	第 $i$ 个任务的完成情况
$x_i$	第 $i$ 个任务的定价
$r_{ij}$	会员 $j$ 到任务 $i$ 的距离
$\omega_j$	第 $j$ 个会员的信誉权重
$B$	定价方案的吸引力的“方差”

## 4. 模型的建立与求解

### 4.1 问题一——原任务定价规律与任务未完成的原因分析

为了分析得到原任务定价规律与任务未完成的原因,本文首先对附件一和附件二中的数据进行数据可视化与统计分析,确定任务定价的主要影响因子,再建立多元回归模型和吸引力均衡模型分析定价规律与任务未完成的原因.

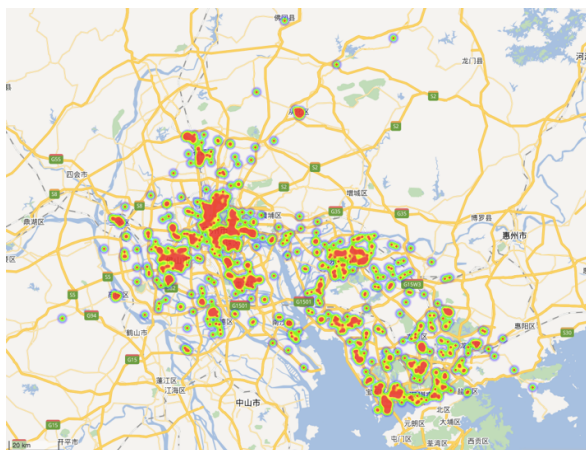
#### 4.1.1 数据解读与统计分析

##### 1. 数据可视化

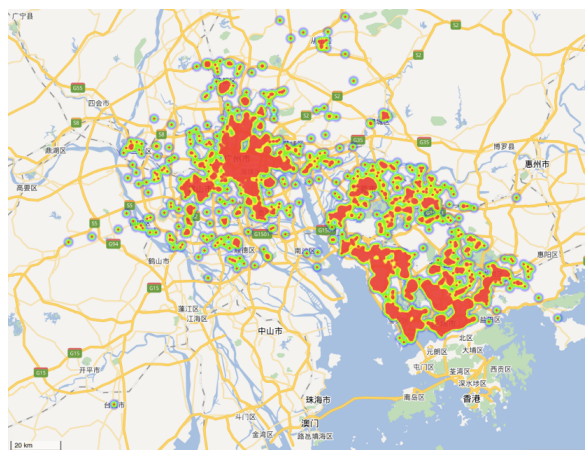
将附件一的任务坐标数据和附件二的会员坐标数据通过“地图无忧”网页建立图层,并导入百度地图中进行标注.任务数量热力图和会员分布热力图如图 1 所示.

根据该项目的每个任务在地图上的分布状况以及其各区描述性统计数据表可以观察到的信息有:

1. 任务大多集中分布在广东省的广州市、深圳市、东莞市以及佛山市等城市的繁华地带,在这些区域人流量较大,交通便利,会员分布也较为密集.



(a) 任务数量热力图



(b) 会员数量热力图

图1 热力图

表1 各区描述性统计数据表

城市	任务数	完成率	会员数	会员分布	价格均值	价格方差	GDP	平均工资
深圳	163	20.86%	629	3.84	67.61	23.19	2.42 亿	5199
广州	319	60.82%	666	2.09	68.07	13.18	2.29 亿	4811
佛山	174	66.09%	216	1.23	71.44	31.2	0.99 亿	3568
东莞	178	100.00%	351	1.98	70.34	21.01	0.83 亿	3845

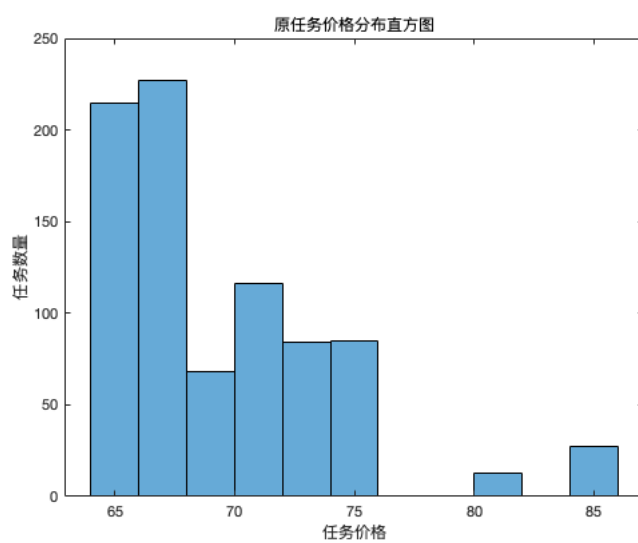


图2 原任务定价分布直方图

2. 较少的任务分布在远离市区的郊区, 在这些区域, 任务分布稀疏, 人口密度与人流较小, 交通不便, 会员数量也较少.

3. 原任务定价分布在  $[65, 85]$  的区间内, 其中价格更多地集中在  $[65, 70]$  的区间内, 呈偏态分布.

4. 各城市间任务完成率相差较大, 其中深圳市任务完成率最低仅为 20.86%, 东莞市任务完成率最高, 接近 100%. 结合各市 GDP 和平均工资数据中可以发现, 经济发展水平相对更高的城市任务完成率反而较低, 这可能是因为较发达的城市人们生活节奏快, 工资高, 空闲时间少, 任务定价吸引力低.

## 2. 数据标准化

为确定已结束项目的定价规律, 需要找出该项目中影响定价的因素及各因素的影响程度. 首先对任务点附近区域会员的数量、任务点附近区域会员的信誉值、任务点到会员的距离、任务的完成状况等指标标准化.

### 1. 相对会员密度

定义第  $i$  个任务点的会员密度  $\rho_i$  为以该点为中心  $0.05^\circ$  对应地理长度为半径的范围内的会员数. 对  $\rho_i$  进行如下的尺度变换, 得到第  $i$  个任务点的相对会员密度  $Den_i$ :

$$Den_i = \begin{cases} \log_{10} \rho_i + 3, & \rho_i \neq 0; \\ 0, & \rho_i = 0. \end{cases}$$

用相对会员密度  $Den_i$  衡量第  $i$  个任务点附近区域会员的数量.

### 2. 会员相对信誉值

定义第  $j$  个会员的信誉值为  $\mu_j$ , 第  $i$  个任务点受信誉值影响的范围是, 以该点为中心  $0.05^\circ$  对应地理长度为半径的圆形区域. 由于会员的信誉值跨度很大, 因此对范围内每个会员的信誉值进行如下的尺度变换, 得到会员相对信誉值  $\mu'_j$ :

$$\mu'_j = \log_{10} \mu_j + 3$$

定义范围内所有会员的编号分别为  $a_1, a_2, \dots, a_n$ , 对范围内所有会员的会员相对信誉值取平均, 得到任务点的相对信誉值  $Rep_i$ :

$$Rep_i = \frac{\sum_{j=1}^n \mu'_{a_j}}{n}.$$

用相对信誉值  $Rep_i$  衡量第  $i$  个任务点附近区域会员的信誉值.

### 3. 任务点到最近会员聚类中心的距离

为衡量任务点到会员的距离, 首先使用 K-means 聚类算法对会员的地理位置进行聚类. 从会员地理坐标分布图中可以看出, 会员绝大部分位于佛山、广州、东莞、深圳四座城市. 因此取  $K = 4$ , 聚类结果如图 3 所示.



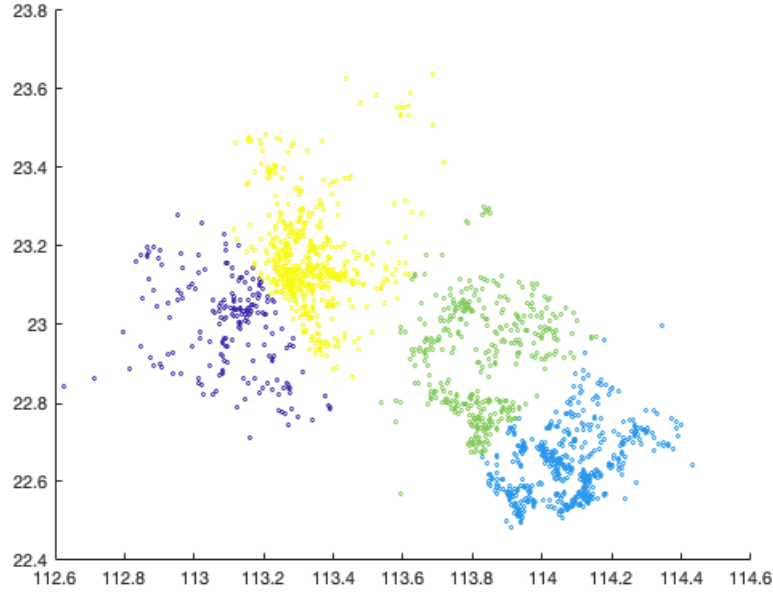


图3 会员聚类示意图

根据会员分布划分 4 个区域：紫色区域主要覆盖佛山市；黄色区域主要覆盖广州市；绿色区域主要覆盖东莞市；蓝色区域主要覆盖深圳市. 四个区域的中心点分别为  $O_1, O_2, O_3, O_4$ . 四个中心点的地理位置见下表.

表2 聚类中心坐标

	$O_1$	$O_2$	$O_3$	$O_4$
经度	113.1031	113.3335	113.8242	114.0686
纬度	22.9975	23.1644	22.9132	22.6415

由于聚类的分界与城市的分界基本一致，而会员通常不会跨城市完成任务，因此用任务点到最近会员聚类中心的距离衡量任务点到会员的距离为  $Dis_i$ .

#### 4. 任务的完成情况

定义第  $i$  个任务的完成情况为  $X_i$ ，当任务完成时  $X_i = 1$ ，当任务未完成时  $X_i = 0$ .

### 3. 相关性分析

#### 1. 会员信誉度与会员限额的关系

由于会员信誉度与会员限额均为连续变量，且观察原始数据可知其相关性并不为线性，故本文采用斯皮尔曼相关系数来刻画会员信誉度与会员限额两个数组的顺序相关性。

性.

$$\rho_s = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2 - 1)}.$$

利用 SPSS 计算两数组的斯皮尔曼相关性系数得到  $\rho_s = 0.744$ ，在 0.01 级别（双尾）相关性显著，为较强相关. 说明商家确定会员任务限额时较多地考量了会员的信誉度.

## 2. 任务定价与相对会员密度的关系

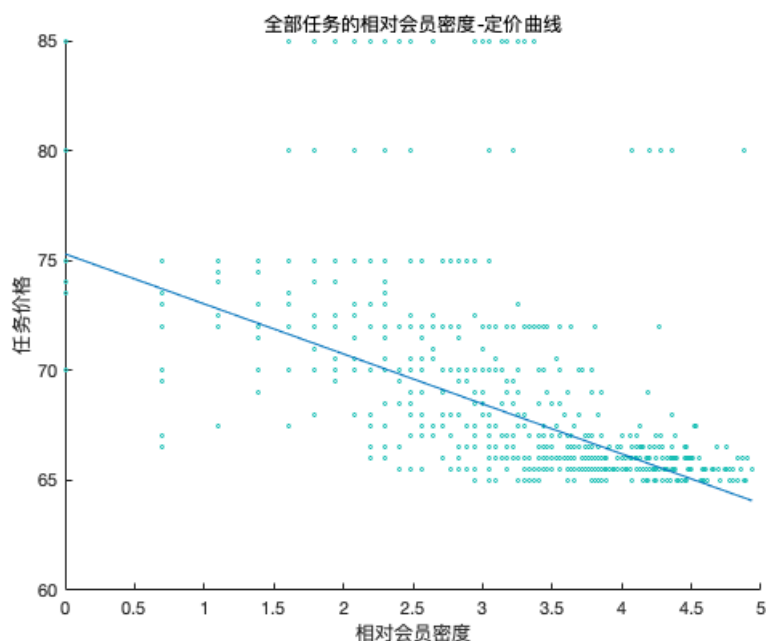


图 4 任务价格与相对会员密度散点图

由相对会员密度与任务定价的散点图和拟合直线可看出，相对会员密度与任务定价基本呈负相关关系. 利用 SPSS 计算两数组的斯皮尔曼相关性系数得到  $\rho_s = -0.730$ ，在 0.01 级别（双尾）相关性显著，为较强相关.

相对会员密度反映了会员在某一区域内的集中情况. 相对会员密度越大，会员集中程度越高，该任务被完成的可能性越大，此时该任务的定价就越低，则相对会员密度  $Den_i$  与任务定价呈负相关.

## 3. 任务定价与任务到聚类中心最短距离的关系

由图 5 可以发现，任务与聚类中心距离越小，任务所在区域会员的分布越密集，周围会员数量越多，任务的定价越低；任务与聚类中心距离越大，任务所在区域会员的分布越稀疏，周围会员数量越少，任务的定价越高. 将任务的发布看做需要被完成的任务需求方，会员的人数看做完成任务的供给方. 在交通便利的市区，会员人数大于任务的数量，供大于求，任务定价较低；当供小于求时，任务定价较高. 通过分析任务定价与任务到聚类中心最短距离的关系得到的定价规律符合经济学的供求定理.

## 4. 任务定价与任务完成率的关系

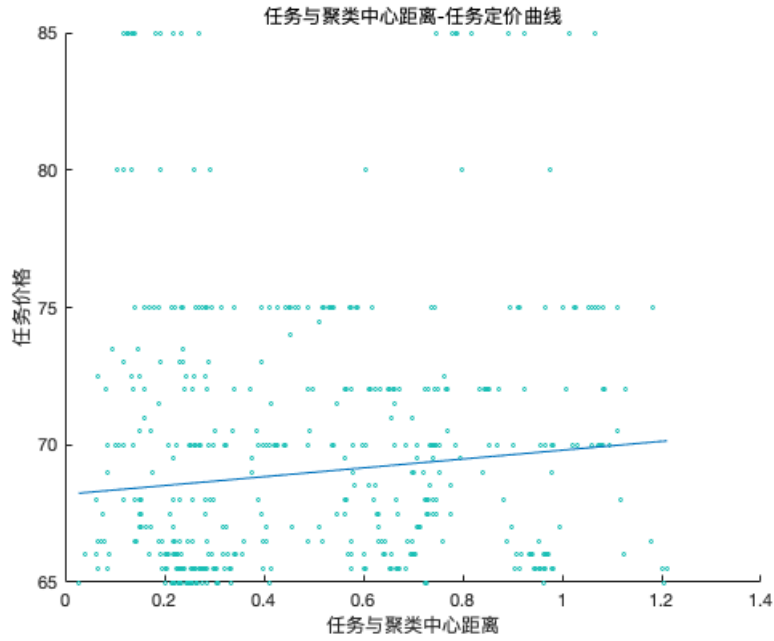


图 5 任务价格与任务到聚类中心最短距离散点图

首先对每一个任务标价计算任务完成率，如下表所示。

表 3 任务完成率统计表

任务标价	任务总数	完成数	完成率	任务标价	任务总数	完成数	完成率
65	65	35	0.5385	71.5	4	3	0.75
65.5	150	76	0.5067	72	5	4	0.8
66	103	46	0.4466	72.5	60	42	0.7
66.5	63	35	0.5556	73	9	7	0.7778
67	38	18	0.4737	73.5	10	7	0.7
67.5	23	17	0.7391	74	5	3	0.6
68	30	25	0.8333	74.5	5	1	0.2
68.5	11	6	0.5455	75	2	1	0.5
69	19	12	0.6316	80	78	59	0.7564
69.5	8	6	0.75	85	13	9	0.6923

由图 6 可知，任务定价与任务完成率具有一定的线性相关性，即任务的标价越高，任

任务标价	任务总数	完成数	完成率	任务标价	任务总数	完成数	完成率
70	96	77	0.8021	71.5	27	24	0.8889
70.5	11	9	0.8182	平均 69	835	522	0.6251

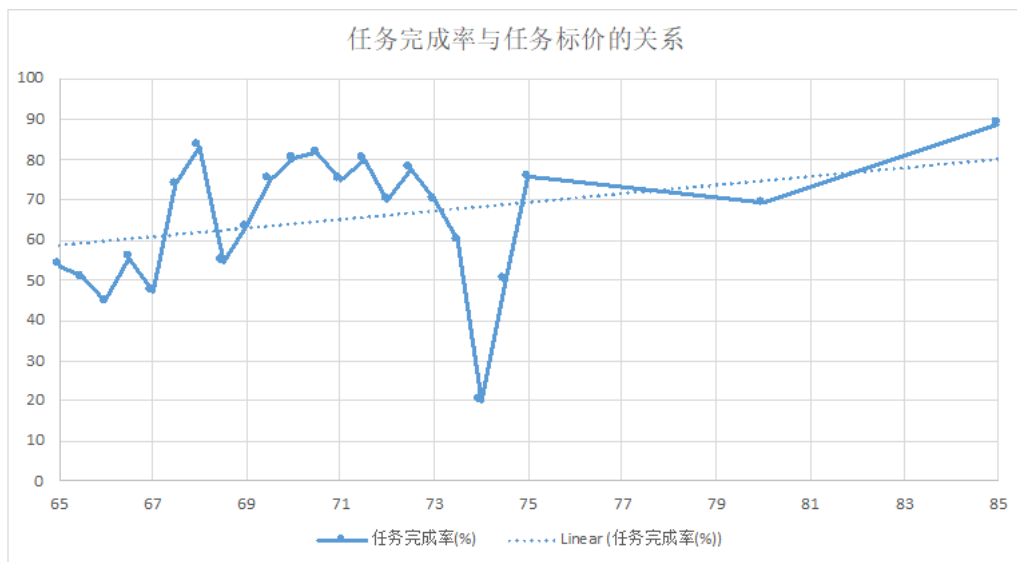


图 6 任务定价与任务完成率的关系

任务完成率相对越高. 计算可得任务定价与任务完成率的 Pearson 线性相关系数为 0.309. 导致相关系数较低的原因可能是任务标价对应的任务数量的分布不均匀. 其中当任务标价为 {74, 74.5, 75} 时, 任务总数极少, 导致计算得到的任务完成率并不具有代表性.

#### 4.1.2 回归分析

根据以上分析, 定价规律中影响任务定价的四个因子分别为相对会员密度  $Den_i$ 、任务点的相对信誉值  $Rep_i$ 、任务点到最近会员聚类中心的距离  $Dis_i$  和任务的完成情况  $X_i$ . 定义第  $i$  个任务的定价为  $x_i$ . 以上面四个因子作为解释变量, 提出原任务定价的多元线性回归模型如下所示.

$$x_i = \beta_0 + \beta_1 Den_i + \beta_2 Rep_i + \beta_3 Dis_i + \beta_4 X_i + u_i.$$

使用附件一中的所有数据点进行回归分析得到如表 4 中所示的结果:

表 4 所有数据点回归分析参数表

$\beta_0$	$\beta_1$	$\beta_2$	$\beta_3$	$\beta_4$
75.1705	-2.219	-0.0556	1.266	0.5361

原项目定价规律为：

$$x_i = 75.1705 - 2.2190Den_i - 0.0556Rep_i + 1.2660Dis_i + 0.5361X_i. \quad (1)$$

根据回归结果可以得出任务定价与任务完成难易程度和任务完成状况的关系. 其中任务完成难易程度由任务点的相对会员密度  $Den_i$  和任务点到最近会员聚类中心的距离  $Dis_i$  共同决定. 任务点的相对会员密度越大, 到最近会员聚类中心的距离越小, 任务越容易被完成, 任务定价相对越低; 任务的完成率越高, 任务定价越高.

其中相对会员密度  $Den_i$  的回归系数  $\beta_1$  在 99% 的水平上显著, 而相对信誉值  $Rep_i$  的回归系数  $\beta_2$  不显著. 说明定价规律主要考虑了任务的相对会员密度, 对任务的相对信誉值基本没有考虑.

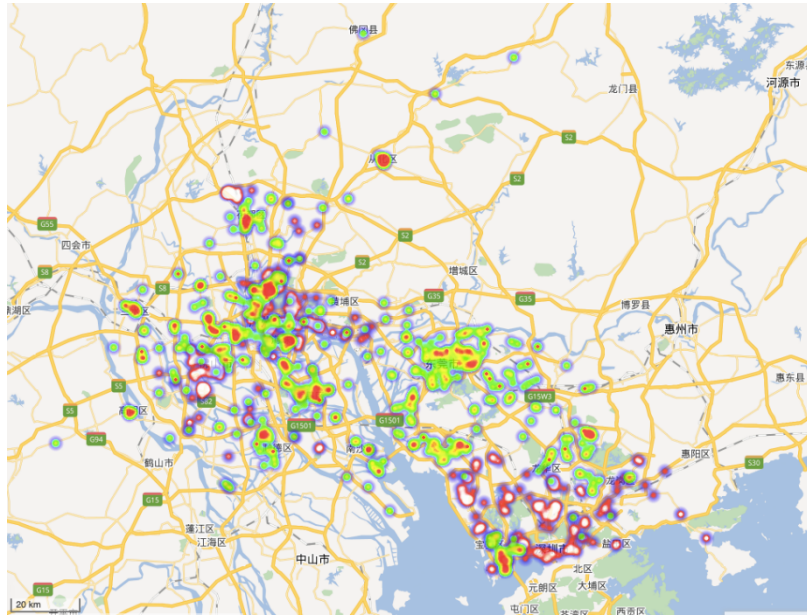


图 7 任务分布热力图

图 7 和图 8 分别表示任务分布热力图和会员分布热力图. 图 7 中, 红-绿过渡的热点表示已完成的任務, 白-紫过渡的热点代表未完成的任務, 热点越大, 任务定价越高. 图 8 中, 热点越大, 会员相对信誉值越高.

观察任务分布热力图可以发现未完成的任務（白紫色热点）大都定价较低（半径较小），则定价低可能是任务未完成的原因之一. 进一步对比两张地图，以地图右下方深圳地区与东莞地区为例进行比较. 在会员分布热力图中深圳的会员密度（热点密度）明

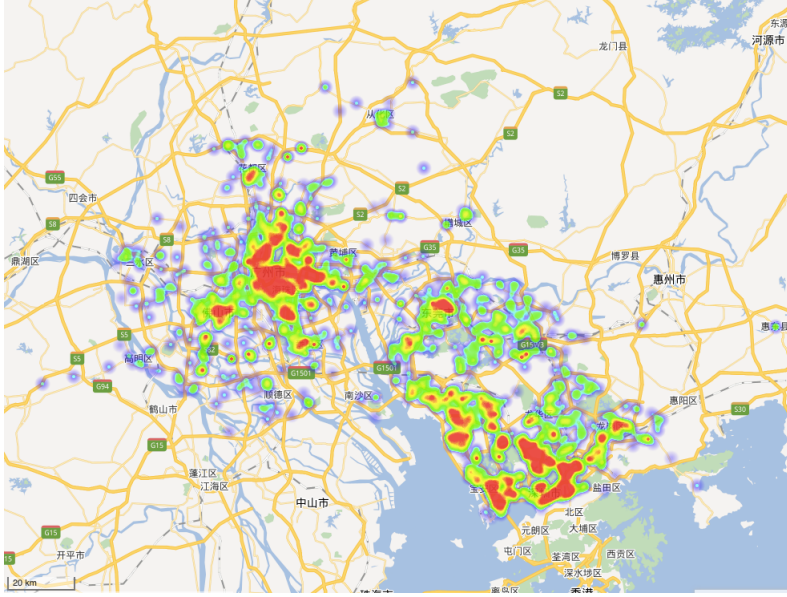


图 8 会员分布热力图

显高于东莞，会员信誉值（热点颜色深度）也高于东莞，根据经验判断深圳的任务完成情况应该也由于东莞，但事实恰恰相反. 深圳市的任务完成率 20.86% 远远低于东莞市的 100%. 原因就在于深圳市的任务定价（热点大小）低于东莞市，深圳市任务的均值为 67.61，而东莞市的任务均值为 70.34. 分析佛山市禅城区及广州市天河区附近的任务未完成的情况可以得到同样的结论. 因此，原任务未完成的主要原因之一是定价过低.

进一步分析定价过低的原因，仅利用任务完成的数据点回归如下所示的多元线性回归模型.

$$x_i = \beta_0 + \beta_1 Den_i + \beta_2 Rep_i + \beta_3 Dis_i + u_i.$$

表 5 已完成任务数据点回归分析参数表

$\beta_0$	$\beta_1$	$\beta_2$	$\beta_3$
77.8338	-2.1733	-0.1182	0.9697

则可以得到如下方程:

$$x_i = 77.8338 - 2.1733Den_i - 0.1182Rep_i + 0.9697Dis_i. \quad (2)$$

比较回归 (1) 和回归 (2) 中  $\beta_2$  参数值的差异，发现回归 (1) 中任务点相对信誉值的权重仅为回归 (2) 中的约二分之一，这说明原定价策略对任务点相对信誉值的考虑明显不足. 根据实际经验当会员密度较大时应当适当降低定价；任务点的相对信誉值较高时应当适当提高定价. 原定价规律正是因为过多地考虑了会员密度的影响，而忽略了任务点相对信誉度的影响，所以对这些任务点定价过低，导致任务未能被完成.

除此之外也存在少数任务定价相对合理但也被完成的情况,可能是因为这些任务点过于偏远、交通不便导致会员不愿意前往完成任务.

### 4.1.3 原定价方案下任务对会员吸引力的均衡性分析

每个任务对会员的吸引力的均衡性的含义是指每一个任务对同一个会员的吸引力是相等的. 当会员按照优先预定对其吸引力最大的任务的规则去预定任务时, 对任务标的的偏好相同, 吸引力的均衡性保证了会员对任务的随机选择和完成率.

本文利用带有权重的“方差”来评判定价方案的吸引力均衡性:

$$B = \sum_{j=1}^m \omega_j \left\{ \sum_{i=1}^n \left( \frac{\frac{x_i}{f(r_{ij})}}{\sum_{k=1}^n \frac{x_k}{f(r_{kj})}} - \frac{1}{n} \right)^2 \right\}.$$

利用附件一数据计算得到的吸引力“方差” $B=120.9527$ . 说明原定价方案对会员吸引力的均衡性较差. 原方案未考虑定价对会员吸引力的均衡性可能是任务完成率较低的原因.

(吸引力均衡性模型的构建详见 4.2.1 模型二的建立)

## 4.2 问题二——均衡性定价方案的建立与求解

### 4.2.1 模型二的建立

问题二要求重新给附件一中的项目设计任务定价方案. 而任务定价方案则需要考虑到成本和完成率这两个因素. 但是考虑到不能完成任务可能会对企业带来损失, 于是模型中我们优先考虑提高任务的完成率.

为了保证每一个任务尽可能被完成, 首先考虑使更多的任务能够被会员所领取, 因此需考虑每个任务对会员的吸引力的均衡性 [2], 即尽可能使每一个任务对同一个会员的吸引力是相等的. 也就是说, 如果所有任务对一个会员的吸引力的总和为 1, 则每一个任务的吸引力的期望是  $\frac{1}{n}$ . 设任务  $i$  的定价为  $x_i$ , 会员  $j$  对任务  $i$  的价格期望只与两者距离有关的, 则可以记为  $f(r_{ij})$ , 那么任务  $i$  对会员  $j$  的吸引力可通过比值体现为  $K \frac{x_i}{f(r_{ij})}$  (其中  $K$  为归一化单位). 则对会员  $j$  而言, 任务  $i$  对其吸引力占比为  $K \frac{x_i}{f(r_{ij})} / \sum_{i=1}^n K \frac{x_i}{f(r_{ij})}$ . 于是可以用带有权重的“方差”来评判定价方案的吸引力均衡性:

$$B = \sum_{j=1}^m \omega_j \left\{ \sum_{i=1}^n \left( \frac{\frac{x_i}{f(r_{ij})}}{\sum_{k=1}^n \frac{x_k}{f(r_{kj})}} - \frac{1}{n} \right)^2 \right\}.$$

其中  $\omega_j$  表示会员  $j$  的信誉权重,  $B$  越大则表示各个任务的吸引力的差别越大,  $B$  越小则表示各个任务的吸引力差别越小. 为了方便, 我们这里暂且就将  $B$  称为“方差”. 考虑到各个会员信誉值的极差太大 (最大为 67997.39, 最小为 0.0001), 故我们通过以下方式来确定



义会员的信誉权重:

$$\omega_j = \log_{10} Rep_j + 5.$$

其中  $Rep_j$  表示第  $j$  个会员的信誉值, 这样使得所有的会员信誉值权重为正并且信誉值最小的那位权重为  $\omega_{1877} = 1$ .

再根据文献 [2] 中所说, 参照出租车的交通补贴模型, 我们可以简单的认为会员  $j$  对任务  $i$  的定价期望只与两者之间的距离有关系, 即

$$f(r_{ij}) = r_{ij},$$

其中  $r_{ij}$  为会员  $i$  到任务  $j$  的距离. 那么再根据前述的吸引力均衡的定价原则, 可以通过求  $B$  的最小值来求解最优的定价方案, 即:

$$\hat{x} = \text{Argmin} \sum_{j=1}^m \omega_j \left\{ \sum_{i=1}^n \left( \frac{\frac{x_i}{r_{ij}}}{\sum_{k=1}^n \frac{x_k}{r_{kj}}} - \frac{1}{n} \right)^2 \right\}.$$

其中  $\hat{x} = (\hat{x}_1, \dots, \hat{x}_n)$  即为所求的任务  $1, \dots, n$  的最优定价,  $x = (x_1, \dots, x_n)$  是任务  $1, \dots, n$  的定价,  $\omega_j$  为会员的信誉权重.

### 4.3 基于遗传算法求解

通过观察发现, 附件一中的任务定价的最小为 65, 最大为 85. 而为了提高任务完成率, 先考虑适当提高价格的上限, 故在此考虑所有任务的价格都在  $[60, 120]$  的区间内. 若按照附件一中只考虑最小精度为 0.5, 则每一个任务的定价有 121 种可能, 但由于任务数量较大, 累计在一起的定价方案就有  $121^{835}$  种, 这显然是不可能利用遍历搜索直接进行计算. 所以很自然就能想到可以使用智能优化算法来求取优化模型的最优解. 遗传算法是目前较为流行的一种智能优化算法, 它通过模拟自然界的进化过程所发生的遗传和变异等过程来搜索最优解. 在这里, 考虑将任务的定价编码成染色体, 通过个体的适应性 (即 “方差”  $B$ ) 的大小来判断个体的优劣性, 通过模拟自然选择的过程将 “方差”  $B$  较大的个体淘汰, 让  $B$  较小个体之间两两交配, 得到子代. 子代再以一定的概率发生变异, 产生新的定价方案. 这样重复, 进化若干代之后则可以得到适应性最好的个体 (即最优的定价方案)[3]. 具体算法如下.

---

#### 算法 1: 遗传算法

---

输入:

进化代数  $T$ , 初始种群生成区间  $[a, b]$ , 初始种群数量  $M$ , 交配概率  $p_1$ , 变异的概率  $p_2$ , 迭代次数  $n$ .

---



---

---

过程:

**for**  $t_1 = 1 : n$

    随机生成初始种群;

    计算初始种群的适应度;

**for**  $t = 1 : T$

**if**  $\min(x) < 60$  或  $\max(x) > 90$

$B = Inf$ ;

**else**

            保留上一代  $B$  较小的个体, 淘汰掉  $B$  较大的个体;

            随机选择双亲, 随机生成下一代;

            下一代以概率  $p_2$  发生变异;

            计算下一代的适应度  $B$ ;

**end if**

**end for**

**end for**

---

输出:

最佳适应度  $B$ , 最优定价方案.

---

按照上述算法, 我们直接利用 MATLAB 中 Global Optimization Toolbox 的工具箱所提供的 `ga` 函数进行计算 (适应度函数参见附录), 并参考了文献 [3] 中给出的遗传接力进化方式. 为了尽可能保留种群中优良基因, 并且避免陷入局部最优, 以及能够搜索更多的范围, 我们设定种群初始大小  $M = 100$ , 进化代数  $T = 200$ , 交配概率为  $p_1 = 0.8$ , 变异概率为  $p_2 = 0.1$ . 我们选择 **Double vector** 的编码方式, 按照均匀分布随机生成初始种群, 按照 **Ranking** 的方式对个体适应度进行排序, 以及通过 **Stochastic uniform**(随机均匀分布) 的方式挑选双亲 [3, 4]. 再考虑到遗传算法具有一定的随机性, 于是对遗传算法整个过程进行了  $n = 20$  次的重复. 并选择了其中均衡性最好的一次作为最终的定价方案 (如图 9 所示), 其部分定价结果如表 6 所示 (完整定价方案结果见附录), 再对数据进行简单处理, 可以得到定价的区间分布情况如图 10 所示.

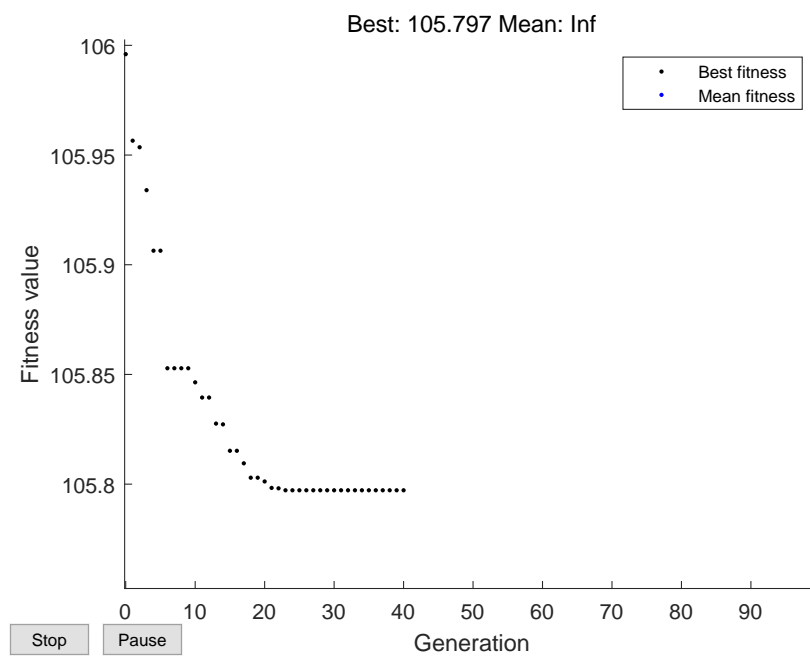


图 9 遗传算法收敛过程

表 6 部分定价结果

任务编号	定价	任务编号	定价	任务编号	定价
A0001	87.3	A0002	104.5	A0003	113.8
A0004	68.5	A0005	41.7	A0006	104.3
A0007	94.3	A0008	64.6	A0009	59.3
A0010	60.6	A0011	91.1	A0012	84.8

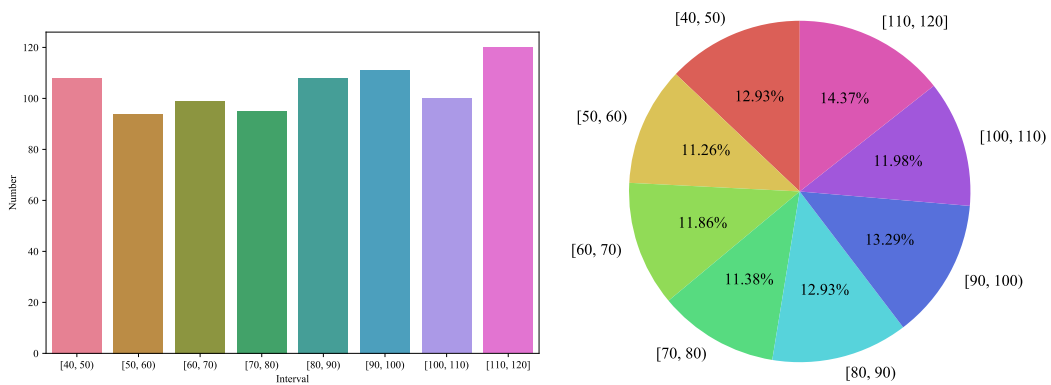


图 10 定价区间分布

## 4.4 问题三——新定价方案任务完成率的预测与评价

### 4.4.1 仿真模型的建立

问题三要求预测在新的定价方案下的任务完成率, 本文考虑通过建立随机仿真模型的方式模拟会员抢单去预测任务完成率.

影响任务是否完成包含两个方面的因素, 一是任务是否被成功分配, 二是被成功分配的任务能否被执行. 而任务的分配又包含两部分的内容, 一方面是平台端以何种方式派发任务, 另一方面是用户以何种方式来抢单. 只有当任务被平台公布, 同时有用户从 APP 中对该任务抢单, 才表明该任务被分配下去了. 任务成功分配给会员后能否被执行则受到店铺拒访、道路施工、天气不好、会员个人状态等随机因素的影响.

本文将随机仿真划分为会员抢单、任务执行两个过程来进行模拟.

#### 1. 会员抢单:

由于附件 2 中一共划分了 31 个预定开始的时间点, 故考虑分为 31 个时间段来考虑; 会员按照附件 2 中的预定任务开始时间上线, 所有上线会员同时抢单. 会员抢单规则如下:

**选单规则:** 如果未被接单的任务吸引力超过会员的吸引力阈值  $\text{Threshold}$ , 会员则参与该任务的抢单. 在抢单时要求会员抢单次数不得超过抢单限额  $M$ . 会员可完成任务的限额随着会员已抢单的任务数的增加而递减, 当会员抢单限额为 0 时, 会员不能继续抢单.

**派单规则:** 在每个预定时间内, 当一个任务只有一个会员预定时, 将该任务派发给该会员; 当多个会员同时对某个任务发出抢单请求时, 平台按照每个抢单会员的信誉值权重作为概率进行随机分配, 权重是模型 2 中的信誉值权重  $\omega_j$ , 具体来说, 假设有  $n$  个会员在同一时间抢单, 这些会员的权重分别为  $\omega_1, \omega_2, \dots, \omega_n$ , 那么第  $j$  个会员的被选中的概率为

$$p = \frac{\omega_j}{\sum_{j=1}^n \omega_j}, \quad p \in [0, 1].$$

#### 2. 任务执行:

会员在抢单成功后, 由于店铺拒访、道路施工、天气不好、会员个人状态等原因可能会无法完成任务, 因此我们假设所有这些随机因素对每个会员造成的影响服从正态分布, 并且与会员的信誉的权重相关. 具体来说, 设随机因素对会员  $j$  的影响大小为  $\mu_j$ , 设定临界值为  $\pm 1.96$ , 规定:

$$\begin{cases} \left| \frac{\mu_j}{\omega_j} \right| > 1.96, & \text{不执行任务;} \\ \left| \frac{\mu_j}{\omega_j} \right| \leq 1.96, & \text{执行任务.} \end{cases}$$

#### 4.4.2 仿真模型求解

##### 1. 吸引力阈值的设定:

本文利用附件 1 中原任务定价和完成率的原始数据来设定吸引力阈值  $Threshold$  的上限. 吸引力阈值  $Threshold$  的设定要尽可能使得在本文的仿真模型下模拟计算出的任务完成率与附件 1 中真实的任务完成率相吻合. 本文利用上述仿真模型对附件 1 中的原始数据进行了多次模拟, 通过遍历最终确定当吸引力阈值  $Threshold = 98$  时, 模拟出的原任务完成率为 62.6%, 较好地匹配了原任务完成率 62.5%, 使得本文建立的仿真规则与实际情况相符合.

##### 2. 算法

---

##### 算法 2: 仿真模拟算法

---

##### 输入:

吸引力矩阵  $f$  ( $f_{ij}$  表示任务  $i$  对会员  $j$  的吸引力), 会员信息矩阵  $x$  (包括抢单顺序编号, 编号, 信誉权重和任务限额), 任务矩阵  $y$  (包括任务定价, 任务编号), 任务数  $m$ , 会员数  $n$ , 吸引力阈值  $Threshold$ , 正态分布临界值  $k$ .

---

##### 过程:

计算每一时间段派发的任务数量矩阵  $a$ ;

##### 过程:

**for**  $i = 1 : 31$

$b$  为  $x$  中抢单顺序编号  $= i$  的行号;

    从  $f$  中取出子矩阵  $min\_matrix = f(:, b)$ ;

**for**  $j = 1 : b(i)$

        剔除  $b$  中限额为 0 的会员;

        找到  $min\_matrix$  中吸引力超过阈值的会员  $d$ ;

**if**  $len(d) == 0$

            continue;

**end if**

        生成一个 0 – 1 随机数  $rand\_1$ ;

---

---



---

```

sum_rand = 0;

for l = 1 : len(q)

    sum_rand += q(l) 代表的会员的信誉值归一化;

    if rand_1 > sum_rand

        记录任务分配矩阵  $g(j) = q(l)$  的会员编号;

        删除任务矩阵  $y$  中的这一行,  $n = n - 1$ ;

        该会员限额 -1;

        break;

    end if

end for

end for

生成一个正态分布随机数矩阵  $g_+$  对应于  $g$  中每一个元素;

 $g_+ = g_+ / x(3, :)$ ;

找出  $g_+$  中元素绝对值大于  $k$  的位置, 并将  $g$  中对应位置变为 0;

```

---

输出:

记录任务完成情况的矩阵  $g$  (大于 0 的元素表示完成该任务的会员, 0 表示任务未派发或未完成), 以及任务完成率  $P$ .

---

### 3. 模拟结果

按照上述算法, 本文利用 MATLAB 进行仿真计算. 经过 100 次模拟后得到新定价方案的平均完成率为 70.78%. 于是得到新的定价方案与附件一中数据的对比.

## 附录 A 模型一

```

%本程序主要计算问题一中各影响因子
clear all;
clc;
load matlab.mat

```

```

task_vip_distance = getAllDist(vip_pos, test_data); %计算任务点与会员点之间的距离
density = getDensity(task_vip_distance, 0.05); %计算每个任务点处的会员密度
task_credit = task_credit(task_vip_distance, vip_credit, 0.05);
    %计算每个任务点所在位置的相对信誉值
[cluster_dis, minDis, test_idx, center] = vip_cluster(vip_pos, 4, test_data);
    %对会员点分布聚类
plotpri_den(isdone, density, task_pri) %绘制全部任务的会员密度-定价曲线
plotpri_dis(isdone, minDis, task_pri) %绘制任务与聚类中心最短距离-任务定价曲线


%获得任务点和会员两两之间的距离矩阵
function distance = getAllDist(vip_pos, task_pos)
vip_num = size(vip_pos,1);
task_num = size(task_pos,1);
distance = zeros(task_num, vip_num);
for i = 1:task_num
    for j = 1:vip_num
        distance(i, j) = ((task_pos(i,1) - vip_pos(j,1))^2 + (task_pos(i,2) -
            vip_pos(j,2))^2)^(1/2);
    end
end
end

%求指定任务点的会员密度
function gd = getDensity(distance, disThresh)
tmp = distance;
task_num = size(distance,1);
vip_num = size(distance,2);
gd = zeros(task_num, 1);
for i = 1:task_num
    count = 0;
    for j = 1:vip_num
        if tmp(i, j) <= disThresh %如果距离小于阈值, 纳入计数范围
            count = count + 1;
        end
    end
    if count == 0
        gd(i,1) = 0;
    else
        gd(i,1) = log(count); %取对数处理
    end
end
end

%根据会员分布得到指定地点的相对信誉值
function tc = task_credit(distance, vip_credit, disThresh)
tmp = distance;

```

```

task_num = size(distance,1);
vip_num = size(distance,2);
tc = zeros(task_num, 1);
for i = 1:task_num
    count = 0;
    sum = 0;
    for j = 1:vip_num
        if tmp(i, j) <= disThresh %任务与会员距离在阈值内，则加入计数
            count = count + 1;
            sum = sum + log(vip_credit(j));
        end
    end
    if count > 0
        tc(i) = sum/count;
    else
        tc(i) = 0;
    end
end
tc_min = min(tc);
for i = 1:task_num
    if tc(i) ~= 0
        tc(i) = tc(i) - tc_min;
    end
end
end

%会员聚类,并计算每个任务点到各个聚类中心的距离和最短距离
function [distance, minDis, idx, C] = vip_cluster(vip_pos, k, task_pos)
[idx,C,~,~] = kmeans(vip_pos,k);
task_num = size(task_pos, 1);
vip_num = size(vip_pos, 1);
figure
for ii = 1:vip_num
    scatter(vip_pos(ii,2), vip_pos(ii,1), 5, idx(ii)*3);
    hold on
end

distance = zeros(task_num, k);
for i = 1:task_num
    for j = 1:k
        distance(i, j) = ((task_pos(i,1) - C(j,1))^2 + (task_pos(i,2) - C(j,2))^2)^(1/2);
    end
end
end

%绘制全部任务的会员密度-定价曲线
function plotpri_den(isdone, density, task_pri)

```

```

figure
task_num = size(isdone, 1);
for i = 1:task_num
    scatter(density(i), task_pri(i), 5, 3)
    hold on
end
b = robustfit(density, task_pri)
x = linspace(min(density), max(density), 1000);
y = b(2)*x + b(1);
plot(x, y);
title('全部任务的相对会员密度-定价曲线');
xlabel('相对会员密度');
ylabel('任务价格');
end

%绘制任务与聚类中心最短距离-任务定价曲线
function plotpri_dis(isdone, minDis, task_pri)
figure
task_num = size(isdone, 1);
valid_num = sum(isdone,1);
new_minDis = zeros(valid_num,1);
new_taskpri = zeros(valid_num,1);
count = 1;
for i = 1:task_num
    if isdone(i) == 1
        scatter(minDis(i), task_pri(i), 5, 3)
        hold on
        new_minDis(count) = minDis(i);
        new_taskpri(count) = task_pri(i);
        count = count + 1;
    end
end
b = robustfit(new_minDis, new_taskpri)
x = linspace(min(new_minDis), max(new_minDis), 1000);
y = b(2)*x + b(1);
plot(x, y);
title('任务与聚类中心距离-任务定价曲线');
xlabel('任务与聚类中心距离');
ylabel('任务价格');
end

```

## 附录 B 模型二中的代码

### 2.1 主程序



%主程序：本程序采用遗传算法接力进化，  
%将上次进化结束后得到的最终种群作为下次输入的初始种群

```
clear all;
n = 1; % 迭代次数
Bestx = zeros(n, 835);
BestFval = zeros(n, 1);

%% 进行 n 次重复
for t = 1: n
    clc;
    close all;
    clear finnal_pop fval reason x;

    %% 进化
    T = 200;%进化的代数
    optionsOrigin = gaoptimset('PopInitRange', [60; 120], 'Generations',T/2,
        'MigrationFraction', 0.1, 'PopulationSize', 100);
    [x, fval, reason, output, finnal_pop] = ga(@sum_111, 835, optionsOrigin);

    %% 进行第二次接力进化
    options1 = gaoptimset('PopInitRange', [60; 120], 'Generations', T/2,
        'InitialPopulation', finnal_pop, 'StallGenLimit', 50, 'StallTimeLimit', 20,
        'MigrationFraction', 0.1, 'PopulationSize', 100, 'PlotFcns', @gaplotbestf);
    [x, fval, reason, output, finnal_pop] = ga(@sum_111, 835, options1);
    Bestx(t,:) = x;
    BestFval(t,1) = fval;
end
```

## 2.2 适应度函数

% 子函数：适应度函数，同时也是所求的目标函数

```
function f = fitnessfun(x)
%% 判断 x 是否在区间 [0, 150] 内
q_0 = min(x);
q_1 = max(x);
if(q_0 < 60 | q_1 > 120)
    f = Inf;
else
    %% 预处理数据
    n = 1877;
    m = 835;
    l = load('summer_data\test.csv'); %导入会员位置信息
    y = load('summer_data\test1.csv'); %导入任务位置信息
```

```

omega = load('summer_data\test2.csv'); %导入会员信誉信息
l_1 = repmat(l(:, 1), 1, n);
l_2 = repmat(l(:, 2), 1, n);
y_1 = repmat(y(:, 1)', m, 1);
y_2 = repmat(y(:, 2)', m, 1);

%% 生成距离矩阵
%r(i, j) 表示第 i 个会员到第 j 个任务的距离
r = zeros(m, n);
r = sqrt((l_1 - y_1) .^ 2 * 111.2 ^ 2 + (l_2 - y_2) .^ 2 * 111.3 ^ 2);

%% 计算适应度函数
a = zeros(n, 1);
b = zeros(n, 1);
for j = 1: n
    a(j) = sum(x(:) ./ r(:, j));
    b(j) = sum(((x(:) ./ r(:, j)) ./ a(j) - 1 / n) .^ 2);
end

c = sum(omega(:) .* b(:));
f = c;
end

```

## 2.3 计算吸引力矩阵

```

%% 预处理数据
n = 1877;
m = 835;
l = load('summer_data\test.csv'); %导入会员位置信息
y = load('summer_data\test1.csv'); %导入任务位置信息
omega = load('summer_data\test2.csv'); %导入会员信誉信息
l_1 = repmat(l(:, 1), 1, n);
l_2 = repmat(l(:, 2), 1, n);
y_1 = repmat(y(:, 1)', m, 1);
y_2 = repmat(y(:, 2)', m, 1);

%% 生成距离矩阵
%r(i, j) 表示第 i 个会员到第 j 个任务的距离
r = zeros(m, n);
r = sqrt((l_1 - y_1) .^ 2 * 111.2 ^ 2 + (l_2 - y_2) .^ 2 * 111.3 ^ 2);

%% 生成定价矩阵
x = load('summer_data\price_matrix_5.csv');
x = repmat(x(:, 1), 1, n);

```

```
%% 计算吸引力矩阵  
f = x ./ r;
```

## 附录 C 模型三中的代码

```
% 本程序是主要模拟派发订单的过程  
clear all;  
clc;  
  
Times = 100; %重复次数  
Prob = zeros(Times, 1);  
for cycle_1 = 1: Times  
    %% 导入数据  
    f = load('summer_data\xiyinli_matrix_1.csv'); %吸引力矩阵  
    x = load('summer_data\VIP.csv'); %会员信息矩阵 包含：第几轮开始可以抢单，  
        %会员编号，信誉权重，以及限额  
    y = load('summer_data\price_matrix_1.csv'); %任务矩阵 包含任务定价，任务编号  
  
    %% 初始化数据  
    m = 1877; %会员数  
    n = 835; %任务数  
    g = zeros(n, 1); %记录每个任务接单情况矩阵，为0表示没人接单  
    Threshold = 100;  
  
    % 计算每一次派单时可以抢单的人数  
    a = zeros(31, 1);  
    for i = 1: 31  
        a(i) = sum(x(:, 1) <= i);  
    end  
  
    %% 模拟派单过程  
    for i = 1: 31  
  
        c_2 = zeros(n, 1); %得到第i次派单任务在任务矩阵中的行数  
  
        for j = 1: n  
            c_2(j) = j;  
        end  
  
        c_3 = y(c_2, 2); %得到了第i次派单的任务编号  
        k = find(x(:, 1) == i); %得到第i次可以抢单的会员的编号  
  
        %% 剔除掉k中已经达到限额的会员  
        for j = 1: length(k)  
            if x(k(j), 4) == 0
```

```

        k(j) = 0;
    end
end
k(find(k == 0), :) = [];

min_matrix = f(:, k(:));

%% 派单 遍历每一个任务
for j = 1: n
    q = find(min_matrix(j, :) >= Threshold); %找出超过吸引力阈值的会员

    % 剔除掉没人要的任务
    if length(q) == 0
        continue;
    else

        %% 将会员信誉权重归一化
        sum_1 = 0;
        for l = 1: length(q)
            sum_1 = sum_1 + x(k(q(l)), 3);
        end

        rand_1 = rand();
        sum_rand = 0;

        %% 利用随机数判断将任务派发给谁
        for l = 1: length(q)
            sum_rand = sum_rand + x(k(q(l)), 3) / sum_1;
            if sum_rand > rand_1
                g(c_3(j)) = k(q(l)); %在任务派发矩阵中记录
                y([c_2(j)], 2) = 0; %在任务矩阵中标记已经派发的任务
                x(k(q(l)), 4) = x(k(q(l)), 4) - 1; %接单的会员限额减去1
                n = n - 1;
                break;
            end
        end
    end
end
end

y(find(y(:, 2) == 0), :) = []; %删除掉任务矩阵中被标记的任务所在的行
end

%% 判断任务是否完成
normal_rand = normrnd(0, 1, 835, 1);
normal_rand_omega = normal_rand .* (1 ./ x(3, :));
p = find(abs(normal_rand_omega) > 1.96);
g(p) = zeros(length(p), 1);

```

```
%% 输出
Prob(cycle_1) = length(find(g > 0)) / length(g);
end

Probability = mean(Prob)
```