

# “拍照赚钱”任务定价的研究分析

## 摘要

本文为了分析和优化拍照任务的定价策略，将任务的地理位置、集中程度，会员的分布状况、信誉水平等作为定价的影响因素。首先对现有的拍照任务定价进行分析，在可视化分析和量化统计的支持下，分析出现有的定价策略的规律。之后，通过筛选会员信息，提出更合理的定价方案，并通过了评价函数的验证。本文运用了包括回归分析理论、多目标规划、最短路径原理等多种理论，建立并优化数学模型，使任务定价有了更合理、更有效的依据。

针对问题（1），首先绘制出了任务价格等高图，结合会员的地理位置分布，定性分析出任务定价 $p$ 和会员分布密度 $N_a$ 的关系。然后统计任务周围 2.5 千米内的会员数量，绘制价格和平均会员密度离散统计图，拟合出函数 $N_a = \frac{16.5}{p - 63.84}$ ，并对此模型进行分析，得出任务未完成的原因主要是任务的价格较低和任务周围缺少活跃会员。

针对问题（2），在问题（1）的基础上，优化了会员成分，提高了会员信息的有效性，建立了新的定价模型。为了量化定价函数的优劣，引入任务评分函数 $F = k \times (x_0 + x) \left[ 1 - \frac{F_0}{M} \right] - \lambda \times U_n$ 。经过验证，证实新的定价策优于原始定价方案。

针对问题（3），为了得到对哪些任务进行打包的一般规律，运用Matlab计算出每个任务与最近任务经纬度差的平方和，统计出现集中任务的所有任务位置和各位置对应的任务数量，根据统计结果确定打包的范围。任务包的定价应避免简单的 1+1=2 的定价方案，故对任务包建立系数激励模型 $P = k_1 k_2 \sum_{i=1}^{n_0} p_i$ 。

针对问题（4），先根据问题（2）的模型，计算出任务独立发布的价格。再根据新任务的地理分布，对集中的任务进行打包。然后采用问题（3）中的对任务包的定价模型，计算每一个任务包的价格。最后采用问题（2）中的评分模型进评价，验证了定价方案具有更高的合理性。

关键词： 最短路径 定价评分函数 多目标规划 众包

## 一、问题重述

### 1.1 问题背景

随着网络的高速发展，一种新的工作模式渐渐的发展起来，公司利用互联网发布工作内容，用户将劳动成果通过互联网发送给公司以获取报酬。本题中的“拍照赚钱”模式亦是如此，用户下载 APP 客户端，并注册成为会员，然后再领取拍照任务。此类基于移动互联网的自助式劳务众包平台，可以向企业提供其所需要的多种商业调查数据，相对于传统的市场调查所花费的代价更小，并且能保证了数据来源的真实合理性，完成调查所花费的时间也大大缩短。

### 1.2 相关信息

APP 中的任务的定价直接影响到会员对任务的选择，因为如果定价不合理（比如价格偏低），就会存在一些任务没有用户去完成，从而导致总任务完成率较低。从附件中提供的数据来看，任务的定价可能与任务点的经纬度、以及会员经纬度、会员数目、会员质量有关。

### 1.3 需要解决的问题

- 1) 根据对题目所给数据的分析，建立起任务价格和会员信息之间的函数表达式。分析部分任务未完成的原因。
- 2) 根据对任务数据和用户数据，建立新的定价模型，并和原定价模型进行比较。
- 3) 对任务点建立打包策略并且制定打包后的定价模型，并分析新定价模型的优劣。
- 4) 对新项目给出合适的定价方案，并分析所给方案的效果。

## 二、问题分析

### 2.1 问题一的分析

问题一要求探究附件一中的任务定价规律，即要从任务地理位置及会员分布入手。此外，会员的信誉值和预订任务限额隐含了该用户完成任务的能力，对于能力较强的用户在考虑时给予较高的权重。从价格方面对未完成的任务进行分析，未完成任务的定价普遍低于已完成任务的定价。此外，任务周围的活跃用户数量也与定价存在关联。

### 2.2 问题二的分析

问题二要求设计新的定价方案并与原方案进行比较。在问题一的基础上，对定价方案进行改进。通过筛选出高信誉会员作为新的参考因素来对模型进行优化。本文建立了任务评分模型，旨在量化分析定价策略的优劣。该模型的基本假设如下：任务评分越高，任务越优质，更容易得到会员的青睐。

### 2.3 问题三的分析

问题三要求对集中的任务进行打包处理。这些任务由于距离成本较低，更符合会员的投机心理。从任务发布者的角度出发，将任务打包发布能避免投机现象，提高任务的完成率。对于打包的定价策略，拟采用激励机制，激励用户完成整组任务，完成的任务越多，平均单个任务的回报越高。

### 2.4 问题四的分析

问题四要求对附件三中的新项目给出定价方案，并评价方案的实施效果。在问题二和问题三的基础上，先对项目进行打包处理，将打包后的数据分为四类，按照问题三中提出的打包价格模型进行计算。按照问题二中提出的任务评分模型进行评估，如果得分更高则说明当年定价策略较合适。

### 三、模型假设

1. 假设会员是理性人，考虑对任务的选择时总是倾向于自身利益的最大化；
2. 假设每项任务的难度均相同，不影响定价；
3. 考虑到用户的时间和精力，假设每个用户每天最多能完成 5 个任务。

### 四、符号说明

$U_{xi}$  : 第  $i$  号会员位置的经度；  
 $U_{yi}$  : 第  $i$  号会员位置的纬度；  
 $H_i$  : 第  $i$  号筛选会员的信誉值；  
 $C_i$  : 第  $i$  号的任务分配额；  
 $B$  : 计算点所在的纬度；  
 $R$  : 赤道半径；  
 $r$  : 极轴半径；  
 $S$  : 弧长；  
 $N$  : 周围用户数量；  
 $N_a$  : 单个任务点周围的平均会员数目；  
 $U_n$  : 任务点附近会员密度；  
 $p$  : 任务价格；  
 $p_i$  : 第  $i$  号任务的价格；  
 $P$  : 任务包价格；  
 $L_c$  : 当前用户的任务分配限额；  
 $L_a$  : 所有用户任务分配限额的总和；  
 $L_f$  : 过滤后的任务分配限额；  
 $T_a$  : 总的任务个数；  
 $N_f$  : 过滤后任务点周围用户平均数量；  
 $N_p$  : 相同价格的任务点总数；  
 $N_{[i]}$  : 每一个任务点周围的会员数；  
 $F$  : 问题二中的任务评分标准；  
 $M$  : 饱和评分。

## 五、模型的建立及求解

### 5.1 问题一

#### 5.1.1 定价策略的分析与求解

针对任务数据，首先提取出任务数据的 GPS 坐标，以散点图的形式呈现于地图上，粗略的得到目标任务的分布情况。对于附件二中的用户数据，先对其进行预处理。剔除具有下列特征的用户：

- a. 坐标严重脱离任务区域的；
- b. 信用度极高，但能接任务数很少的；
- c. 能接任务数很高，但是信用度极低的；
- d. 能接任务数极低同时信用度极低的；

对数据进行分析得到量化的筛选标准：

- 1.  $112^{\circ} \leq U_{x_i} \leq 115^{\circ}, i = 1, 2, 3, \dots$
- 2.  $22^{\circ} \leq U_{y_i} \leq 24^{\circ}, i = 1, 2, 3, \dots$
- 3.  $1 \leq H_i, i = 1, 2, 3, \dots$
- 4.  $10 < C_i, i = 1, 2, 3, \dots$

将处理后的用户和坐标分别呈现在两坐标系中，如图 1 所示：

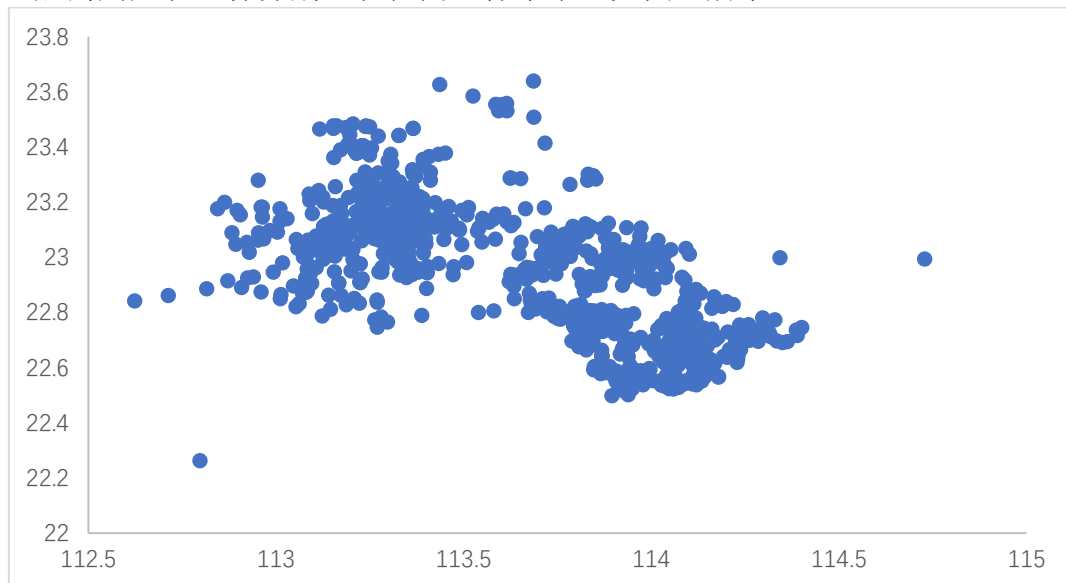


图 1 用户分布图

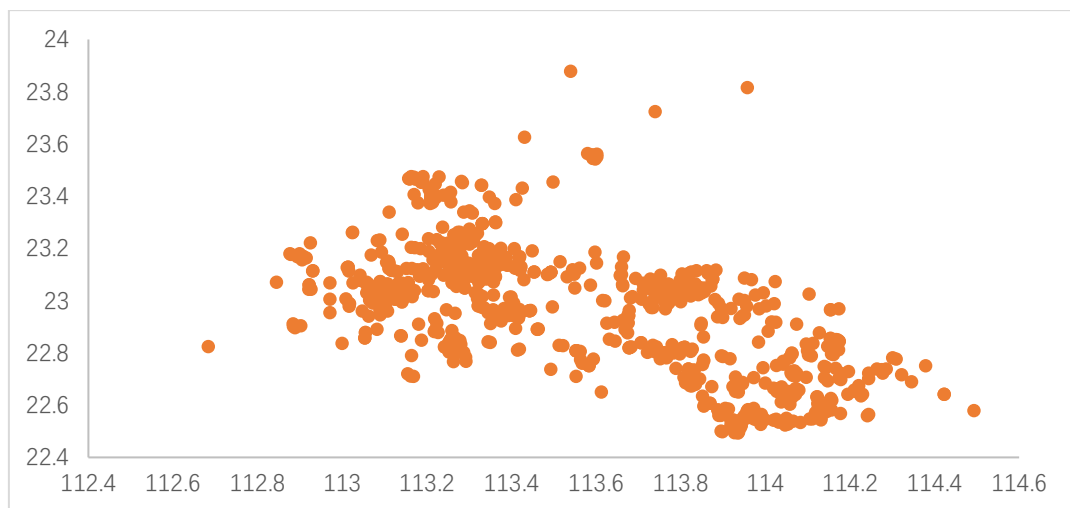


图 2 任务分布图

从图中可以看出用户的分布和任务分布都存在聚集现象，由此猜想任务的价格会和用户的分布相关。为了进一步理清价格分布规律，本文绘制了价格等高图（图 3）。在价格等高图中，同一颜色的等高线有着相同的任务价格，并且越接近深蓝色，任务的价格越低，越接近橙色，任务的价格越高。发现价格较低的地区用户也比较集中，价格较高的任务周边会员数量较少，这进一步验证了之前的假设，为了明确两者之间的函数关系，本文绘制了用户分布和价格等高图（图 4）。

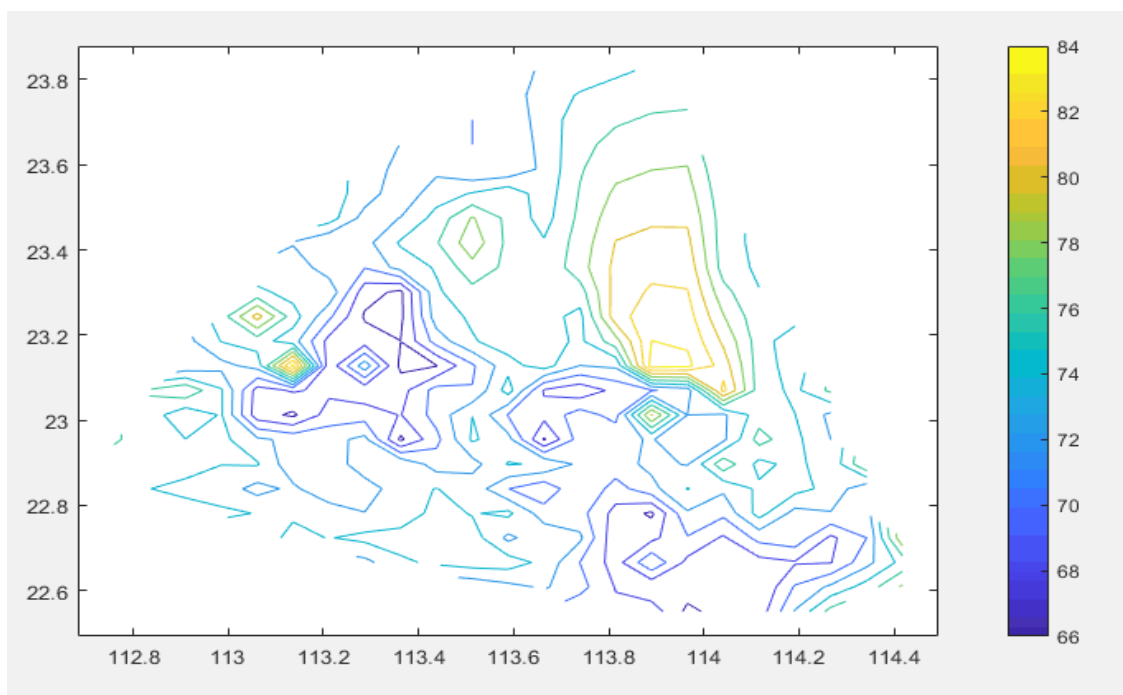


图 3 价格分布等高图

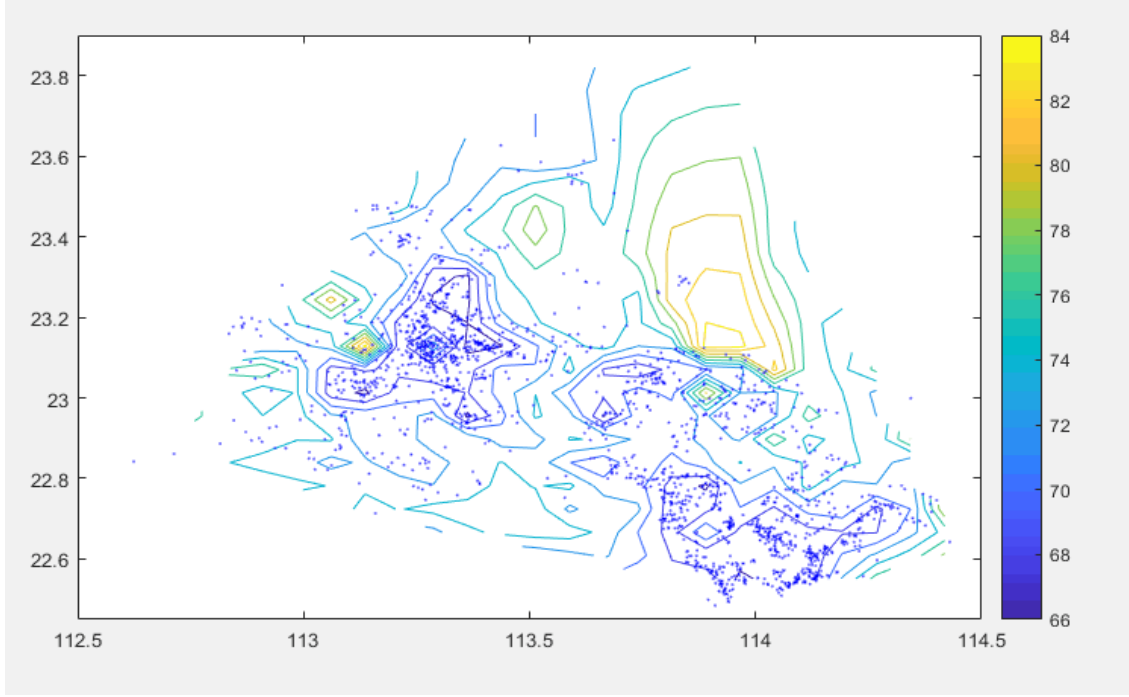


图 4 用户分布和价格等高图

下面通过回归分析法对两者之间的关系进行量化。

**Step 1:**

针对题中给出的 835 组任务，以任务为中心，对该点周围 2.5km 内的有效会员数量进行统计。-

**1) 地球面距离计算**

题中给出的是经纬度，由于地球是一个球面，首先要两组经纬度转换成球面弧长。考虑采用如下算法<sup>[1]</sup>：

纬度每相差一度的距离基本是固定的，大约为 111km。经度之间的弧长差和两点所在的纬度相关，而平均每一经度的弧长随着纬度的增高而降低。采用公式 (1) 进行计算：

$$S = \frac{\cos B \times (R - B \times (R - r) / 90) \times 2\pi}{360} \quad (1)$$

其中， $B$  为计算点所在的纬度， $R$  为赤道半径， $r$  为极轴半径，取  $R = 6378.137km$ ， $r = 6356.752km$ 。若两计算点不在同一纬度，可取这两点所在的纬度的每一度的经向平均弧长的均值作为计算弧长的参数，如下：

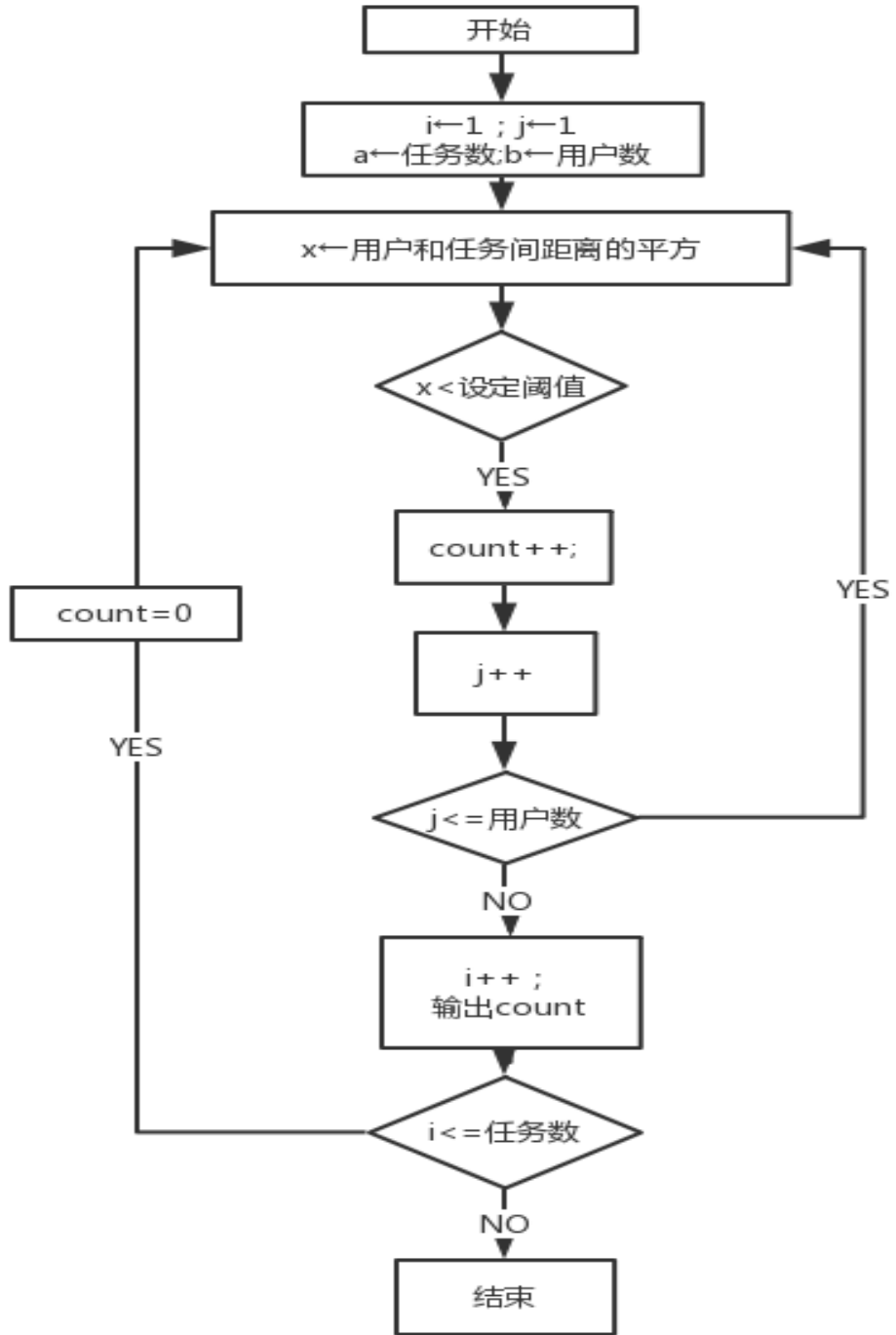
$$S = [\cos B_1 \times (R - B_1 \times (R - r) / 90 \times 2\pi / 360 + \cos B_2 \times (R - B_2 \times (R - r) / 90 \times 2\pi / 360)] / 2$$

由上式可得每度经向的平均弧长均值  $S_1$  为：

$$S_1 = [\cos B_1 \times (6378.137 - 237.611 \times B_2) + \cos B_2 \times (6378.137 - 237.611 \times B_2)] \times \pi / 360$$

**2) 符合条件的用户统计**

通过遍历的方式对任务周围符合要求的用户进行统计，算法流程图如下：



### Step 2

为了降低单个任务点周围用户数量分布异常对结果产生较大的影响，将单个价格对应的周围会员数定义为：

$$N = \frac{\sum_i^{P_n} c[i]}{P_n} \quad (2)$$

### Step 3 定价规律函数拟合

通过上述分析，得到了任务周边会员数量和价格的两组数据，对其进行回归分

析，希望建立起两者之间的函数关系。为了使函数更好的反映价格规律，采用最小二乘法进行拟合，函数图像如图 5 所示，横坐标代表任务价格，纵坐标代表周围平均会员数量。

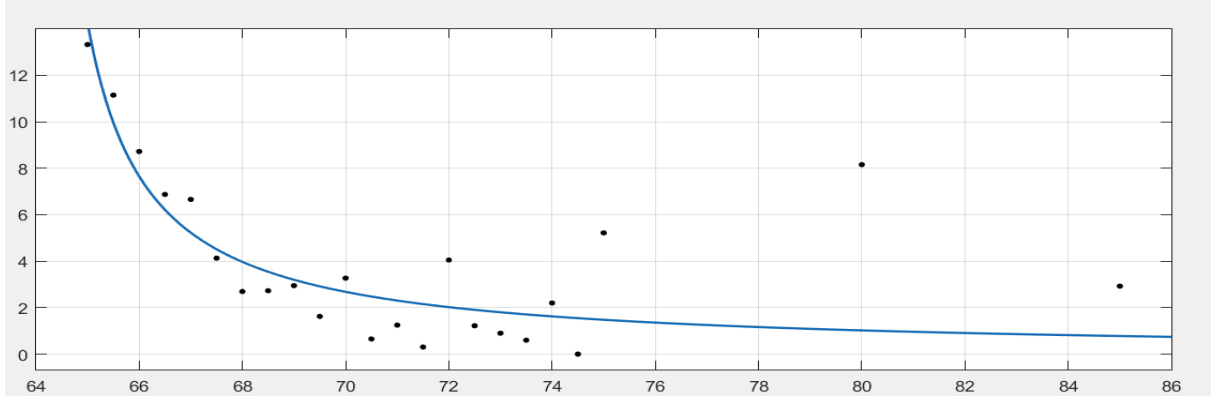


图 5 函数拟合

由上述拟合得到如下表达式：

$$N_a = \frac{\alpha_1}{p + \beta_1} \quad (3)$$

题中设定的任务对周围会员的感知范围为 2.5 千米时，得到的  $\alpha_1=16.5$ ， $\beta_1$ -63.84。

### 5.1.2 任务失败的原因分析

通过对数据的整理，以及对所做出图象的分析，本文归纳出了以下几条任务未完成的原因：

- 1) 对于一些价格很高的任务点，虽然价格很高，但是其周围的会员数量很少甚至没有，用户们去那里做任务最后的效益没有就近选择的，因此导致该任务点无人选取
- 2) 对于那些周围虽然存在许多会员数量的任务点（1）由于任务价格订的太低缺乏激励性，导致并无会员去选择执行这个任务（2）当地的一些现实问题，比如可能是台风等天气的影响或者其他因素。

## 5.2 问题二

### 5.2.1 新定价方案的制定与效果分析

#### 1) 有效用户的筛选

在问题一的评价模型中，仅只是对数据有严重错误的用户进行剔除，这必然是不符合实际的。题中给出信息，任务分配时实际上是根据预定限额所占比例进行配发，也就是说，对于可接任务限额不满足一定需求的用户实际上是无法分配到任务的，对这类用户进行剔除能更加有效的反映任务能否被完成的状况。

剔除函数：

$$\frac{T_a}{L_a} \times L_c \geq 1 \quad (4)$$



根据上述剔除函数，粗略的得到当前用户可接任务限额  $L_c \geq 15.3$  时才能得到 1 个任务。但是用户集中存在一些限额大于 75 的超级用户，也就是按照限额每天能分到至少 5 个任务。按照对问题的假设用户每天最多能完成 5 个任务，所以，本文将限额大于 75 的用户均看作 75 进行计算。对于可接任务额限小于 8 的用户，也进行剔除。对剩余用户的总额限进行统计，得到处理过的总额限  $L_f$  为 7276，新剔除函数如下：

$$\frac{T_a}{L_f} \times L_c \geq 1 \quad (5)$$

经计算，得到  $L_c \geq 8.6$  即可接任务额限在 9 及以上时才能接到任务，这部分用户即为所要筛选的有效用户，数量为 351 个。

## 2) 价格函数的构建

在第一问价格函数的基础上，将对有效用户的更换成上述的 351 个，按照问题一中提供的算法进行计算。为了处理部分任务点周围可能存在有效会员数为 0 的情况，不妨将定价函数的最小值设置为 1，即当  $F_n$  为 0 的时候将  $F_n$  置为 1。以此为限制条件更新上述的价格函数，此时系数  $\alpha_2=8.98$ ， $\beta_2=-62$ 。

$$P = \frac{\alpha_2}{F_n} - \beta_2 \quad (6)$$

## 3) 任务评分模型分析与假设

1. 任务评分越高，任务越优质，用户更加倾向于完成评分高的任务。
2. 任务评分会因价格的提高而增加，但评分是有限度的，当价格足够高的时候，评分将趋近于固定值  $M$ ，这时无论怎么提高价格，对任务评分都不会由实质性的提升。
3. 任务评分会因平均价格的提高而降低。
4. 设  $F(x)$  为当提升的价格为  $t$  时的任务评分， $M$  为饱和评分，代表着实际问题中任务所能评得的最高分。
5.  $U_n$  为任务周围的有效用户数量，有效用户数量越低，该任务更可能被抢到。
6.  $\lambda$  为衰减因子，任务评分会随着任务距离最近的用户集中心越来越远而降低。

## 4) 评分模型的建立

$$F = k \times (x_0 + x) \left[ 1 - \frac{F_0}{M} \right] - \lambda \times U_n \quad (7)$$

$k$  和  $\lambda$  均为正系数， $F_0$  代表着初始评分， $x_0$  代表着原定价策略中每个任务的初始价格，当  $x=0$  时， $F'=F_0$  即为原定价的任务评分，当改变定价策略后，若有着更高的平均得分即可认为是用户总体而言完成了更多的任务。

## 5) 方案比较

对于附件一中给出的 835 份任务，计算出平均价格为 69.11 元，当采取新的价格函数后，得到新的平均价格为 75.08 元，较原来有显著提升。借助问题二中所建立的任务评分模型对方案进行检测。由于  $k$  和  $\lambda$  均大于零，对于单个任务。其  $F_0$  由原始定价决定即为固定值，所以可以确定当总体价格提升， $k \times (x_0 + x) \left[ 1 - \frac{F_0}{M} \right]$  会增加，又由

于用户集是筛选过的，导致单个任务临近范围内的用户数量减少即 $\lambda \times U_n$ 降低，所以，总体上评价函数（7）的值是增加的，即新的价格函数能更好的完成任务。

### 5.3 问题三

#### 5.3.1 打包模型的建立

出于更加高效地发布、分配任务的考虑，将位置比较集中的任务一起打包发布。对于此改动出现的原因与可能造成的影响进行如下思考：

1. 在某个位置分布着一定的会员，同时也分布了位置较为集中的多个任务。任务单独发布时，存在该地区会员优先争抢具有价格优势的的任务的现象，该地区的其余会员只能放弃或只考虑部分任务。最终在任务进行阶段，同一地区用户拥挤，同时较远地区的任务完成度低。

2. 打包处理的原则与基本原则基本一致，为了更合理地完成任任务，避免会员的拥堵和浪费，提高任务完成度。

3. 任务包可以根据被打包的任务成员的地理位置，划分为同一处的任务进行打包和位置集中度高的任务进行打包。

4. 当任务被打包后，因为任务成员具有地理位置的一致性，所以外部对任务成员位置的认知也具有统一性，但对于任务包内部任务成员的位置差异绝对存在，任务完成顺序具有一定程度的重要性。

5. 出于③和④的考虑，任务包的报酬规律应避免简单的  $1+1=2$  的定价模型，通过任务特征，得到任务系数，通过微调系数调整任务报酬规律。

#### 5.3.2 模型的分析、建立与求解

##### 1) 数据预处理

在某些位置同时存在多个任务且任务的价格彼此之间不同。因此统计任务列表中的地理位置相同的任务数量，得到以下表格。

编号	纬度	经度	数量	编号	纬度	经度	数量
1	23.1087118	113.4928846	4	17	22.9413279	113.6767951	2
2	23.2945141	113.3313063	4	18	22.9609933	113.4427752	2
3	22.8206529	113.6793236	3	19	22.9654734	114.1542792	2
4	22.9345908	113.4167995	3	20	22.9879601	113.0170859	2
5	23.0386685	113.2034172	3	21	23.0479764	113.2892933	2
6	22.5239495	113.9434416	2	22	23.0723988	113.0896832	2
7	22.6414407	114.4227421	2	23	23.0824818	113.8756202	2
8	22.6800886	113.8347861	2	24	23.1152677	113.1169399	2
9	22.7333562	113.8312919	2	25	23.1282410	113.0139313	2
10	22.7385904	113.8184493	2	26	23.1381659	113.3728361	2
11	22.7810825	114.0574014	2	27	23.1441612	113.2937609	2
12	22.8210453	113.8090357	2	28	23.2128091	113.2563331	2
13	22.8225055	113.2418331	2	29	23.2205046	113.2826379	2
14	22.8387028	113.7066879	2	30	23.2315862	113.2643764	2
15	22.8785231	113.0553623	2	31	23.3948403	113.2214471	2

表 5-1：任务位置与对应的任务数量

通过分类汇总，对同一位置不同任务编号的任务进行研究，并选取以下具有特征任

务，分析得出以下结论：

1. 同一位置的多个任务虽然定价相同，但是会员因为随机选择任务，部分任务被闲置。
2. 对于同一位置的不同定价的任务，反映出来的一般规律，表现为任务标价高的任务被完成且任务标价低的任务只是部分完成。
3. 结合问题一，不难得出，对于同一位置的任务都不完成的现象，可以归因于错误的定价方案，采取问题二的优化方案能提高任务完成率。

任务编号	任务 gps 纬度	任务 gps 经度	任务标价	任务执行情况
A0032	22. 5239495	113. 9434416	65. 5	0
A0462	22. 5239495	113. 9434416	65. 5	1
A0070	22. 7385903	113. 8184493	66. 5	0
A0078	22. 7385903	113. 8184493	66. 5	1
A0160	22. 9345908	113. 4167995	70	0
A0268	22. 9345908	113. 4167995	70	1
A0278	22. 9345908	113. 4167995	70	1
A0164	22. 9609932	113. 4427752	66. 5	0
A0415	22. 9609932	113. 4427752	72	1
A0141	23. 1087118	113. 4928846	66. 5	0
A0151	23. 1087118	113. 4928846	66. 5	1
A0155	23. 1087118	113. 4928846	66. 5	1
A0210	23. 1087118	113. 4928846	67	1
A0166	23. 2945140	113. 3313063	69	0
A0321	23. 2945140	113. 3313063	69	1
A0335	23. 2945140	113. 3313063	69	1
A0341	23. 2945140	113. 3313063	69	1
A0038	22. 6414407	114. 4227421	75	0
A0063	22. 6414407	114. 4227421	75	0
A0087	22. 7333561	113. 8312919	66	0
A0099	22. 7333561	113. 8312919	66	0

表 5-2:同一位置任务的标价与执行情况

## 2) 打包任务数量的分析与思考

除了地理位置完全相同的任务，地理位置相近的任务也在考虑打包的范围内。毫无疑问，与任务成员分散的任务包相比，位置相同的任务组成的任务包更具有吸引力，所以，打包策略将任务分为零散任务和任务包，同时任务包又分为将同一位置的任务进行打包的任务包和将不同位置的任务进行打包的任务包。

打包策略能强制性要求会员承担完成任务包内其他任务的责任。根据目前的任务分布，初步将打包后的每一任务包内包含的任务数量设置在 2-4 个范围内，若同一位置任务数量超过 5 个，则受每日最大任务完成量的影响，将任务包最大容量设置为 5。因为任务包内的任务成员位置特征高度统一，所以会员选择从哪个任务开始完成不影响任务包的完成。

下面按照任务包的规模从最短路径方面进行分析。

### 1. 任务包中任务数量为 2。

当任务数量为 2 时，完成任务仅需要从任务 A 前往任务 B。如图 5-3(a) 所示。

2. 任务包中任务数量为 3。

完成任务时通常选择 3 条道路中距离最短的两条路。如图 5-3(b) 所示。

3. 任务包中任务数量 $\geq 4$ 。

运用 Dijkstra 算法, 把图中顶点集合  $V$  分成两组, 第一组为已求出最短路径的顶点集合 (用  $S$  表示, 初始时  $S$  中只有一个源点, 以后每求得一条最短路径, 就将加入到集合  $S$  中, 直到全部顶点都加入到  $S$  中, 算法就结束了), 第二组为其余未确定最短路径的顶点集合 (用  $U$  表示), 按最短路径长度的递增次序依次把第二组的顶点加入  $S$  中。在加入的过程中, 总保持从源点  $v$  到  $S$  中各顶点的最短路径长度不大于从源点  $v$  到  $U$  中任何顶点的最短路径长度。此外, 每个顶点对应一个距离,  $S$  中的顶点的距离就是从  $v$  到此顶点的最短路径长度,  $U$  中的顶点的距离, 是从  $v$  到此顶点只包括  $S$  中的顶点为中间顶点的当前最短路径长度<sup>[2]</sup>。如图 5-3(c) 所示。

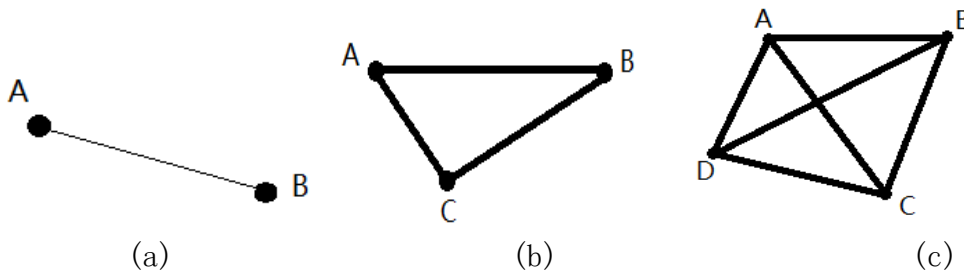


图 5-3 不同规模任务包的路线示意图

显而易见, 当任务数量超过 3 个时, 任务内部的完成顺序很大程度上影响了任务完成的速度, 影响了会员完成任务的积极性, 加大了任务包的完成难度。而只包含两个任务的包与包含三个任务的包相比, 任务难度几乎相同, 却严重浪费任务包策略的优势, 综上所述, 对于任务包的成员数量在 3 个及以上时, 采取经济鼓励制度。

### 3) 建立价格模型

通过上述分析得如下表达式:

$$P = k_1 k_2 \sum_{i=1}^{n_0} p_i \quad (8)$$

其中,  $P$  为任务包的标价,  $k_1$  为任务完成数量鼓励系数,  $k_2$  为基于任务包成员位置的分布衰减系数,  $n_0$  为任务包内的任务数量,  $p_i$  为独立发布时任务包内各个任务的价格。系数数值如下:

$$k_1 = \begin{cases} 1.0, & n_0 = 2 \\ 1.1, & n_0 \geq 3 \end{cases}, \quad k_2 = \begin{cases} 0.95, & s_0 = 0 \\ 1.0, & s_0 \rightarrow 0 \end{cases}$$

其中,  $n_0$  为任务包内任务数量,  $s_0$  为任务包内任务成员离散程度。

### 5.3.3 模型的分析与检验

统计不同条件下的任务的报酬情况:

任务成员数量 $n_0$	分布离散程度 $s_0$	报酬系数 $k_1 k_2$	任务包定价分析与评价
$n_0 = 2$	$s_0 = 0$	0.95	任务成员数量少, 任务分布在同一位置, 整体收益衰减。
$n_0 = 2$	$s_0 \rightarrow 0$	1.0	任务成员数量少, 任务分布在临近位置, 整体收益不变。
$n_0 \geq 3$	$s_0 = 0$	1.045	任务成员数量多, 任务分布在同一位置, 整体收益略微增加。

$n_0 \geq 3$	$s_0 \rightarrow 0$	1.1	任务成员数量多，任务分布在临近位置，整体收益较大幅度增加。
--------------	---------------------	-----	-------------------------------

由上表可知，打包收益系数调整方法，鼓励会员完成更多数量任务，活跃于更多地区，提高了任务完成率，模型基本合理。

## 5.4 问题四

### 5.4.1 新任务定价方案的确立

通过对任务坐标的分析，我们发现新任务的聚集现象较为严重，故采用问题三中提出的打包方案进行处理。由于数据较多无法在论文中全部体现，所以分别选择了具有不同特征的任务包，分为四类进行展示：

#### 1) 任务成员数较多，且位置相同

该位置发布了 43 条任务，由于每日完成任务数量的限制，每 5 条进行打包，将其中一个任务包的任务信息整理如下：

任务编号	纬度	经度	初始价格
C0696	23.29451408	113.3313063	62.21
C0698	23.29451408	113.3313063	62.21
C0703	23.29451408	113.3313063	62.21
C0708	23.29451408	113.3313063	62.21
C0740	23.29451408	113.3313063	62.21

实例 1 任务数量 5 离散程度 0

运用问题二中的价格函数计算初始价格，通过公式（9）计算出整个任务包价格为：

$$P = k_1 k_2 \sum_{i=1}^{n_0} p_i = 325, \text{ 平均价格为 } 65。$$

#### 2) 任务成员数较少，且位置相同

任务编号	纬度	经度	初始价格
C0211	22.53470014	113.934485	66.49
C1332	22.53470014	113.934485	66.49

实例 2 任务数量 2 离散程度 0

$$\text{任务包价格为: } P = k_1 k_2 \sum_{i=1}^{n_0} p_i = 136, \text{ 平均价格为 } 68。$$

#### 3) 任务成员数较多，且位置集中

任务编号	纬度	经度	初始价格
C0742	23.22596441	113.272686	62.08
C1821	23.2258322	113.2730093	62.08
C2000	23.22589883	113.2729862	62.08
C1826	23.22628842	113.2726572	62.08

实例 3 任务数量 4 离散程度较小

$$\text{任务包价格为: } P = k_1 k_2 \sum_{i=1}^{n_0} p_i = 273, \text{ 平均价格为 } 68。$$

#### 4) 任务成员数较少，且位置集中

任务编号	纬度	经度	初始价格
C0037	22.732721	114.261045	62.09
C0038	22.73287891	114.2611436	62.09

实例 3 任务数量 2 离散程度较小

$$\text{任务包价格为: } P = k_1 k_2 \sum_{i=1}^{n_0} p_i = 124, \text{ 平均价格为 } 62。$$

### 5.4.2 新项目任务定价模型的分配评价

由于新项目中仅仅给定了任务的坐标，本文拟采用问题二中提出的任务评价函数公式（7）进行评价。首先将 $x_0$ 设置为按照公式（6）计算出的价格。并将价格增量 $x$ 取为0，解出每个任务的初始评分 $F_0$ 。按照公式（8）计算出的价格并代入，得到的任务评分越高，模型效果良好。

## 六、模型的评价

### 6.1 模型的优点

本文的模型充分考虑到了用户和公司的双方利益。从用户的角度，本文建立的模型对于用户而言获得相同金额的汇报花费更少的时间和精力，从公司的角度来说，价格以较小的提升就能有效的提高任务完成率。

### 6.2 模型的缺点

在模型假设中我们认定每个任务难度是相同的，但是现实生活中，每个任务的难度是存在着差异。

## 七、参考文献

- [1] 邓光发. 计算球面距离的简便公式[J]. 中学数学, 1999, (01):48-49.
- [2] 华山大师兄. 最短路径—Dijkstra 算法和 Floyd 算法.  
<http://www.cnblogs.com/biyeymyhjob/archive/2012/07/31/2615833.html>

## 八、附录

本文使用 Matlab 以及 Excel 进行计算，程序见支撑材料。

程序一：nihe.m

```
clear;
target = 'target.xlsx';
t = xlsread(target);
t1 = t(:,1);
t2 = t(:,2);
t3 = t(:,3);

user = 'user.xlsx';
u = xlsread(user);
u1 = u(:,1);
u2 = u(:,2);
u3 = u(:,3);

count = zeros(835,1);

for i=1:835 %任务
    for j=1:351 %用户
        dis = (t1(i)-u1(j)).^2 *111*111*0.847329 + (t2(i)-u2(j)).^2*111*111;
        if dis<6.25
            count(i)= count(i)+1;
        end
    end
end

ave = zeros(23,1);
temp = 0;
for i=(temp+1):temp+65
    ave(1) = ave(1) + count(i);
end
ave(1) = ave(1)/65;
temp = temp+65;

for i=(temp+1):temp+150
    ave(2) = ave(2) + count(i);
end
ave(2) = ave(2)/150;
temp = temp+150;

for i=(temp+1):temp+103
    ave(3) = ave(3) + count(i);
end
ave(3) = ave(3)/103;
temp = temp+103;

for i=(temp+1):temp+63
    ave(4) = ave(4) + count(i);
```

```

end
ave(4) = ave(4)/63;
temp = temp+63;

for i=(temp+1):temp+38
    ave(5) = ave(5) + count(i);
end
ave(5) = ave(5)/38;
temp = temp+38;

for i=(temp+1):temp+23
    ave(6) = ave(6) + count(i);
end
ave(6) = ave(6)/23;
temp = temp+23;

for i=(temp+1):temp+30
    ave(7) = ave(7) + count(i);
end
ave(7) = ave(7)/23;
temp = temp+30;

for i=(temp+1):temp+11
    ave(8) = ave(8) + count(i);
end
ave(8) = ave(8)/11;
temp = temp+11;

for i=(temp+1):temp+19
    ave(9) = ave(9) + count(i);
end
ave(9) = ave(9)/19;
temp = temp+19;

for i=(temp+1):temp+8
    ave(10) = ave(10) + count(i);
end
ave(10) = ave(10)/8;
temp = temp+8;

for i=(temp+1):temp+96
    ave(11) = ave(11) + count(i);
end
ave(11) = ave(11)/96;
temp = temp+96;

for i=(temp+1):temp+11
    ave(12) = ave(12) + count(i);
end
ave(12) = ave(12)/23;

```



```

temp = temp+11;

for i=(temp+1):temp+4
    ave(13) = ave(13) + count(i);
end
ave(13) = ave(13)/4;
temp = temp+4;

for i=(temp+1):temp+5
    ave(14) = ave(14) + count(i);
end
ave(14) = ave(14)/23;
temp = temp+5;

for i=(temp+1):temp+60
    ave(15) = ave(15) + count(i);
end
ave(15) = ave(15)/60;
temp = temp+60;

for i=(temp+1):temp+9
    ave(16) = ave(16) + count(i);
end
ave(16) = ave(16)/9;
temp = temp+9;

for i=(temp+1):temp+10
    ave(17) = ave(17) + count(i);
end
ave(17) = ave(17)/10;
temp = temp+10;

for i=(temp+1):temp+5
    ave(18) = ave(18) + count(i);
end
ave(18) = ave(18)/5;
temp = temp+5;

for i=(temp+1):temp+5
    ave(19) = ave(19) + count(i);
end
ave(19) = ave(19)/5;
temp = temp+5;

for i=(temp+1):temp+2
    ave(20) = ave(20) + count(i);
end
ave(20) = ave(20)/2;
temp = temp+2;

```

```

for i=(temp+1):temp+78
    ave(21) = ave(21) + count(i);
end
ave(21) = ave(21)/23;
temp = temp+78;

for i=(temp+1):temp+13
    ave(22) = ave(22) + count(i);
end
ave(22) = ave(22)/13;
temp = temp+13;

for i=(temp+1):temp+27
    ave(23) = ave(23) + count(i);
end
ave(23) = ave(23)/27;
temp = temp+27;

ls = [65
65.5
66
66.5
67
67.5
68
68.5
69
69.5
70
70.5
71
71.5
72
72.5
73
73.5
74
74.5
75
80
85];

price = zeros(835,1);
ave_price = 0;
for i=1:835
    price(i)=8.98/count(i) +62;
    ave_price = ave_price + price(i);
end
ave_price = ave_price/835

```

程序二：MinRoute.m

```
clear;clc;
A=load('所有任务地理位置.txt');
num = size(A); % 加载数据;
within = 0.00001;

for i = 1:num(1)
    A(i,3) = 999999999;
end

for i = 1:num(1)
    for j = 1:num(1)
        if(j == i)
            continue;
        else
            temp = (A(i,1)-A(j,1))^2 + (A(i,2)-A(j,2))^2;
            if(temp < A(i,3))
                A(i,3) = temp;
            end
        end
    end
end
end
```

程序三：Code.m

```
clear;clc;
A=load('form.txt');
num = size(A); % МгдиЪ§Он;
within = 0.00000015;

for i = 1:num(1)
    A(i,3) = 999999999;
end

for i = 1:num(1)
    for j = 1:num(1)
        if(j == i)
            continue;
        else
            temp = (A(i,1) - A(j,1))^2 + (A(i,2) - A(j,2))^2;
            if(temp < A(i,3))
                A(i,3) = temp;
            end
        end
    end
end
end
```

程序四：problem4.m

```
clear;
target = 'target.xlsx';
t = xlsread(target);
t1 = t(:,1);
t2 = t(:,2);
```

```

user = 'user.xlsx';
u = xlsread(user);
u1 = u(:,1);
u2 = u(:,2);
u3 = u(:,3);

count = zeros(2066,1);

for i=1:2066 %任务
    for j=1:351 %用户
        dis = (t1(i)-u1(j)).^2 *111*111*0.847329 + (t2(i)-u2(j)).^2*111*111;
        if dis<6.25
            count(i)= count(i)+1;
        end
    end
end
end

```