

“拍照赚钱”任务定价策略的研究

摘要

随着互联网普及率的提高和网络虚拟产品交易的兴起,基于移动网络的自助式劳务众包平台逐渐成为企业商业检查和信息搜集的新方式,但不合理的任务定价会增加任务发布者的成本或降低用户参与度。本文通过对“拍照赚钱”这一众包任务定价规律的深入研究,给出较为合理的定价方案。

针对问题一,应用 Excel 的三维地图功能将会员和任务的位置信息定位在地图上,发现会员位置集中分布在三个区域,因此在 SPSS 中对会员位置进行聚类分析。求出任务位置与聚类中心距离的调和平均值,将任务定价与调和平均距离作回归分析,发现二者呈正相关的线性函数关系,即项目的任务定价规律为:任务定价随调和平均距离增大而提高。然后计算出任务点 10 公里范围内的单位距离定价、会员的平均任务限额、平均信誉值和任务数量,从任务个体和区域两个角度对未完成任务和已完成任务的以上指标进行差异性分析,从而得出导致任务未完成的四个方面的原因。

针对问题二,以任务完成情况为被解释变量,任务定价、平均距离、平均任务限额、任务密度为解释变量,构建 Logistic 回归模型,得出任务定价与其他变量及任务完成概率的函数关系。从任务的完成率和定价两个方面对原定价方案进行优化,建立以任务完成的概率为自变量,任务定价和完成率为目标变量的规划模型,得出新的任务定价,使在两个方面同时达到理想值。与原定价方案进行比较,发现新的定价方案既能降低任务发布者的成本,又能提高任务的完成率。

针对问题三,首先采用 DBSCAN 算法对位置比较集中的任务进行联合打包,得到 438 包任务。然后在问题二模型的基础上,将打包的任务作为一个新任务发布,结合打包的特点对原指标的计算方式进行修改,由此得到打包后新的定价方案。为了验证修改后定价模型对最终任务完成情况的影响,建立会员任务选择博弈模型,先从预订时间和信誉值两方面对会员进行排序,然后会员依次遵循四个原则对任务包进行选择,在所有的会员选择完后,观察此时的任务完成情况,发现任务的完成率有显著提高。

针对问题四,结合前三问已有的模型和结论,先计算新项目中任务的相应指标数据,然后用 DBSCAN 算法对位置较集中的任务进行打包,将其代入修改后的定价模型得到任务的定价方案,应用会员任务选择博弈模型,分别得到新的定价方案和问题一中原有定价方案下的任务完成情况。对比两组的价格和任务完成情况,发现新的定价方案既能降低任务发布者成本,又能提高任务完成率,实施效果较好。

最后,对本文的模型进行评价,指出其优缺点并提出了改进方案。

关键字: 聚类 SPSS Logistic 回归模型 任务完成率 DBSCAN 算法 任务选择博弈模型

一、 问题重述

1.1 背景资料与条件

随着互联网新兴社会媒体的发展和智能化自助式服务的兴起，众包已经发展成为一种新型的电子商务模式并在市场中占据越来越重要的地位。众包是一种开发式创新，其完成与否与任务发布者的任务定价密切相关。“拍照赚钱”就是一种基于移动互联网的自助式劳务众包平台，用户在该平台下载 APP 并注册成为会员，然后领取需要拍照的任务，赚取 APP 对任务所标定的酬金。这不但能为企业提供各种商业检查和信息搜集，相比传统的市场调查方式大大节省了调查成本，而且有效地保证了调查数据真实性，缩短了调查的周期。对任务的定价既要尽可能减少任务发布者的成本，又要保证任务有较高的完成率。因此，研究任务的定价方案具有重要意义。

1.2 需要解决的问题

1. 研究附件一中项目的任务定价规律，分析任务未完成的原因。
2. 为附件一中的项目设计新的任务定价方案，并和原方案进行比较。
3. 实际情况下，多个任务可能因为位置比较集中，导致用户会争相选择，一种考虑是将这些任务联合在一起打包发布。在这种考虑下，如何修改前面的定价模型，对最终的任务完成情况又有什么影响？
4. 对附件三中的新项目给出你的任务定价方案，并评价该方案的实施效果。

二、 模型假设

- 1、假设任务实际完成情况能够直接反映价格制定的合理程度。
- 2、假设在问题三会员的选择策略中，只要任务数不超过限额，会员就能完成此任务。
- 3、假设提供的会员信息及任务信息具有时效性和准确性，能够代表实际情况。
- 4、假设不考虑交通情况、区域经济水平等对任务定价的影响。

三、符号说明

| | |
|----------|-------------------------|
| d_i | 任务点到观测点的距离 |
| S_j | 第 j 个任务位置到聚类中心的调和平均距离 |
| S_i | 任务位置到第 i 个会员中心点的距离 |
| p_j | 第 j 个任务的定价 |
| m_i | 第 i 个会员的预订任务限额 |
| C_i | 第 i 个会员的信誉值 |
| γ | 任务完成的概率函数 |
| l | 会员平均任务限额 |
| t | 任务密度 |
| S | 任务位置到会员位置的平均距离 |
| X | 任务定价 |
| M | 概率期望值 |

四、问题分析

4.1 问题一的分析

问题一要求对已结束的项目任务数据进行分析，研究其定价规律并分析其中未完成任务的原因。

我们首先在 Excel 表格中将附件二给出的会员位置信息定位在地图上，发现会员位置点大致在三处集中分布，进一步运用 SPSS 对会员位置聚类分析，得到三类会员位置的中心点。计算每个任务位置分别到三处会员位置中心点的距离，求其调和平均数，并将此值作为衡量距离的综合指标。最后应用 MATLAB 对任务定价和调和平均距离进行线性回归分析，发现两变量呈正相关，即附件一中项

目的任务定价规律为：定价随任务位置与会员位置间的调和平均距离增大而提高。

在分析未完成任务的原因中，我们从任务个体和区域两个角度进行分析。我们以每个任务位置为圆心，10 公里为半径作任务圆，位于此范围外的会员到任务点的距离过远，因此认为只有在任务圆内的会员才有选择该任务的可能。对于每个任务个体，我们从任务定价、任务圆中平均任务限额、会员平均信誉值和任务密度四个方面构建指标进行分析。分别对完成的和未完成的任务所对应的四个指标数据进行差异性分析，从单个任务的角度分析任务未完成的原因。对于不同聚类区域，分别对三个聚类中心作半径为 30 公里的区域圆，计算范围内任务的完成率、会员平均任务限额、平均信誉值和任务密度，将三个区域的指标数据进行对比，从区域影响的角度分析任务未完成的原因。

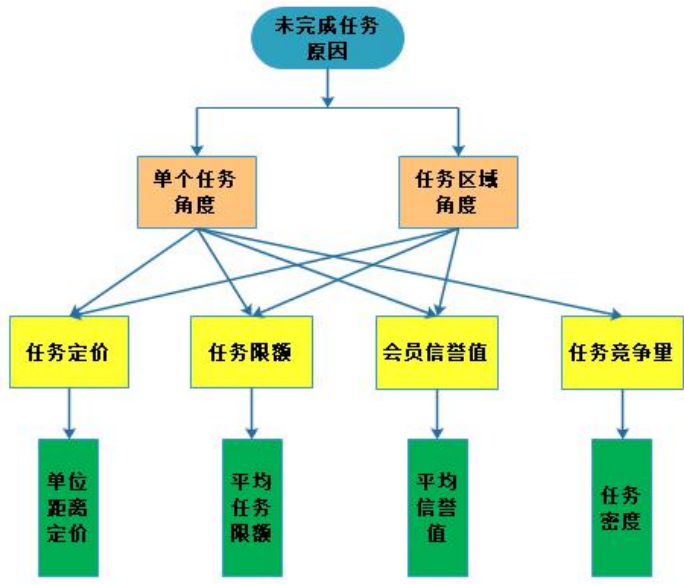


图 1 未完成任务原因的分析

4.2 问题二的分析

问题二要求为附件一中的项目设计新的任务定价方案，并和原方案进行比较。

我们首先构建 Logistic 回归模型，以任务定价、平均距离、平均任务限额、任务密度为解释变量，任务完成情况为被解释变量，计算出任务的定价与其他变量及任务完成概率函数中的参数值，从而得到其函数关系。然后，我们从任务的完成率和定价两个方面对原定价方案进行优化，首先令所有任务概率相等，概率的值作为任务完成率，建立以概率为自变量，任务定价和完成率为目标变量的规划模型，由此得出能使在任务在定价和完成率两个方面同时达到理想值的新的任

务定价方案。将此方案与原定价方案进行比较，发现新的定价方案既能降低任务定价，减少任务发布者的成本；又能提高任务的完成率，使会员的收益增加。

4.3 问题三的分析

问题三要求考虑将位置比较集中的任务联合在一起打包发布，在此情况下，对定价模型进行修改，并分析对最终的任务完成情况的影响。

从任务位置来看，由于任务分布的不均匀性且存在较为孤立的任务点，所以我们采用 DBSCAN 算法对位置比较集中的任务进行打包，将其定义为一个新的任务发布，最终得到 438 包任务。在问题二 Logistic 模型的基础上，对原有的指标进行修改，其中平均任务限额和任务密度由任务包内所有任务的求平均得到，而任务位置到会员位置的平均距离用回路附加的方法求得，从而得到在打包的形势下的新定价方案。

在此定价方案下，我们先对会员的能力属性进行排序，优先考虑会员的预订任务时间，在预订任务时间相同的情况下，信誉值较高的会员优先选择任务完成。同时会员在选择任务时满足四个原则：①第 i 名会员选择的任务没有被前 $i-1$ 名会员选中过；②定义会员的最低接受任务定价，当定价低于其最低接受时不再选择任务；③不能选择任务数大于其预订任务限额的任务包；④总是选择单位距离定价最高的任务包。在遍历所有的会员后，观察此时的任务完成情况，从而得出新的定价模型对任务完成情况的影响。

4.4 问题四的分析

问题四要求对附件三中的新项目给出任务定价方案，并评价实施效果。

结合前三问已有的模型和结论，先应用问题一中的模型计算任务定价、平均距离、平均任务限额和任务密度四个指标数据，然后用 DBSCAN 算法对位置较集中的任务进行打包，将其代入问题三修改后的定价模型得到任务的定价方案。将此时的定价与通过原有规律定出的价格分别应用会员任务选择博弈模型，得到两种定价策略下任务的完成情况。对比两组的价格和任务完成情况，发现新的定价方案实施效果较优。

五、模型建立与求解

5.1 问题一的建模与求解

在问题一中，我们先用 Excel 的三维地图功能将会员和任务的位置信息定位在地图上，发现会员位置集中分布在三个区域，因此在 SPSS 中对会员位置进行

聚类分析。然后求出每个任务位置与聚类中心距离的调和平均值，将任务定价与调和平均距离作回归分析，以此来观察任务的定价规律。

在任务未完成的原因分析中，我们计算任务点 10 公里范围内的单位距离定价、会员的平均任务限额、平均信誉值和任务数量，从任务个体和区域两个角度对未完成任务和已完成任务的以上指标进行差异性分析。

5.1.1 会员位置聚类分析

聚类分析是根据个体的特征，对研究对象按照一定的标准进行分类，分析中通常用距离来表示个体的差异程度。为了得到附件一中项目的任务定价规律，我们考虑任务位置和会员位置的关系对其的影响。

首先用快速聚类法对会员的位置信息进行聚类，因为快速聚类法要求分类的组别已知，它以牺牲多个解为代价来获得较高的执行率。所以我们根据附件二给出的会员信息数据，应用 Excel 的三维地图功能将每个会员的位置经纬度定位到平面地图上，如图 2 所示：



图 2 平面地图上会员的位置

从图中可以看出会员的位置点大致分布在三处，进而应用 SPSS 软件，随机选出三个观测量作为初始聚类中心点，用公式（1）计算同类中一点距观测点的欧式距离：

$$d_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} \quad (1)$$

其中， (x_0, y_0) 表示观测点的坐标， d_i 表示任务点到观测点的欧式距离。

根据距离最小的原则将每个观测量分配到这三类中，然后将每一类中的观测量计算变量均值，这三个均值又形成三个新的聚类中心点，不断进行迭代后得到

三个会员位置的聚类中心点经纬度分别为 22.959° N、113.814° E，22.654° N、114.050° E，23.123° N、113.272° E。

5.1.2 定价规律的给出

在 5.1.1 中我们将会员位置聚类分析得到会员位置的三大聚集点及它们的中心，因为当任务位置距任意一个会员聚类中心较近时就可以认为该位置与会员的距离较近，所以我们提出了调和平均距离这一指标。

分别计算每个任务位置到三处会员聚类中心点的距离，记为 $s_i (i=1,2,3)$ 。应用公式（2）求三段距离的调和平均数 s ，即

$$\frac{1}{s} = \frac{1}{s_1} + \frac{1}{s_2} + \frac{1}{s_3}, \quad (2)$$

将此值作为衡量距离的综合指标。

最后应用 MATLAB 对附件一中任务定价和调和平均距离进行线性回归分析，得到两变量关系函数为

$$y = 35.8816x + 65.1883, \quad (3)$$

其中， y 表示任务定价， x 表示调和平均距离。

综上可得，附件一中项目的任务定价规律为：任务位置与聚类中心间的调和平均距离越大，任务定价越高。

5.1.3 分析体系的构建

在对项目的任务定价规律的研究中，我们得出任务位置与聚类中心间的调和平均距离越大，任务定价越高的结论。由此可知，当聚类中心距任务点过远时，对该会员的任务定价过高，任务发布者的成本太大。因此，我们以每个任务点为圆心，10 公里为半径作任务圆（如图 3 所示），规定只有此范围内的会员可以接相应的任务。从任务定价、会员限额、会员信誉值、周围任务密度四个方面构建指标对任务的完成率进行分析。

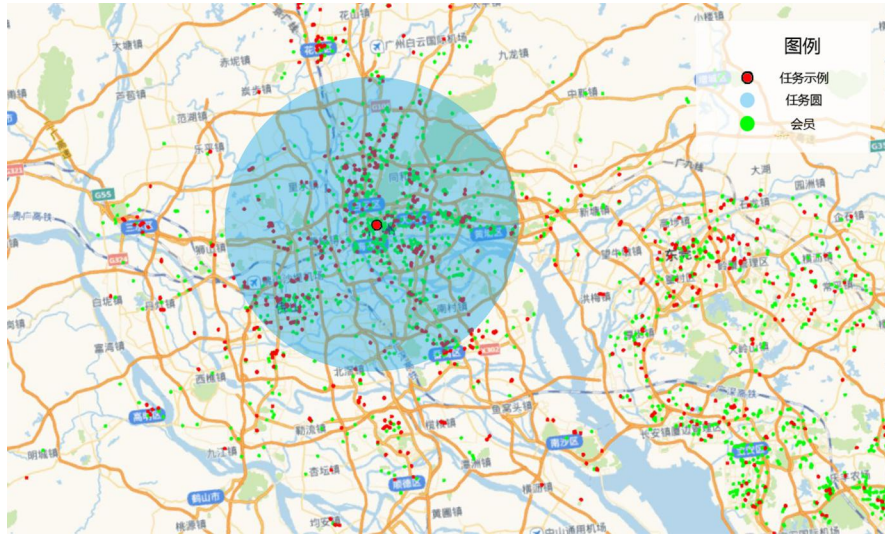


图 3 任务圆示例

1、单位距离定价

已知任务的定价与任务和会员间的调和平均距离成正比，会员希望得到较高的任务报酬，而任务发布者想尽量降低成本。因此在定价方面不合理可能导致任务不能完成。我们构建单位距离定价这一指标，定量的描述任务定价与调和平均距离的关系，对比分析已完成任务和未完成任务在该指标上的差异。单位距离定价计算公式为

$$\overline{p_j} = \frac{p_j}{S_j}, \quad (4)$$

其中， p_j 表示第 j 个任务的标价， S_j 表示第 j 个任务位置到聚类中心的调和平均距离。

2、平均任务限额

每个会员都有自己的预订任务限额，当任务量超过会员的任务限额时会造成因人手不足而导致部分任务无法完成的问题。因此，我们从任务限额方面考虑，构建平均任务限额这一指标，由任务圆中所有会员的任务限额除以任务圆中的所有会员数，计算公式如下：

$$\overline{m_i} = \frac{\sum_i m_i}{n} \quad (5)$$

其中， m_i 表示第 i 个会员的预订任务限额， n 表示在任务圆中的会员数。

3、平均信誉值

信誉是指讲信用的好声誉，会员的信誉值越高，代表他在接到任务的完成率越高，在其他条件相同或相近的情况下，任务发布者更希望由信誉值高的会员来完成任务。当一个任务周边的会员普遍信誉值较低时，该任务就有较大的可能性不会被完成。因此我们从会员信誉值方面考虑，构建平均信誉值这一指标，在任务圆内，由所有会员的信誉值除以所有的会员数，计算公式如下：

$$\bar{c_i} = \frac{\sum_i c_i}{n} \quad (6)$$

其中， c_i 表示第 i 个会员的信誉值， n 表示在任务圆中的会员数。

4、任务密度

任务密度是描述对于某个任务的任务圆范围内的任务数量，由范围内的任务数量与任务圆面积的比值得到。任务密度的增加会导致任务之间存在竞争，会员在选择时，在竞争中相对处于劣势的任务因不会被选择导致不能完成。

5.1.4 任务未完成的原因分析

为了更直观的观察已完成任务和未完成任务与聚类中心的关系，我们在地图上分别用不同的颜色做出了标注，其中绿色表示完成的任务，红色表示未完成的任务，紫色表示会员的位置，如图 4 所示：

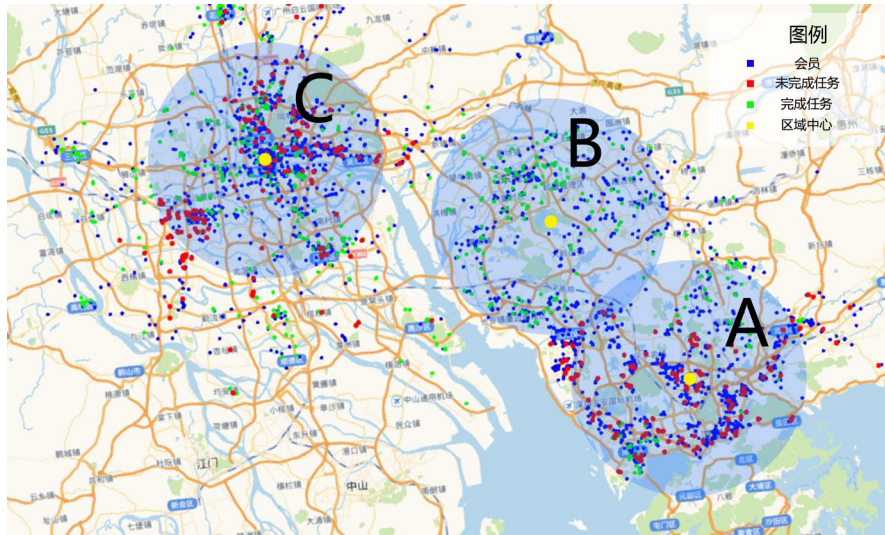


图 4 任务和会员的位置

在任务圆中，我们运用 MATLAB 分别对已完成的任务和未完成的任务上述四个指标值进行计算，得到结果如表 1 所示（在已完成和未完成的任务中各取 10 组）：

表 1 (a) 已完成任务的分析指标

| 单位距离 定价 | 平均任务 限额 | 平均信誉值 | 任务密度 |
|---------------|------------|--------|------|
| 183.02 | 8.39 | 377.65 | 181 |
| 197.74 | 7.47 | 342.28 | 165 |
| 178.45 | 8.51 | 390.64 | 160 |
| 181.43 | 8.92 | 456.59 | 164 |
| 191.80 | 8.25 | 406.06 | 142 |
| 188.00 | 8.49 | 412.61 | 142 |
| 190.54 | 6.54 | 319.75 | 157 |
| 197.52 | 6.93 | 295.16 | 146 |
| 180.01 | 7.72 | 386.97 | 142 |
| 211.38 | 6.45 | 256.75 | 140 |

表 1 (b) 未完成任务的分析指标

| 单位距离 定价 | 平均任务 限额 | 平均信誉值 | 任务密度 |
|------------|------------|--------|------|
| 117.24 | 5.23 | 189.59 | 221 |
| 117.16 | 5.24 | 190.40 | 171 |
| 126.45 | 4.57 | 188.35 | 177 |
| 127.51 | 4.49 | 178.14 | 166 |
| 121.81 | 5.23 | 178.48 | 217 |
| 118.38 | 5.25 | 182.51 | 167 |
| 100.88 | 6.64 | 127.60 | 158 |
| 97.70 | 8.13 | 65.43 | 176 |
| 137.49 | 4.57 | 165.18 | 144 |
| 161.94 | 4.64 | 160.52 | 121 |

由表 1 中指标数据可以看出, 已完成的任务在单位距离定价、平均任务限额和平均信誉值三个指标上都普遍高于未完成任务, 而在任务密度上, 已完成任务明显低于未完成任务。

然后, 分别以 5.1.1 得到的三个聚类中心为圆心, 30 公里为半径作任务圆, 计算任务圆内的任务完成率、所有会员的平均任务限额、会员的平均信誉值和任务密度, 由此观察是否任务所在的区域位置对任务的完成率造成了影响。

由此对任务未完成的原因作出以下四点分析:

1、任务的定价较低, 导致会员接受任务的意愿较小。在发布任务时, 应考虑会员离任务位置的调和平均距离, 只从会员的角度来说, 在一定距离的情况下, 任务的定价越高, 会员越愿意接受并完成任务得到相应的酬金。表中数据反映了已完成任务的单位调和平均距离的定价都在 180 以上, 而未完成任务的定价大多在 130 以下, 因此部分任务距离远而定价低直接导致了任务的未完成。

2、会员能接受的任务额度较小, 出现因人手不够而有任务无人接的局面。由于会员本身的能力和素质的影响, 当某区域中任务量超过会员的预订任务限额时, 多出来的任务便不会被完成, 从会员的平均任务额度可以看出, 已完成任务大于未完成任务。

3、会员的信誉值较低, 出现会员接受任务但不能完成的情况。从平均信誉值来看, 未完成任务的数值要明显低于已完成任务, 因此, 任务圆范围内的会员信誉值低会提高任务的未完成率。

4、任务所在区域的性质会对任务的完成产生影响。由于不同位置区域的任务密度不同，会员预订任务限额和信誉值存在差异，导致聚类区域内整体的任务完成度较低，从而影响到单个任务的完成率。

如图 4 所示，分别得到 A、B、C 三个聚类区域的相应数据（见表 2）：

表 2 三个聚类区域的指标数据

| 聚类区域 | A | B | C |
|---------|--------|--------|--------|
| 预订任务限额 | 4.864 | 7.665 | 6.848 |
| 会员平均信誉值 | 173.87 | 346.57 | 267.26 |
| 任务密度 | 176.77 | 184.41 | 339.54 |
| 任务完成率 | 0.35 | 0.93 | 0.56 |

从表 2 中可看出，A 区域的任务完成率远低于 B 区域，C 区域的任务完成率接近 50%。将区域 A 和 B 对比分析，在任务密度相近的情况下，A 区域在会员的能力属性，即会员的预订任务额度和平均信誉值明显低于 B 区域，由此可以推断出 A 区域因总体会员能力较差而降低了总体任务的完成率；将区域 B 和 C 对比分析，虽然 C 区域在会员能力属性上与 B 区域差距较小，但由于任务密度过大，任务间存在竞争关系，导致部分任务因无人接受而未完成，因此，区域内任务密度过大会降低任务的完成率。

5.2 问题二的建模与求解

在问题二中，我们构建了 Logistic 回归模型，以任务定价、平均距离、平均任务限额、任务密度为解释变量，任务完成情况为被解释变量，得出任务定价与其他变量及任务完成概率的函数关系。

接下来从任务的完成率和定价两个方面对原定价方案进行优化，建立以单个任务完成的可能概率为自变量，任务定价和完成率为目标变量的规划模型，得到可以在两个方面同时达到理想值新的任务定价并与原定价方案进行比较。

5.2.1 Logistic 回归模型的建立

在原项目一任务定价方案的基础上，我们发现任务定价最终反映于任务的完成情况，而任务的完成情况不仅受会员位置与任务位置的调和平均距离影响，还与平均任务限额和会员任务密度有关。因此我们把问题一构建的平均任务限额、任务密度两个指标和任务到会员位置的平均距离以及附件一中给出的任务定价四个变量作为解释变量，把完成情况作为被解释变量构建 Logistic 回归模型，其基本形式为：

$$P(Y=1) = \frac{\exp(\beta_0 + \beta_1 l + \beta_2 t + \beta_3 s + \beta_4 X)}{1 + \exp(\beta_0 + \beta_1 l + \beta_2 t + \beta_3 s + \beta_4 X)} \quad (7)$$

其中， β_i 为多元线性回归模型中的回归系数， l 为平均任务限额， t 为任务密度， s 为任务位置到会员位置的平均距离， X 为任务定价；上式表示由 4 个变量影响下，任务完成情况 P 为 1 的概率。

对 (7) 式进行对数变换，得

$$\ln \frac{P}{1-P} = \beta_0 + \beta_1 l + \beta_2 t + \beta_3 s + \beta_4 X \quad (8)$$

此时即可按照多元线性回归的方法得到回归参数，但是对于任务的完成性， P 的取值只有 0 和 1，这就导致 $\ln(P/(1-P))$ 形式没有意义。因此，我们定义单调连续的任务完成概率函数 γ ，令

$$\gamma = P(Y=1), 0 < \gamma < 1 \quad (9)$$

此时，Logistic 模型就可变形为

$$\ln \frac{\gamma}{1-\gamma} = \beta_0 + \beta_1 l + \beta_2 t + \beta_3 s + \beta_4 X, 0 < \gamma < 1 \quad (10)$$

虽然 (8) (10) 两式形式相同，但此时的 γ 为连续函数，只要对原始数据进行合理的映射处理，就能够用线性回归方法得回归系数。我们取分界值为 0.5，

即 γ 到 P 的映射关系为： $P = \begin{cases} 0, \gamma \leq 0.5 \\ 1, \gamma > 0.5 \end{cases}$ 。

认为当任务的完成概率为该值时任务能恰好完成，将其代入程序中得此时的 Logistic 回归模型为：

$$\gamma = \frac{\exp(-9.136 + 0.381l - 0.004t + 10.874s + 0.062X)}{1 + \exp(-9.136 + 0.381l - 0.004t + 10.874s + 0.062X)} \quad (11)$$

通过各个变量的回归系数我们发现，任务完成概率与平均任务限额、平均距离和价格成正相关，与任务密度呈负相关，这与实际情况相符。

5.2.2 定价方案的设计

在设计新的任务定价方案时，我们考虑从任务的完成率和定价两个方面对原定价方案进行优化，即在尽可能提高任务完成率的情况下降低定价，减少任务发布者的成本。

首先我们定义一个期望概率值 M ，令每一个任务的完成概率都为期望概率值，目的是通过调整定价使所有任务的竞争力相同，也就是每个任务被完成的概率相同，则总体任务完成的频率等于单个任务完成的概率。

由 5.2.1 可得，任务的定价可由任务完成概率、平均任务限额、任务密度和任务到会员位置的平均距离四个变量求得，对公式 (11) 变形可得

$$X = \frac{\ln \frac{M}{1-M} - \beta_0 - \beta_1 l - \beta_2 t - \beta_3 s}{\beta_4} \quad (12)$$

由此看出，对于一个确定的任务，其平均任务限额、任务密度和任务到会员的平均距离都已是确定的值，所以任务定价 X 仅为期望概率值 M 的函数，满足 $X = f(M)$ 。我们以 M 为自变量，以通过 M 得出的所有任务的平均价格为一个指标， M 本身，即总体任务完成的频率作为另外一个指标进行规划。通过赋予两个指标权重，得到一个综合评价指标，使二者同时达到较为理想的值，保证在任务完成率尽可能大时平均价格尽可能小，降低任务发布者的成本。

对数据进行归一化处理后，构建任务定价的规划模型：

$$\min F = \omega_1 \bar{X} - \omega_2 M \quad (13)$$

$$s.t. \begin{cases} M \geq 0.5 \\ M \leq 1 \end{cases} \quad (14)$$

其中， $\bar{X} = \frac{\sum X_i}{n}$, ($i=1, 2, \dots, n$)， n 为任务总数。

经过对两指标权重的多次试验，最终确定 $\omega_1 = 0.605$, $\omega_2 = 0.395$ ，通过求解规划模型得到新的任务定价方案。

5.2.3 定价方案的比较

问题一中我们分析得到，附件一中原项目的任务定价规律为：任务位置与会员位置间的调和平均距离越大，任务定价越高。即原定价方案只考虑了定价与任务和会员间距离的关系，忽略了会员的能力属性以及任务所在区域对任务定价的影响。

因此，我们在 5.2.2 中给出了新的定价方案，在考虑到会员平均任务限额、平均信誉值和任务密度的基础上，从任务的完成率和定价两个方面对原方案进行优化，将原任务定价方案和新的定价方案中的任务完成率及任务的平均定价进行比较，数据如表 3 所示：

表 3 定价方案数据比较

| 定价方案 | 原任务定价方案 | 新任务定价方案 |
|-----------|---------|---------|
| 任务完成概率 | 62.5% | 66.3% |
| 已完成任务平均定价 | 69.82 | 63.53 |
| 未完成任务平均定价 | 67.93 | 75.80 |
| 总体任务平均定价 | 69.11 | 68.14 |

由表 3 可以看出,我们设计的新的任务定价方案在任务完成率上较原来的定价方案增长了 3.8 个百分点,在任务完成的数量上有了一定的改善;对于已完成的任务,我们的定价策略在保证其能完成的前提下,任务的平均定价有所降低,为任务发布者减轻了成本负担;对于未完成任务,易知将任务定价适当提高会增加会员对任务的参与度,而我们的定价策略恰好能实现这一点。并且,从总体任务的平均定价来看,新设计的定价方案的任务定价有所降低,同时满足了任务完成率的提高和任务定价的降低,效果较原方案有一定的改善。

对原任务定价方案和新的定价方案中任务的平均定价作直观的折线图如图 5 所示,从图中可以直观的体现出已完成任务和未完成任务在不同定价策略下的任务定价变化,明显设计的新方案更为合理。

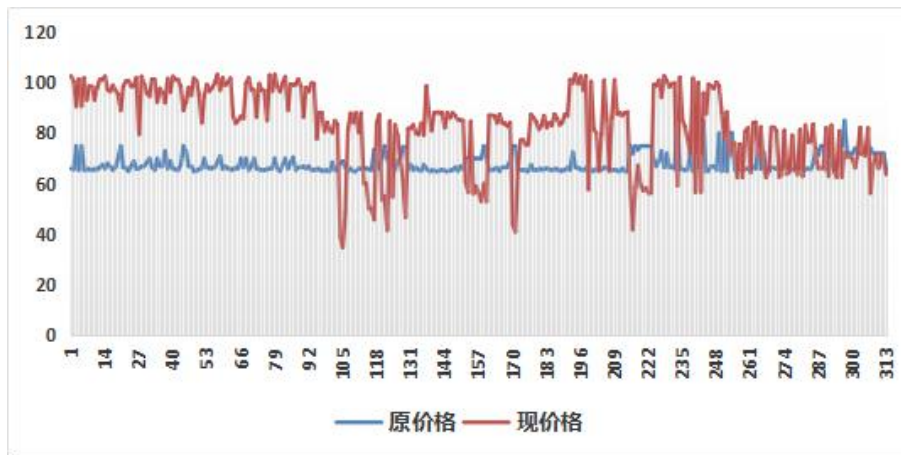


图 5 (a) 未完成任务定价比较

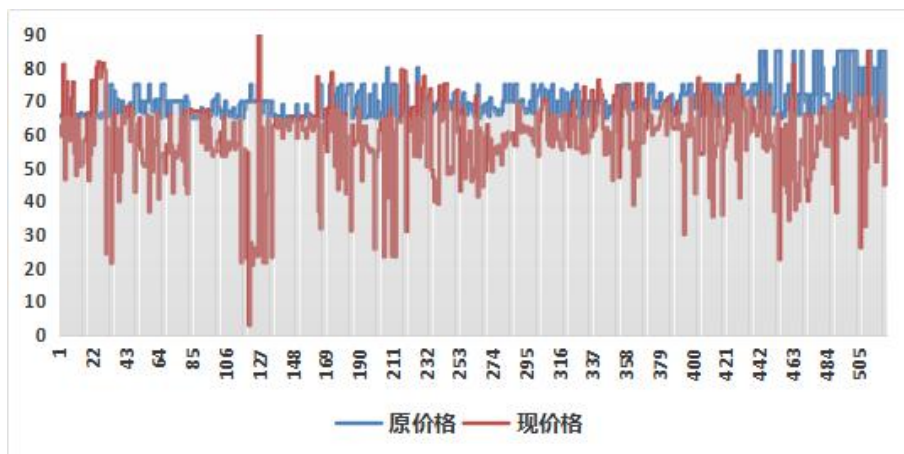


图 5 (b) 已完成任务定价比较

综上所述，我们设计的新的任务方案较原方案提高了任务的完成率，并给出了更为合理的任务定价。

5.3 问题三的建模与求解

在问题三中，我们首先采用 DBSCAN 算法对位置比较集中的任务进行联合打包，得到 438 包任务。然后在问题二模型的基础上，将打包的任务作为一个新任务发布，结合打包的特点对原指标的计算方式进行修改，由此得到打包后新的定价方案。

为了验证修改后定价模型对最终任务完成情况的影响，我们又建立了会员任务选择博弈模型，先从预订时间和信誉值两方面对会员进行排序，然后会员依次遵循四个原则对任务包进行选择，在所有的会员选择完后，观察此时的任务完成情况，探究此定价方案对任务完成情况的影响。

5.3.1 任务联合打包策略

从任务位置来看，任务的位置分布具有不均匀性、存在较为孤立的任务点，而且事先不知道要形成的任务包的数量。所以我们应用 DBSCAN 算法，对位置比较集中的任务进行打包，将其定义为一个新的任务发布。

DBSCAN 算法是一种比较有代表性的基于密度的聚类算法。在本题中，它与划分和层次聚类方法不同，它能够把具有足够高密度的任务区域划分为任务包，并可在有偏差的空间数据库中实现任意区域形状的任务打包。

取打包半径阈值为 10 公里，每个任务包中最少包含 1 个任务，任选一个任务位置点，找出与其距离在阈值范围内的所有任务点，如果附近任务点的数量 ≥ 1 ，则当前点和其附近点构成一个任务包。然后递归，以相同的方法处理任务包中所有没选过的任务点，从而对任务包进行扩展，直到任务包中所有的任务都已选中过。最终我们将所有的任务联合为 438 包，如图 6 所示。

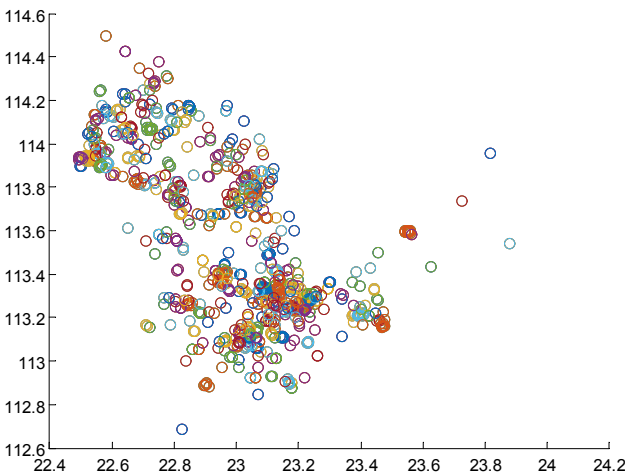


图 6 打包效果图

其中，横坐标表示纬度，纵坐标表示经度，每种颜色代表一个任务包。

5.3.2 修改的定价策略

在问题二建立的 Logistic 回归模型的基础上，结合任务打包处理的特点，对原有解释变量的计算方法进行修改。

首先我们将被打包的一系列任务重新定义为一个任务，将原来任务包中各个任务具有的属性综合为新任务的属性。

其中平均任务限额 l 和任务密度 t 两个变量取包内所有任务的平均，而任务位置到会员位置的距离 s 用回路附加的方法求解，即先在任务包中任取一个任务，计算它到会员的位置，在此基础上，加上任务包中该任务到距它最近的下一个任务的距离，以此类推，直至任务包中的所有任务点遍历完。

$$l = \frac{\sum_{i=1}^m l_i}{m} \quad (15)$$

$$t = \frac{\sum_{i=1}^m t_i}{m} \quad (16)$$

$$s = \frac{\sum_{i=1}^m s_i}{m} + \sum_{j=1}^{m-1} s_{j(j+1)} \quad (17)$$

其中， m 表示任务包中的任务点数， $s_{j(j+1)}$ 表示相邻两任务点间的距离。

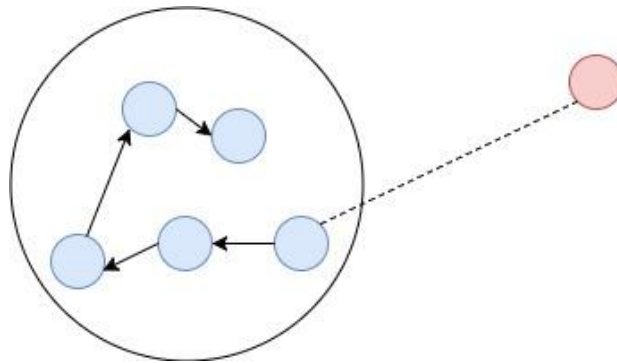


图 7 会员到任务包距离计算示例

将修改后的变量代入公式 (10) (12) (13)，从而得到在打包的情况下的新定价方案。

5.3.3 定价方案对任务完成率的影响

为了评估设计的定价方案对任务完成情况的影响，我们建立了会员在选择任务时的博弈模型。在自助式劳务平台上，会员可以根据任务的相应信息决定是否接受和完成该任务，但由于会员本身在预订任务限额和信誉值上的差异，会员处于同一选择优先级是不合理的。

因此，我们首先对会员的能力属性进行排序，先从会员的预订任务时间方面定出先后顺序，时间较早的给予优先任务选择权。在预订任务时间相同的情况下，信誉值较高的会员可以对任务进行优先选择。

同时，规定会员在选择任务时满足四个原则：

- ①第 i 名会员选择的任务没有被前 $i-1$ 名会员选中过；
- ②总是选择单位距离定价最高的任务包；
- ③不能选择任务数大于其预订任务限额的任务包；
- ④定义会员的最低接受任务定价：任务包的定价除以平均距离，当定价低于其最低接受时不再选择任务。

会员任务选择的博弈模型如图 8 所示：

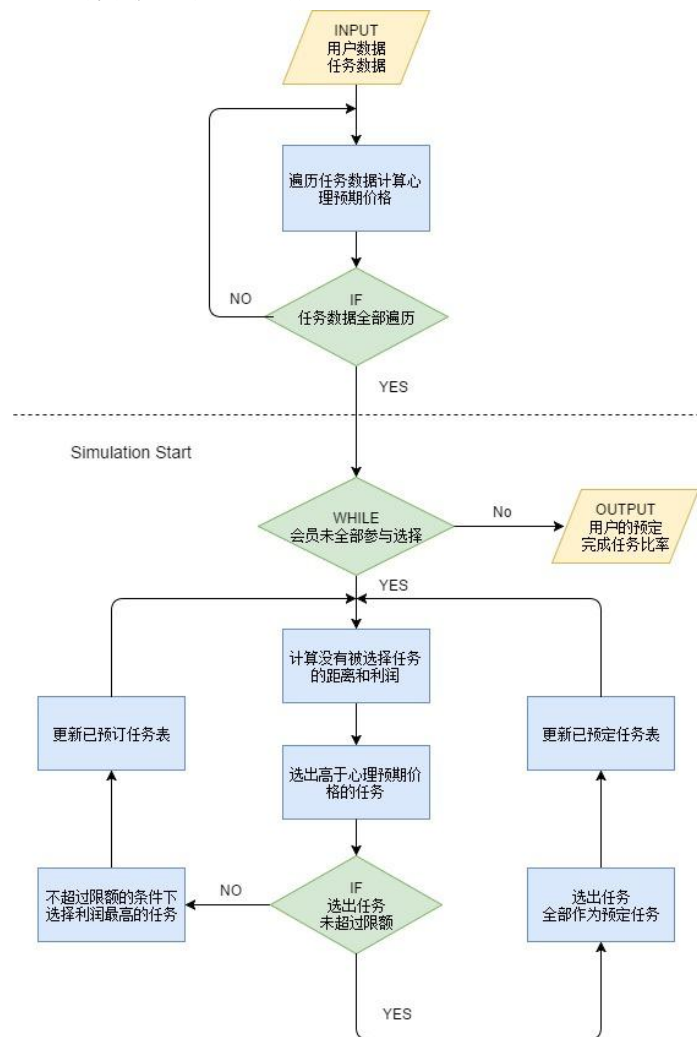


图 8 会员任务选择博弈模型

如图 9 所示,对位置比较集中的任务联合打包后能够使任务的总体平均价格有所降低。在遍历所有的会员后,观察此时的任务完成情况,得到任务的完成率为 85.84%。相对于打包前得到的任务完成率 66.3%,由于将位置较为集中的任务联合在一起,会员可以在不多走较多路程的情况下同时完成几个任务,在提高收益的同时增加了会员的任务参与度,任务的完成情况有了极大的改善,提高了任务完成率。

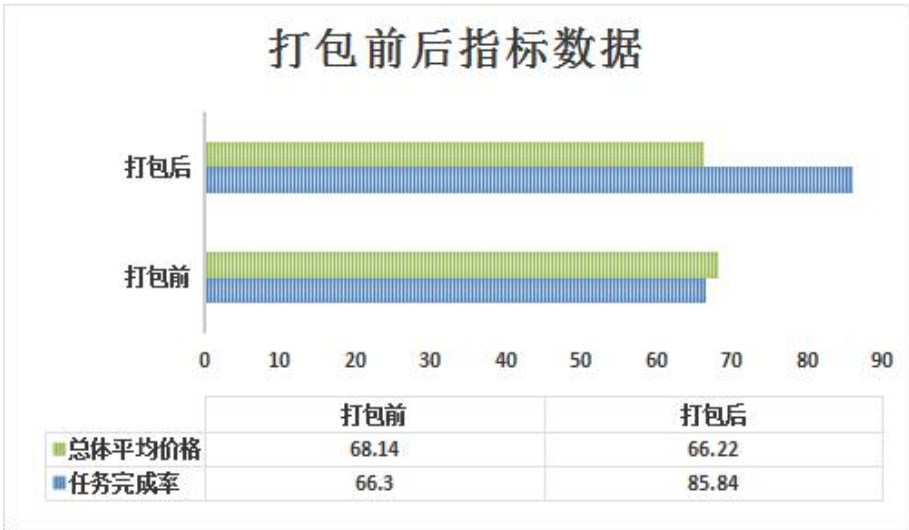


图 9 打包前后任务完成率的对比

5.4 问题四的建模与求解

在问题四中,结合前三问已有的模型和结论,我们先计算出新项目中任务相应的指标数据,用 DBSCAN 算法对位置较集中的任务进行打包,将其代入问题三修改后的定价模型得到任务的定价方案。将此时的定价与通过原有规律定出的价格应用会员任务选择博弈模型分别得到任务的完成情况。同样从任务的完成率和任务定价两个方面对两定价方案进行对比,评价新的定价方案的实施效果。

5.4.1 任务圆的构建

由问题一已得的结果,将附件三中所有的任务位置用 Excel 定位在平面地图上,然后以任务位置为圆心,30 公里为半径做任务圆,在圆内计算平均调和平均距离、平均任务限额和会员平均信誉值。

5.4.2 DBSCAN 打包策略

继续沿用第三问中的 DBSCAN 算法,对位置较集中的任务进行打包。由于附件三中新项目任务数量远多于附件一已结束项目的任务,而且从任务的位置分

布图上看直观看出，新项目任务的分布十分密集，因此，我们取打包半径阈值为 1 公里，最后得到 1273 包任务。

5.4.3 任务的定价方案和效果评价

在问题二建立的 Logistic 回归模型的基础上，结合任务打包处理的特点，对原有的解释变量的计算方法进行与 5.3.2 相同方式的修改。

即平均任务限额和任务密度两个变量取包内所有任务的平均，而任务位置到会员位置的距离用回路附加的方法求得。将修改后的变量代入公式（10）（12）（13），得到对新项目的任务定价方案，计算得此时的任务完成率为 77.69%。

将新项目的任务数据代入到问题一的定价模型，得到一种定价策略，并应用会员在选择任务时的博弈模型计算此时任务的完成率为 52.03%。对两种定价方案中所有任务的定价取平均值，得新项目的定价平均为 67.68 元，原定价模型得到的平均定价为 73.62 元。因此，新的任务定价方案较原方案任务的完成率更高且任务定价更低，这既减小了任务发布者的成本，又提高了会员的参与度，为会员提供更高的利润，方案的实施效果具有显著的提高。

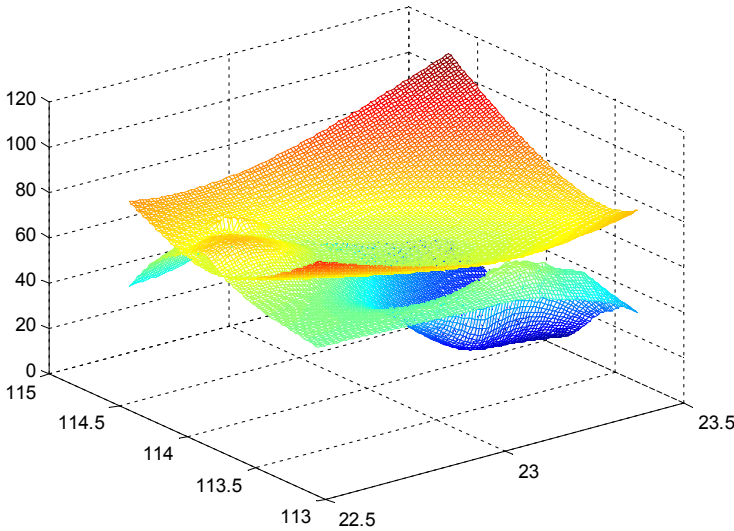


图 10 两定价方案的任务定价对比

图中，底面两坐标轴表示经纬度，纵轴表示任务定价。位于下方的平面描述的是新的定价策略，上方的平面描述原定价策略，可以看出，新的方案中任务定价普遍低于原方案。

六、模型的评价及改进

6.1 模型的评价

6.1.1 模型优点

1、分析任务未完成的原因时，既从单个任务角度对比了已完成和未完成任务的各项指标，又从区域特点进行分析，探究总体区域对任务完成情况的影响。

2、问题二中从任务定价和任务完成率两个方面对原有的任务定价模型进行优化，保证会员利益的同时即能改善任务的完成情况，又能在一定程度上降低任务定价，减少任务发布者的成本。

3、建立了会员选择任务的博弈模型，建立的定价模型能在模拟会员进行博弈决策时得到检验。

4、图像处理与显示方面，我们使用了 Excel-Power Map 与 SPSS 双重作图，直观的反映了任务的位置及完成情况与会员位置的关系，可视性更强，更加清晰、条理和直观。

5、构建的会员任务选择博弈模型具有较强的推广性，可以用于模拟其他网络虚拟产品交易的情况。

6.1.2 模型缺点

1、定价模型只适用于信息较全面的地区，对于会员信息数据不完整的地区难以定价。

2、多元线性回归时缺乏对非线性情况的讨论，有一定的局限性。

3、在问题二评价定价模型效果时，任务完成概率和任务定价两指标权重的确定存在一定的主观性。

6.2 模型的改进

本文在设计任务定价方案时函数关系中参数的意义不明确。在查阅相关文献中，我们发现了一种竞争导向服务产品定价模型，该模型对众包定价具有指导意义，其模型为：

$$\lim_{x \rightarrow \infty} P_n = \frac{\alpha - \gamma + \theta + C\beta + \frac{C\theta}{P_c} + \sqrt{\left(\alpha - \gamma + C\beta + \frac{C\theta}{P_c}\right)^2 + 8C\gamma\left(\beta + \frac{\theta}{P_c}\right)}}{2 + \left(\beta + \frac{\theta}{P_c}\right)}$$

其中服务产品需求函数为： $P = \alpha - \beta Q, \alpha, \beta \geq 0$ 。各参数的含义如下： P_n 是指参考上一期定价 P_{n-1} 而给出的定价， P_c 为其他服务产品的同期平均报价（通常假定为常量）， C 为服务产品的平均成本。 β 表示当前价格对需求的影响权重， γ 表示外包行业平均报价（即其渠道服务产品的同期报价的参考效应）对需求的影响权重， θ 表示近期任务出价的变化（即近期类似任务出价）对需求的影响权重。

由于模型中涉及的大多数参数无法得到，所以我们并没有采用。在条件允许的情况下，该模型可以更客观、准确的得出任务定价。

七、参考文献

- [1] 祁芸. 众包商业模式及其经济学分析[J]. 商业时代, 2012, 34.
- [2] 刘树人. 基于逆向拍卖的采购与定价管理[D]. 湖南：湘潭大学学报. 2013.
- [3] 刘晓钢. 众包中任务发布者出价行为的影响因素研究[D]. 重庆：重庆大学. 2012.
- [4] 卓金武. MATLAB 在数学建模中的应用（第2版），北京：北京航空航天大学出版社，2014.
- [5] 陈胜可，刘荣. SPSS 统计分析从入门到精通（第三版），北京：清华大学出版社，2010.

八、附录

8.1 每个任务指标值的计算

```
mission=xlsread('mission.xls',1,'B2:C836');
vip=xlsread('2.xlsx',1,'B2:E1878');
result=xlsread('mission.xls',1,'E2:E836');
price=xlsread('mission.xls',1,'D2:D836');
for i=1:835
    sum2=0;
```

```

sum3=0;
sum4=0;
sum5=0;
number=0;
for j=1:1877
    if sqrt((mission(i,1)-vip(j,1))^2+(mission(i,2)-vip(j,2))^2)<0.3
        sum2=sum2+vip(j,3);
        sum3=sum3+vip(j,4);
        sum4=sum4+sqrt((mission(i,1)-vip(j,1))^2+(mission(i,2)-vip(j,2))^2);
        number=number+1;
    end
end
for j=1:835
    if sqrt((mission(i,1)-mission(j,1))^2+(mission(i,2)-mission(j,2))^2)<0.3
        sum5=sum5+1;
    end
end
if number==0
    k(i,1)=0;
    k(i,2)=0;
    k(i,3)=0;
else
    k(i,1)=sum2/number;
    k(i,2)=sum3/number;
    k(i,3)=sum4/number;
end
k(i,4)=number;
k(i,5)=sum5;
end
n=0;

```



```

z=0;

for m=1:835
    if result(m,')==0
        n=n+1;
        failed(n,:)=[k(m,:),price(m,:)];
    else
        z=z+1;
        success(z,:)=[k(m,:),price(m,:)];
    end
end

as=mean(failed);
ad=mean(success);

```

8.2 问题二中目标规划模型

```

[p,fval]=fmincon('fun1',0.5,[],[],[],[],0.5,1);
k=xlsread('zhibiao.xlsx',1,'A1:C835');
result=xlsread('zhibiao.xlsx',1,'D1:D835');
price=xlsread('mission.xls',1,'D2:D836');
jw=xlsread('mission.xls',1,'B2:C836');
x1=0;
for x=1:835
    if result(x)==0
        x1=x1+1;
        oldprice(x1,:)=price(x);

newprice(x1,:)=(log(p/(1-p))+9.136-0.381*k(x,1)-0.004*k(x,2)-10.874*k(x,3))/0.062;
        jw1(x1,:)=jw(x,:);
    end
end

```

```

end

old=mean(oldprice);
new=mean(newprice);
y1=0;
for y=1:835
    if result(y)==1
        y1=y1+1;
        oldprice1(y1,:)=price(y);

newprice1(y1,:)=(log(p/(1-p))+9.136-0.381*k(y,1)-0.004*k(y,2)-10.874*k(y,3))/0.06
2;

        jw2(y1,:)=jw(y,:);
    end
end

old1=mean(oldprice1);
new1=mean(newprice1);

```

8.3 任务打包及定价模型修改

```

D = xlsread('1.xls',1,'B2:C836');
canshu=xlsread('zhibiao.xlsx',1,'A1:C835');
D=D';
eps = 0.01;
MinPts = 1;
O = zeros(1, size(D, 2));
C = cell(size(D,2));

d = zeros(size(D, 2), size(D, 2));
for i = 1:size(D, 2)
    for j = size(D, 2):-1:i

```

```

        if pdist(D(:, [i j]))' <= eps
            d(i, j) = 1;
        end
        d(j, i) = d(i, j);
    end
    %d(i,i) = 0;
    if sum(d(i, :)) >= MinPts
        O(i) = i;
    end
end

```

```

k = 0;
Tau = 1:size(D, 2);
while sum(O) ~= 0
    Tau_old = Tau;

    j = 1;
    while O(j) == 0
        j = j+1;
    end

    %ob = O(j);
    Q = zeros(1, size(Tau, 2));
    Q(j) = j;
    Tau(j) = 0;

```

```

while sum(Q) ~= 0
    m = 1;
    while Q(m) == 0
        m = m+1;
    end
end

```

```

end
Q(m) = 0;

if sum(d(m,:)) >= MinPts
    for l = 1:size(d, 2)
        if d(m, l) == 1 && Tau(l) ~= 0
            Q(l) = 1;
            Tau(l) = 0;
        end
    end
end

end

end

k = k+1;
for i = 1:size(Tau, 2)
    if Tau(i) ~= 0
        Tau_old(i) = 0;
    end
    if Tau_old(i) ~= 0
        O(i) = 0;
    end
end

end

C{k} = Tau_old;
end

for i = 1:k
    C{i} = find(C{i}~=0);
    D1 = D(:,C{i});
    scatter(D1(1,:),D1(2,:));
    hold on

```

```

end
D=D';
for i=1:k
    sum1=0;
    sum2=0;
    sum3=0;
    jing=0;
    wei=0;
    fujia=0;
    a=C{i,1};
    r=size(a,2);
    for j=1:r
        sum1=canshu(a(1,j),1)+sum1;
        sum2=canshu(a(1,j),2)+sum2;
        sum3=canshu(a(1,j),3)+sum3;
        jing=jing+D(a(1,j),1);
        wei=wei+D(a(1,j),2);
    end
    if r>1
        for j=1:r-1
            fujia=fujia+sqrt((D(a(1,j),1)-D(a(1,j+1),1))^2+(D(a(1,j),2)-D(a(1,j+1),2))^2);
        end
    end
    final(i,1)=sum1/r;
    final(i,2)=sum2/r;
    final(i,3)=sum3/r+fujia;
    final(i,4)=jing/r;
    final(i,5)=wei/r;
end

```

8.4 会员任务选择博弈模型

```
# Python3程序源代码
# -*- coding:utf-8 -*-

import csv
import math
import time

time1 = time.time()

filename1 = 'chooseorder.csv'      #经度，纬度，限额，会员编号
filename2 = 'tasklist3packed.csv'  #经度，纬度，任务价格，任务编号，
平均距离

tasknumber = 0
phyvalue = 0

with open(filename2, 'r') as taskreader:
    tasks = csv.reader(taskreader)
    headerrow = next(taskreader)
    code = 4
    taskinterest = 0
    count = 0
    out = []
    out.append(code)
    for task in tasks:
        count = count + 1
        try:
            taskinterest = taskinterest + (float(task[2]) / float(task[4]))
        except ZeroDivisionError:
            code = code + 1
            out.append(code)
```

phyvalue=taskinterest/count

```
taskcheck=[]

with open(filename1,'r') as chooseorder:
    choosepeople = csv.reader(chooseorder)
    headerrow1 = next(chooseorder)
    count1 = 0
    for chooseperson in choosepeople:
        wantlist = []
        finallist = []
        count1 = count1 + 1
        with open(filename2,'r') as taskreader:
            tasks = csv.reader(taskreader)
            headerrow2 = next(taskreader)
            for task in tasks:
                distance =
math.sqrt((float(chooseperson[0])-float(task[0]))**2+(float(chooseperson[1])-float(tas
k[1]))**2)

                interest = float(task[2])/distance
                if interest>int(out[0])*phyvalue and (task[3] not in
taskcheck):

                    wantlist.append([task[3],float(task[2])])

#print(len(wantlist))
wantlist.sort(key=lambda x: x[1])

if len(wantlist) > float(chooseperson[2])*count:
    chooselist = wantlist[-int(float(chooseperson[2])+1):]
    chooselist = wantlist[-2:]
else:
```



```

        chooselist = wantlist[:]

        #print(len(chooselist))

        for choosed in chooselist:

            print('Task '+choosed[0]+' has been chosen by
'+chooseperson[3])

            taskcheck.append(choosed[0])

    print()

    print('Success:'+str(len(taskcheck)))

    print('Failed:'+str(count-len(taskcheck)))

    print('Success Rate:'+str(len(taskcheck)/count*100)+'%')

    time2 = time.time()

    print('Used Time:'+str(time2-time1)+'s')

    input('Press any key to continue')

```