

“拍照赚钱”的任务定价

摘要

本文针对基于众包任务的定价问题,引入会员期望价格和任务定价的博弈关系,综合考虑有效收益率和任务执行情况,建立定价最优控制模型。

问题一中定义会员期望报价,结合顾客价值模型及感知价值定价法提出可能影响任务完成率的潜在因素并进行相关性分析,比较其相关系数得出任务未完成的原因可能有以下几点:任务点矩形辐射范围内的会员数量、限额较大,或者是周围任务点较密集,周围任务执行情况较低。

问题二在制定定价最优方案的过程中,我们采用最优化方法进行求解,运用静态博弈论原理定义 0-1 选择函数,给出受任务定价及距离影响的会员报价函数和受会员期望价格影响的定价函数,以完成率和任务支出之比最大为优化目标,综合分析平台期望收益和用户资源利用率,得到定价因素为 0.3~0.5 范围内,用户满意度在 0.1~0.2 范围内的相对最佳方案。我们选择满意度为 0.1 的定价因素为 0.5,得出新定价方案下任务完成率为 76.37%,高于历史任务完成率 38.90%;新定价方案下平台任务定价均值为 69.15 低于历史平台任务定价均值 69.82,说明此定价方案优于原定价。

问题三针对任务分布集中导致部分区域会员竞争激烈、降低会员积极性的现象,考虑将任务联合发布。我们建立的基于打包任务数量和会员优先度的打包模型,根据任务是否打包对任务进行重新考虑,利用会员配额限制,降低高配额会员间的竞争,让平台能够给出更合适的价格;同时又激励低配额会员去选择其他同类型任务,提升任务整体的完成率。得出结果为:任务完成率为 83.74%,平台任务定价均值为 68.21 元,均优于第二问数据,表明包策略能有效改善任务总体执行情况,提升平台总体收益。

问题四在第二问的报价模型基础上,通过添加历史任务报价数据改进会员期望价格函数,并修改有效交易下的平台支出这一优化目标得到新的报价函数。并与原始定价比较进行市场反馈检验,能够发现完成率有着显著提高,完成任务的平均报价也低于历史值,平台收益效率显著提高,模型改进效果明显。

关键词: 众包任务定价 期望价格 静态博弈论 相关性分析 定价最优控制模型 期望收益 任务完成率 任务打包模型

一. 问题分析与重述

问题一分析：探寻已给定任务的定价规律，分析任务是否完成与哪些因素有关。由于根据实际情况我们可以提出任务定价的影响因素，利用贪心算法求得使所有相关因子与定价之间相关系数最大的半径 R ，由相关系数分析定价规律；利用主成分分析法找出任务完成度的潜在影响因素及其相关性和显著性。

问题二分析：基于附件一的数据给出新报价模型，并与原始方案比较。通过分析原有方案得知其执行程度以及报价都有待优化，于是构建平台与会员间的价格博弈关系，让平台收益最大化。分析每个会员对于任务的期望价格，该价格仅与距离、类似任务的报价有关；预估每个会员的期望价格后，平台给出最终报价值，如果该报价高于会员的期望价格，那么会员选择接受任务，反之则拒绝。目标函数是让完成率和平台收益尽可能高，并通过会员的配额、参数范围进行限制，得到最优报价函数，最后分析各指标与原方案的差异。

问题三分析：考虑集中任务联合打包发布，修改前面定价方案，讨论最终任务完成情况。对会员进行优先度重排，通过任务与会员的两两相关性构建 0-1 相关矩阵，由打包任务数量按照会员优先度对任务进行打包，打包任务总价等于各任务单价总和减某个参数，去掉打包任务剩下的任务考虑为问题二模型，比较此定价方案下与单独分布下的总任务完成率与总任务价格变化情况。

问题四分析：基于附件三的新数据，给出自己的定价方案，并预估它可能的实施效果。先根据问题三的结论，对部分任务做联合发布处理，再依靠问题二中给出的报价策略模型，改进会员的期望价格函数，该函数仅与新任务点附近历史任务报价、距离有关；然后根据第二问中求得的最优参数，给出新的报价函数，得到每一份任务的定价；最后估算该报价模型下可能得到的市场反馈，并与原始任务定价做对比。

二. 模型假设

模型建立的前提是平台与用户之间已经形成一个成熟的市场，影响交易的因素有：

1) 平台外部因素：

- ①任务供大于求，平台有一定数量的待完成任务，且买方仍可以从较低的定价中或益。
- ②任务属性：不考虑复杂供应市场，任务具有非关键性且标准化程度低，在任务完成复杂度上只有距离因素的差别，即在用户辐射范围之内所有任务的难易度可做无差别对待。
- ③竞争市场：用户具有足够程度的竞争力，且用户数量越多竞争越激烈，市场价格也随之降低，平台在此次交易中可获得更高的利润。

2) 平台政策因素：

- ①收益定价：平台将期望收益作为分配任务时主要考虑的因素，忽略交易过程中产生的成本，兼顾其他收益如交易效率的提高，任务参与度的提升。
- ②价值预估：对任务成本和用户期望报价应做出充分的估计，避免定价不合理导致与用户期望不符或亏损。
- ③客户满意度的提升：适当进行激励政策（比如任务打包），以实现用户资源利用最大化。

三. 背景分析

3.1 顾客价值模型

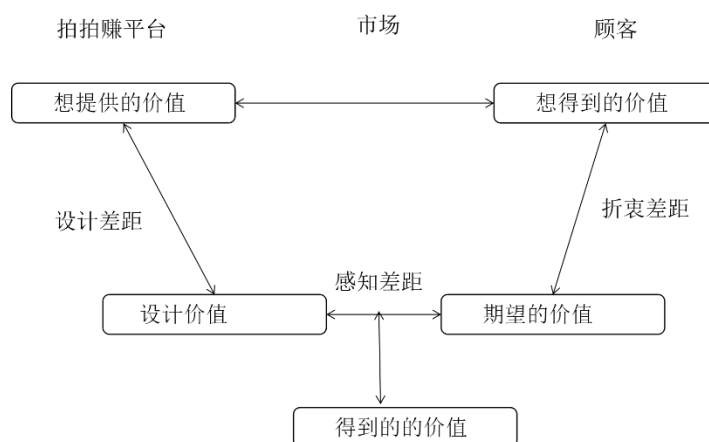


图 1: 顾客价值模型图 (Jeanke, Ron, Onno)

对顾客而言, 顾客从自身角度出发希望获得的是“想要得到的价值”。由于平台与用户之间存在信息不对称, 平台对用户期望价值的分析未必客观准确, 所以“想提供的价值”与顾客“想得到的价值”之间存在“信息差距”。顾客的主观性价值感知, 使“期望价值”与设计价值间出现“感知差距”。当顾客使用产品后, 所“得到的价值”与期望价值之间的差距为“满意差距”。顾客是否选择领取任务与平台是否将此会员视为潜在客户都是通过设计价值和期望价值相比较得出的。通过缩小各个差距, 平台可以提供真正为顾客所需的价值。

3.2 电子逆向拍卖 (ORA)

随着信息技术的快速发展, 越来越多的企业开始运用此种新型采购方式, 它对企业电子采购和供应链管理产生深远的影响。通过逆向拍卖, 采购商可以提高分配效率, 降低采购成本, 供货方可发挥控制优势, 达到利益最大化。

Priceline是率先推出逆向拍卖的网站, 即顾客递交出他想要的物品/服务, 以及愿意为此支付的价格, 由Priceline在提供此物品/服务的报价单搜索是否有愿意以此价位出售的卖方。当顾客的报价高于卖方的保留价时, 他们成交, 同时Priceline收取固定交易费。

3.3 静态非合作博弈及静态均衡行为分析

众包市场上有更多的机会主义动机, 任务交易完成之前, 会员都不知道其它参与竞争者的位置, 配额, 预估价格等情况, 只能根据本次拍卖收益和其他任务定价决定。初次参与者根据自己偏好, 以前参与者可根据自己之前的历史收益来决策, 但本题中并未给出, 所以假设参与者在一次博弈中均以达到各自利益最大化为目的, 不考虑用户满意度对本次的影响。而且在本次博弈结束后, 其自身的信息 (位置, 配额) 也不会改变。

经典的博弈论要求用户与平台都是完全理性的, 但是实际情况是受到认知能力和对以往交易的满意度的影响, 双方存在有限理性博弈, 即用户的期望价格在长期看来会有所变化。此时除了追求利润最大化之外还应考虑客户满意度以建立长期稳定合作, 避免机会主义动机过强形成交易不良状态。

四. 任务参与度影响因素分析

4.1 前期准备

有效商圈：以任务发布点为中心沿一定的方向和距离扩展，能够有效吸引参与者完成任务的地理区域。（有效：针对会员而言，会员能够并愿意领取任务）

会员辐射范围：与预定任务限额和距离有关，能够接受任务的地理区域，预定任务限额越大，辐射范围越广。

为了更好地刻画距离和辐射圈范围，我们将附件中所给的地理坐标利用高斯—克吕格投影转换为公里网（椭球参数：北京/54,6度带，投影带序号：16）。

完成任务的难易程度会影响参与者的兴趣，本题任务完成的难易度体现在距离上，由成本加成定价可知距离越长意味着工作量和完成成本的增大，则参与者的积极性随之降低。我们可以认为位于会员辐射范围之外的任务点被完成的可能性为0，因此我们只对会员辐射范围之内的各个任务点进行完成度分析，并且可认为辐射范围内的任务在距离无差别。

4.2 分析与假设

4.2.1 提出可能影响完成度的潜在因素

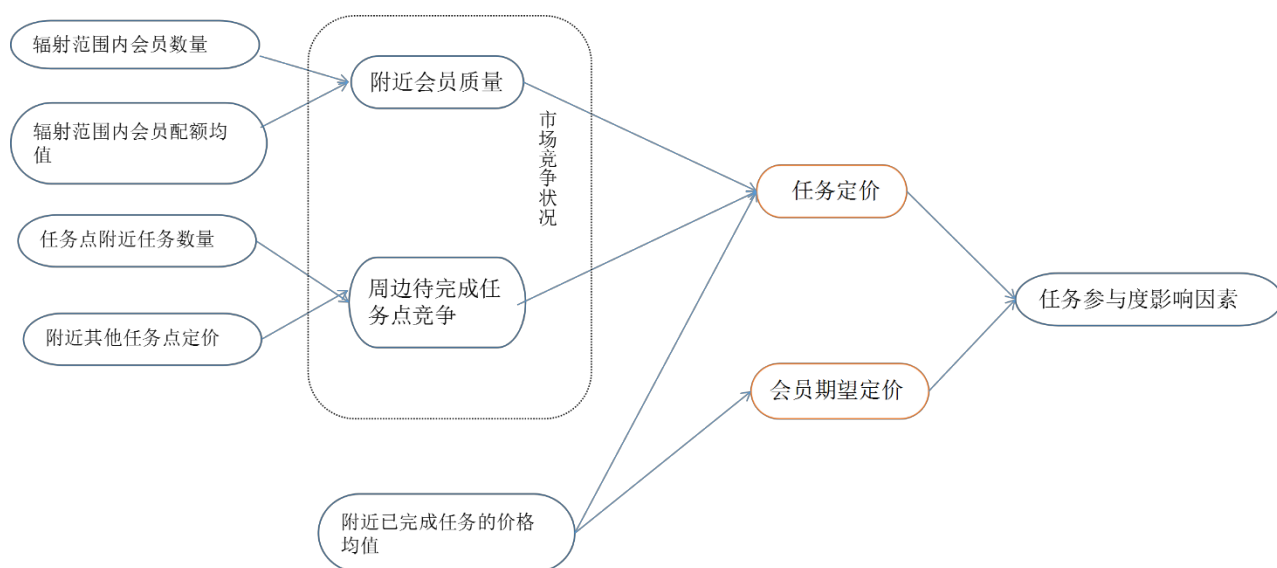


图 2：影响任务完成度的因素分析图

从会员的角度出发，任务发布之后是否会被完成是由发布时任务的定价和会员期望定价共同决定的，会员期望定价又受历史价格影响。其中任务发布时的定价和距离是任务自身特征。

对于本题而言任务的完成难度（技术要求，工作量）以及任务参数（完成期限）均未给出，在此我们可以认为任务之间的复杂度无差别。所以在同样的会员辐射范围内，未被完成的任务便可认为是由于定价不合理导致的。

据了解，任务发布时的定价受市场竞争状况的影响，由周边会员质量和任务点分布，出价情况共同决定。

任务发布之前，公司一般会根据前期市场情况定价同时分析这一任务的市场竞争状况。

Cater 指出当同类任务市场标价较高时，会员将会面临更高的机会成本，同时失去了一些获得高额报酬的机会，转而参加其他期望收益更高的任务ⁱ。因而有如下两个假设

假设一：任务发布之后，任务点附近未完成任务的平均出价附近和已结束任务的定价均值与任务参与度呈正相关。

假设二：任务点附近未完成的任务数量与任务完成度呈负相关。任务附近的会员数量的增多，以及会员配额限度增大会引起会员之间竞争加剧，市场价格有下降的趋势，从而不满足会员期望定价，造成交易失败。因而有

假设三：任务点辐射范围内的会员数量与任务完成度呈负相关。

假设四：任务点辐射范围内的会员配额限度均值与任务完成度呈负相关。

4.2.2 矩形建模法与贪心算法

以商圈内的热点为中心通过设置半宽和半高矩形来表示商圈在坐标轴上位置的方法叫做矩形建模法。相较于圆形建模法，矩形建模法可以通过会员的坐标来判断其是否在任务点的商圈范围内，使得计算效率更高。

利用贪心算法确定矩形商圈的边长：由于各个任务点在任务复杂度上并无差别，其辐射范围（商圈大小）可以人为设定一个值 R ，单位为公里，使得在此 R 值下，所有影响完成度因子的相关系数之和最高，通过一系列局部最优的选择得到整体最优解。利用贪心算法确定矩形商圈的边长：由于各个任务点在任务复杂度上并无差别，其辐射范围（商圈大小）可以通过贪心算法遍历 $[0.5, 10]$ 中以 0.5km 为间隔的一系列 R 值，通过一系列局部最优的选择得到整体最优解，使得在此 R 值下，所有影响完成度因子与定价的相关系数之和最高。

4.2.3 相关性分析确定影响因子的相关系数

将上述所有可能的影响因子列出，根据主成分分析法原理将特征值大于 1 的因子提取为主要影响因素，主要评价步骤有：

(1) 输入样本集并进行中心化： $X_i \leftarrow X_i - \frac{1}{m} \sum_{i=1}^m X_i$

(2) 计算样本的协方差矩阵： XX^T

(3) 对协方差矩阵做特征值分解

(4) 取最大的 d' 个特征值所对应的特征向量 $w_1, w_2, \dots, w_{d'}$ ，输出投影矩阵 $W =$

$(w_1, w_2, \dots, w_{d'})$

(5) 从最近重构性出发设置重构阈值 $t=80\%$ 然后选取式下式成立的最小 d' 值

(6) 确定主成分，然后与任务完成情况进行相关性分析。

ⁱ 李荣喜.2006. 基于价格参考效应的消费者需求和产品定价模型[J]. 管理评论.(11):49-51.

表一： 任务完成度潜在影响因素

维度	潜在影响因素	有效 N	极小值	极大值	均值	标准差	相关系数
任务参数	周围任务数量	835	0	12	2.5198	2.3902	-0.1017
	周围任务均价	835	0	85	54.2229	27.8212	0.0346
	周围任务完成度	835	0	1	0.6232	0.4181	0.7658
	周围会员数量	835	0	77	14.1545	13.0694	-0.2731
	周围会员配额	835	0	701	101.5593	109.2739	-0.1391

由相关性分析结果可知：若任务点辐射范围内的其他任务参与度较高，那么此任务被完成的可能性大致在 0.7658；周围任务均价提升 1, 则此任务完成度将提高 0.0346；而周围任务数量每增加 1 个，任务完成度会降低 0.1017；辐射内会员数量和配额每增加 1，完成度则会分别降低 0.2731 和 0.1391。

从而可分析出任务未完成的原因可能有以下几点：任务点矩形辐射范围内的会员数量和配额较高，或者是周围任务点较密集。

五. 定价最优控制模型

5.1 分析与假设

1. 会员在其辐射范围内对任务的期望价格是关于距离和周围任务点定价的函数。且根据规定，会员领取的任务数量不能超过其预定任务限额。

2. 会员可确定一个可接受定价，称之为期望价格，任务点在其矩形辐射范围（商圈）内会有 n 个会员产生 $d_1, d_2, d_3 \dots d_j$ 共 j 个期望报价。如果任务定价不低于期望价格，则会员接受平台对此任务点的定价。

3. 交易的过程实际上是会员与平台博弈的过程，平台只知道会员的期望价格是任意一个在区间 $[-a, a]$ 的随机变量，而会员只知平台发布任务的成本为独立的随机变量且成本应小于定价，所以不用考虑任务发布成本。只需让平台合理定价，使利润最大化。

4. 要同时满足最大化期望收益和最大化任务完成率，使得会员在给出合理期望报价的情况下效用同时达到最大。

5.2 符号说明

符号说明	
m	任务数量
n	会员数量
t_i	某一任务
c_i	任务 t_i 给出的定价
v_j	某一会员
d_j	会员 v_j 对任务 t_i 的期望价格
ρ_1	用户满意度
ρ_2	定价参数
w	任务 t_i 完成率
P	平台支付给成交者的支出费用

5.3 多元优化模型的建立

5.3.1 模型分析

最优控制模型是由以下六个部分构成的：完成度的选择函数，任务量的约束条件，任务的利润函数，完成率及优化函数，会员的报价函数以及评价函数。

任务点总数为 m ，某一任务 t_i 定价为 c_i ，会员总数为 n ，其中会员 v_i 对于任务 t_i 的期望价格为 d_j 。

1. 选择函数：

交易的过程实际上是会员与平台博弈的过程，当会员可接受平台定价时，将成为该任务的潜在领取者，对应的选择函数值为 1；当平台报价低于会员期望报价是，会员对于领取此任务的意愿很低，对应的选择函数为 0，则有 0-1 函数 $\text{choose}(t_i, v_j) = \begin{cases} 1, c_i \geq d_j \\ 0, c_i < d_j \end{cases}$ ，并以所有选择函数值和最大为优化目标[1]：

$$\max_{1 \leq i \leq m} t_i = \sum_{j=1}^n \text{choose}(t_i, v_j)$$

2. 约束条件：

会员 v_j 对于所有 $\text{choose}=1$ 任务不超过最大任务量。

3. 会员的报价函数：

在辐射区域内，会员价格期望过高会导致对定价不满意，任务参与度会随之降低。由于用户报价受周围其它任务定价及距离的影响，可将

$$d_j = \rho_1 \max_{i \leq i \leq m} \{v_i\} + (1 - \rho_1) \frac{\sum v_i}{k} + \mu x \varepsilon \quad 0 \leq \rho_1 \leq 1$$

其中 k 为会员辐射范围内的任务点数量， μ 为服从 $N(0, 1)$ 的标准正态分布， $[-1.96, 1.96]$ 为其置信水平为 95%的置信区间，仅考虑 μ 落在此区间内的取值。 x 为会员与任务之间的距离，取值区间为 $[0, R]$ ， $R=2.5$ (公里)。 ε 取值为 0.5~0.8 之间的数，对价格形成微小扰动。

4. 任务定价函数[2]：

为了防止只追求过高的完成度而使平台定价过高，从而收益损害收益，将定价限定在

会员期望价格平均值与最高值之间。

$$c_i = \rho_2 \max_{1 \leq j \leq n} \{d_j\} + (1 - \rho_2) \frac{\sum d_j}{l} \quad 0 \leq \rho_2 \leq 1$$

其中 l 为任务点辐射范围内的会员数量， ρ_2 为定价参数。

5. 完成率：

$$w_i = \frac{\text{num}(\text{choose} == 1)}{R_f}$$

其中 R_f 为任务点辐射范围内所有潜在用户， $\text{num}(\text{choose} == 1)$ 为任务点辐射范围内接受任务 t_i 定价的会员总数。

6. 任务的支出：

$$P = \sum_{i=1}^m c_i$$

其中 i 为被会员接受的任务。

7. 优化目标：

$$\text{Target} = \max \frac{w}{P}$$

其中 w 任务完成率， P 平台支付给成交者的总消耗。

表 2：用户满意度与定价因素对任务完成率、平台总消耗的影响 1

用户满意度	定价因素	完成率	平台总消耗	优化目标
0	0	54.80%	5.6345E+04	9.7258E-06
0	0.1	61.63%	5.6475E+04	1.0912E-05
0	0.2	67.26%	5.6605E+04	1.1882E-05
0	0.3	71.55%	5.6734E+04	1.2611E-05
0	0.4	75.25%	5.6864E+04	1.3233E-05
0	0.5	77.93%	5.6994E+04	1.3673E-05
0	0.6	80.35%	5.7123E+04	1.4065E-05
0	0.7	82.33%	5.7253E+04	1.4380E-05
0	0.8	84.12%	5.7382E+04	1.4659E-05
0	0.9	86.10%	5.7512E+04	1.4971E-05
0	1.0	97.96%	5.7642E+04	1.6995E-05
0.1	1.0	97.96%	5.8503E+04	1.6745E-05
0.2	1.0	97.96%	5.9424E+04	1.6486E-05
0.3	1.0	97.96%	6.0373E+04	1.6227E-05
0.4	1.0	97.96%	6.1329E+04	1.5974E-05
0.5	1.0	97.96%	6.2289E+04	1.5727E-05

由上可以看出，如果想让平台的收益最大，那么应该尽可能的选择比较大的定价因素与比较小的用户满意度。但同时考虑到外包社区持续性运转、以及用户实际能够完成的任务量受会员配额的限制，知道我们不可能选取定价因素为 1，用户满意度为 0。具体来说，如果平台给出的报价仅仅只能够满足会员比较低的期望，那么可以认为会员对于任务是“勉强接受”，长此以往会严重打击会员对于接任务的积极性，导致人才流失，没有了稳定的外包会

员数量,外包交易无法良性运转,外包平台的任务执行度则会大打则扣,公司利益受到威胁,

甚至有亏损的可能性;如果报价平台选取很高的定价因素,引起会员争相选择,虽然执行度得到较大的提升,但有可能用户选择的任务很多,逼近自己能够接受的配额任务限制,也便无法保证任务完成质量,同样也会对外包公司利益造成损失。

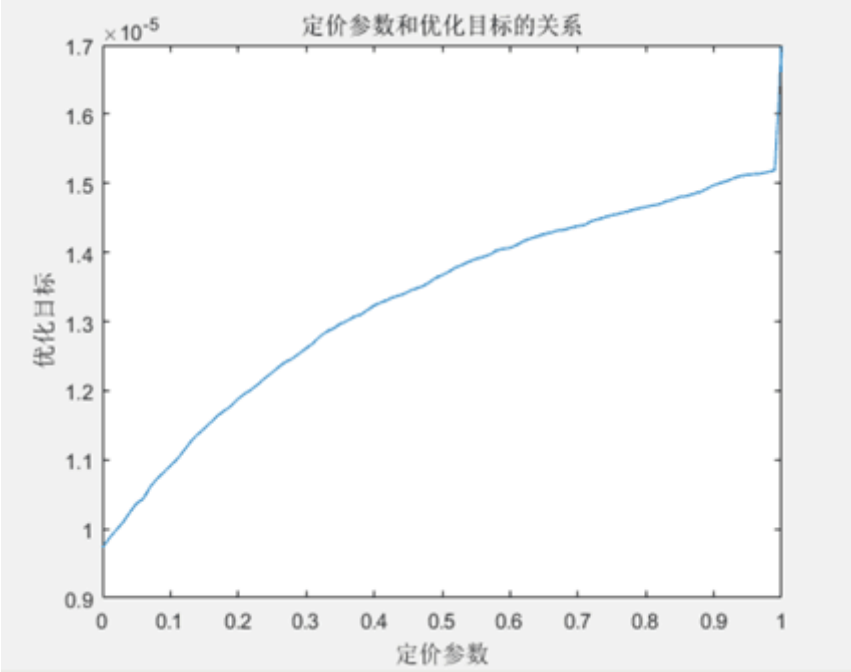


图 3: 定价参数和优化目标的关系图

表 3: 用户满意度与定价因素对任务完成率、平台总消耗的影响 2

满意度	定价因素	完成率	平台总消耗	优化目标	有效支出
0.2	0.3	67.91%	5.8148E+04	1.1679E-05	69.6380
0.2	0.4	72.22%	5.8330E+04	1.2382E-05	69.8563
0.2	0.5	74.89%	5.8512E+04	1.2798E-05	70.0746
0.2	0.6	77.41%	5.8695E+04	1.3189E-05	70.2929
0.1	0.3	69.81%	5.7432E+04	1.2155E-05	68.7811
0.1	0.4	73.49%	5.7585E+04	1.2762E-05	68.9643
0.1	0.5	76.37%	5.7738E+04	1.3228E-05	69.1475
0.1	0.6	79.08%	5.7891E+04	1.3661E-05	69.3808

因此选择满意度为 0.1, 定价因素为 0.5, 为比较优秀的结果。

经过进一步的计算知道, 当定价因素为 0.3~0.5 范围内, 用户满意度在 0.1~0.2 范围内时, 既能达到一个比较好的任务完成率, 同时用户能够选择的任务数量减少, 受到了配额的限制, 保证了任务完成的质量; 还能让会员对价格有一个比较好的满意度, 利于维持外包社区长时间、持续性的运转。

5.3.2 结果分析

在微观经济学中，“拍拍赚”平台与会员之间形成市场当定价参数趋近于1时近似于一个完全竞争市场，此时 $c_i = \max d_j, d_j = \frac{\sum v_i}{k}$ ，资源利用效率最高，当其均衡条件满足边际收益=边际成本时会使得亏损值最小。但从长期来看，在完全竞争的情况下，由于任务点数量多单独每一个任务点的增减对市场价格水平的影响可忽略不计，此时的均衡价格等于边际成本，利润将趋于0。

六. 基于会员优先度的打包模型

6.1 建立关系矩阵

我们先通过会员预定任务开始时间对会员进行优先度排序(时间相同按照预定限额大小排序，预定限额相同按照信誉度大小排序)，每位会员对应一个序号，此举可以让任务完成质量高并且效率高的人更大几率获得自己想要完成的任务，正反馈调节促使排序靠后的会员更高质量的完成任务以提升排序，每一个任务有一个自然编号。

根据我们第二问的假设，会员对某一任务的预估价与会员辐射范围内任务数及其历史定价以及会员与任务之间的距离有关，我们可以定义任务与会员相对应的关系矩阵，即

$$B = (b_{ij})_{M \times N}, \text{ 其中 } \begin{cases} b_{ij} = 1, & \text{第 } j \text{ 号会员处于第 } i \text{ 个任务辐射圈内} \\ b_{ij} = 0, & \text{第 } j \text{ 号会员处于第 } i \text{ 个任务辐射圈外} \end{cases}, 1 \leq i \leq M, 1 \leq j \leq N,$$

M 表示总任务个数，行由任务号有序排列， N 表示总会员数，列由会员编号排序。再对每一列进行约束，当 $\sum_{i=1}^M b_{ij} > \alpha_j$ 时，其中 α_j 表示第 j 号会员的预定限额。我们创建一个关于该

$$\text{会员的新的向量 } \beta_j = \{\beta_{jl}, 1 \leq l \leq M\}, \text{ 其中 } \begin{cases} \beta_{jl} = d_l - c_{jl}, & b_{lj} = 1 \\ \beta_{jl} = -100, & b_{lj} = 0 \end{cases}, d_l \text{ 表示项目给任务 } l$$

定价， c_{jl} 表示 j 会员对任务 l 的预估价，然后找出 β_j 中的 $\frac{\alpha_j}{M}$ 分位数 $P_j[3]$ (指 β_j 中大于等于 P_j 的

$$\text{元素个数与 } M \text{ 之比为 } \frac{\alpha_j}{M} \text{ 的数)，取 } \gamma_j = \{\gamma_{jl}, 1 \leq l \leq M\}, \text{ 其中 } \begin{cases} \gamma_{jl} = 1, & \beta_{jl} \geq P_j \\ \gamma_{jl} = 0, & \beta_{jl} < P_j \end{cases} \text{ (不考虑多}$$

个值与 P_j 相等的情况)。考虑分位数的意义是每一个会员在多个任务选择中作出对自己收益最大的选择方案。对 B 做变换： $b_{ij} = b_{ij} \times \gamma_{ji}, 1 \leq i \leq M$ 。得到新的关系矩阵 D 使得每一个会员所辐射的任务的个数不超过自己的预定限额。

6.2 建立打包模型

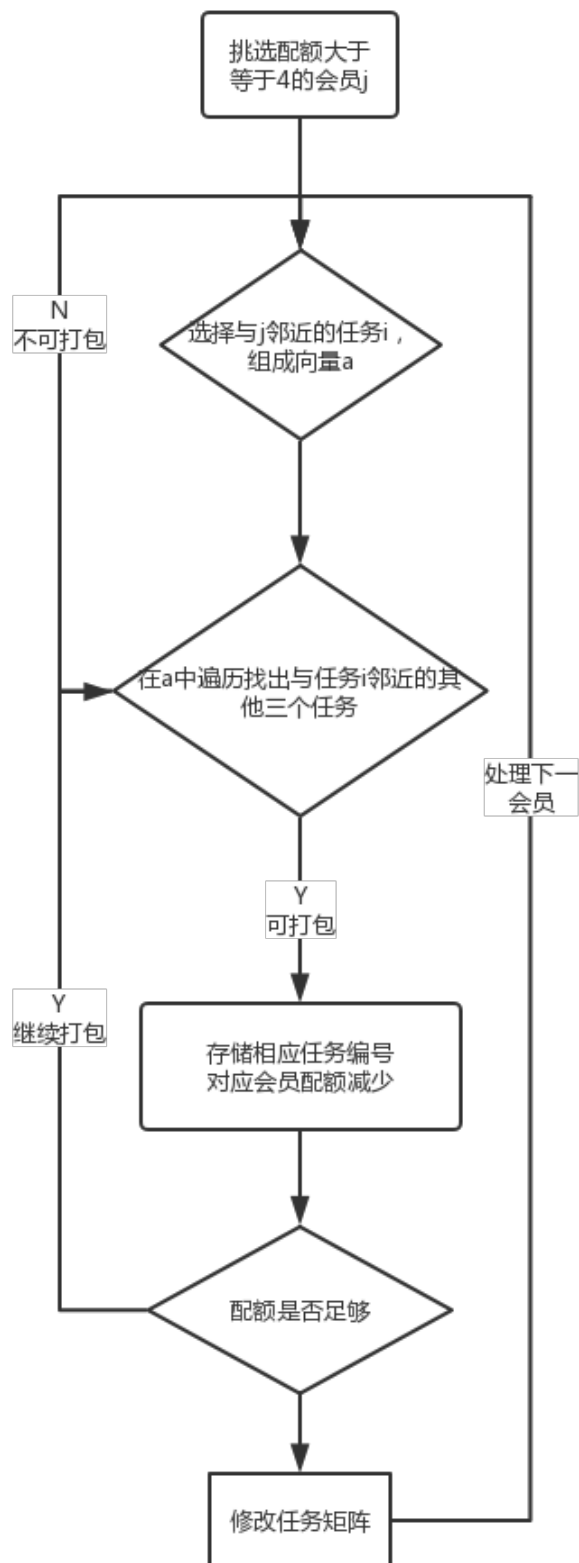


图 4: 打包模型流程图

任务是否打包与其密集程度有关，由第二问的假设，判断一个任务是否可以打包由这个任务辐射圈内的任务数量决定，通过我们模拟分析，单一任务周围任务数的中位数为 2，会员预定限额的中位数为 3，如果我们选择打包数为 4，那么排除了大部分会员参与任务的竞争，保证了排序相对靠前的那部分任务完成质量高效率高的会员的利益，让大部分会员为了能够上升排名而愿意去做一些性价比更低的任务，除此，打包数为 4 能使会员辐射任务范围包含全部打包的任务（由会员辐射范围公式求得）。所以选择打包数为 4 较为合适，即当第 i 个任务的周围任务数量 $\delta_i \geq 3$ 时，该任务可以进行打包。如何进行任务的打包呢？

从会员优先度进行考虑，如对第一位会员进行分析，若该会员任务预定限额小于 4，则对下一位会员进行判断，直到找到某一位会员的任务预定限额大于等于 4，此时，按任务号从小到大依次判断，当 $b_{i1} = 1$ 且 $\delta_i \geq 3$ 时，对任务号从小到大依次筛选满足条件为：

$$(x_i - x_j)^2 + (y_i - y_j)^2 \leq R^2 \text{ 的任务，若有多个任务满足上述条件则选择任务号小的三个任务与第一个任务进行打包。}$$

以上四个任务对应四个序号，设第一个序号为 s_1 ，对其后的会员进行逐一判断运算，若 $b_{s_1 j} = 1$ ，则 $\alpha_j = \alpha_j - 4$ ， α_j 为新矩阵的第 j 号会员的预定限额，再把 B 矩阵的对应四行的删掉，此时 $M = M - 4$ ，得到 $B = (b_{ij})_{M \times N}$ 的新矩阵。重复上述步骤，最终能得到一个新矩阵 $B = (b_{ij})_{T \times N}$ ，这个新矩阵表示每一位会员不能选择打包方案，

对任务的选择与第二问模型相同，对此矩阵对应参数运用第二题中所给模型求得最优解（即得到最小的成本和最大的任务完成率）。

求得最终任务完成率为 83.74%，平台任务定价均值为 68.21 元。

6.3 模型分析

我们对所有打包的任务进行分析，根据经济学原理可知打包后任务总价可适当小于打包前各任务单价总和，用各任务单价总和减去 $\varepsilon\sigma$ ， ε 为相对系数均值（多次取值求最优解所得）， σ 为服从参数为 1 的泊松分布。因为对于会员而言，竞争并完成打包任务所付出的劳动会相应减少，取得的总收益比原来增加。而且在运用打包策略后，预定限额低的会员无法进行竞争打包任务只能选择其他任务，等价于使得总任务数与总会员数的比值减小，出现供小于求效应，即对新的矩阵而言，它周围的任务数减少，任务竞争性增强，任务定价小于原来定价，并且期望任务完成率增加，证明此是否打包与第二问的目标函数有正相关性，且效果显著。

利用会员任务限额是否小于 4 进行分类是比较理想的一种分类，其原因在于对任务限额大于等于 4 的会员 σ_j 而言，对第 i 个任务 ω_i ，如果 ω_i 的周围任务数量 $\delta_i \geq 3$ ，表明在以 ω_i 坐标为坐标中心以 R 为半径建立的圆（任务 ω_i 的辐射范围）而言，周围任务 ω_{it} ， $1 \leq t \leq \alpha_i$ ，如果 $b_{ij} = 1$ ，那么会员 σ_j 在任务 ω_i 辐射范围内（由关系矩阵定义）， σ_j 与周围任务 ω_{it} 的距离必然小于 $2R$ ，由于会员 σ_j 能够选择的任务辐射范围半径 $R_j = \sqrt{\alpha_j}R \geq 2R$ ，（ $\alpha_j \geq 4$ ，由假设）。

所以 ω_i 的周围任务 ω_{it} 均在会员 σ_j 的任务辐射范围内，那么 $b_{it,j} = 1$ 对 $1 \leq t \leq \alpha_i$ 均成立，即打包的四个任务同时与该会员相关或同时与该会员无关，对算法约束条件的确立具有显著优越性。

由于周围任务数量的中位数为 2 且会员预定限额中位数为 3，进一步我们出于对保证大多数会员权益（即任务完成相对较好的会员可以选择收益更大的任务并且相对竞争较小）考虑任务限额是否小于 3 进行打包，与我们上述打包模型结果比较，只需对上述算法增加一个对每一个打包任务的所有子任务是否在由优先度排序对应的会员的任务辐射范围内进行判断的语句即可。

6.4 结果分析

一. 在此我们仅对四个任务打包进行了运算，从得到结果来看，打包策略是有效的，但同时我们意识到，这仅仅是众多打包策略中最简单的一种。对于平台而言，如果考虑更多种类的打包方案会不会让收益更大？还是说进一步加大了对低配额会员的限制，让平台对于新会员的吸引力减弱，从而不利于外包社区长期运营，对平台长期收益有较大损害。对于打包策略多元化的问题，仍值得继续深入研究、精进；

二. 打包任务能否不拘泥于地理位置的相邻性。如果充分考虑会员的行动路径，可以将会员常用路径上的任务打包，这样的联合发布方式极大的方便了会员，提升了任务的吸引力，同时也让会员产生沉浸式体验，沉浸体验会让会员对平台有较高的忠诚度，更容易持续参与，因此为平台带来持续客观的效益；

三. 类比于商场将促销商品时将优劣商品合并销售的策略，平台可以将鲜有人光顾的任务与利润高、竞争激烈的“抢手”任务联合发布，既能改善总体任务的执行度，也可以缓解会员间对任务的竞争程度，让会员有更好的满意度。

七. 问题四建模

7.1 思路分析与假设

对于问题四，基于附件三的新数据，给出自己的定价方案，并预估它可能的实施效果。先根据问题三的结论，对部分任务做联合发布处理，再依靠问题二中给出的报价策略模型，通过添加历史任务报价数据改进会员期望价格函数，该函数仅与新任务点附近历史任务报价、距离有关；然后根据第二问中求得的最优参数，给出新的报价函数，得到每一份任务的定价；最后估算该报价模型下可能得到的市场反馈，并与原始任务定价做对比。

假设：对于新任务，会员知道与其相邻的历史任务的报价，并以此估算自己的期望报价；

7.2 模型建立

对于新任务的报价思路：同于问题二，是基于对历史数据的“学习情况”、平台与会员的价格博弈给出新任务的平台报价策略模型。

首先根据第二问的模型，找出新任务 i 出现点周围的会员，可以将他们视作该任务的潜在客户；再对于每一个会员 j ，再次利用第二问提出的期望价格模型，会员 j 对自己周围历史任务的价格进行计算分析，同时考虑到自己与新任务点 i 的距离，得到自己可以接受的理想期望报价 $d_{i,j}$ ：

$$d_{i,j} = \rho_1 * \max\{v_k\} + (1 - \rho_1) * \frac{\sum v_k}{k}$$

其中 v_k 表示与会员 j 邻近的历史任务报价；最后外包公司再估计每一个新任务 i 的所有潜在客户给出的期望报价 $d_{i,j}$ 后，再通过公式：

$$c_i = \rho_2 * \max\{d_{i,j}\} + (1 - \rho_2) * \frac{\sum d_{i,j}}{k}, k \text{ 指对新任务的所有期望报价数}$$

其中 $\rho_2 \in [0,1]$ 表示定价系数，但若新任务周围并没有会员，也就是说产生了一个“孤僻”的任务点，我们让这里的报价 $c_i =$ 历史任务报价均值，这样便得到外包公司对于新任务的报价策略。

7.3 模型结果及评价

参考模型二给出的评价指标，同样的我们也选择“完成率”、“平台总支出”、“任务平均报价”、“满意度”对我们的价格策略做出评价，并与之前的历史数据做比较。但很明显可以看出相比于问题二，这里的任务基数更大，用平台总支出显得不是那么合适，于是引入有效交易下的平台平均支出这一概念，其计算方式：

$$\text{有效支出} = \frac{\text{平台为已执行任务的支出总和}}{\text{已执行的任务数量}}$$

根据问题二的结论，我们得到的价格模型中取参数 $\begin{cases} \text{会员满意度 } \rho_1=0.2 \\ \text{定价系数 } \rho_2=0.5 \end{cases}$ ，并得

到如下表格：

表 4：新任务指标与历史任务指标对比

	完成率	平台总支出	有效支出	总任务数
新任务	87.29%	143890	69.6477	2066
历史任务	38.90%	36950	69.8199	835

7.4 评价与改进

从市场检验上来看，该模型能够较好的提高任务完成率，并且能够把平台对任务的有效支出降低。由上可知，在新的报价策略模型下计算得到的报价相比于历史任务，完成率有着显著提高，完成任务的有效报价也稍稍低于历史任务，意味着新报价策略可以付出更少的资金，吸引会员完成更多的任务，这样就大大提高了公司的效益。同时我们看到这里参数

$\begin{cases} \text{会员满意度 } \rho_1=0.2 \\ \text{定价系数 } \rho_2=0.5 \end{cases}$ ，平台报价让会员有不错的满意度，这能够让外包社区更加长时间的运营，对于公司的长期利益有着极大的帮助。但同时该方案仍然存在着诸多不足之处有待改进。

一. 该问题中的打包策略不是最优的，如果能够再精进联合发布方式的话应该能够对完成率以及有效支出有较好的改进；

二. 这里选取的参数是由第二问提供, 虽然可以确保这是一个较优的结果, 但并不能保证这一定是最优的参数, 也就是说第二问的最优参数不一定完全适合于这里的新问题, 最后求得的市场检验指标也有待进一步的提高。

八. 模型的评价

本文主要有两个模型, 一是会员对任务的期望价模型, 二是基于会员优先度的打包模型。对于模型一, 我们的优点在于对模型引入了服从标准正态分布指数的随机变量, 此随机变量反映了受其他客观因素影响使会员对期望价取值的改变值。比如, 受偶然极端天气影响, 会员的任务期望价增大; 某一个任务距离该会员很远, 用不加随机变量的原模型那么期望价是比较高的, 如果该会员是正好要去该任务地点附近, 那么相应期望价可能会降低。引入此随机变量的意义就在于消除偶然误差的对期望价变化的影响, 更加符合实际情况。

我们的缺点在于没有考虑会员的生活圈, 对会员我们只有一个坐标取值考虑其任务辐射范围, 但会员会在工作地点与居住点之间大范围活动, 表明会员任务辐射范围会大于我们的模型值, 对模型建立周围的任务点会有影响, 从而影响模型的实用性。

对于模型二, 我们的优点在于打包之后利用会员配额限制, 降低高配额会员间的竞争, 同时降低平台定价成本; 同时低配额的会员只能选择其他任务, 提升任务整体的完成率。我们的缺点就在于对打包任务数量取值限定为一个常值, 实际情况中, 可能把相近地点的多个任务组合为多个打包方案, 不同方案打包任务数量不同, 使得任务定价方付出更少的定价总额; 也可能把会员不太愿意完成的任务即期望下不会被完成的任务与热点任务进行打包, 使得任务完成率提高。这是此模型亟待改进之处。

九. 结语

随着信息技术的快速发展, 越来越多的企业开始运用众包任务分配的交易方式, 它对企业电子采购和供应链管理产生深远的影响。本文通过数学方法分析了影响众包平台定价的影响因素, 结合市场规律, 引入经济学原理解释了数学模型计算得到的结果, 使之符合现实生活中的经济学现象, 对企业的运营具有一定的参考价值。对改进后的模型进行了市场反馈检验, 验证了会员优先度打包模型对于增加企业收益率有比较好的效果。

参考文献

- [1] 李荣喜. 2006. 基于价格参考效应的消费者需求和产品定价模型[J]. 管理评论, (11): 49-51.
- [2] 李荣喜. 2008. 基于损失避免的消费者需求和产品定价模型[J]. 决策参考, 5: 52. -54.
- [3] 徐雅卿, 魏轶华. 2008. 胡奇英. 基于 Priceline 的买方/卖方定价收益管理问题[J]. 管理科学学报, 11(3): 63-69.

附录:

利用 MATLAB 编程 Excel 做数据处理 MapGIS

附录一：第一题程序

```
fid = fopen('已完成的任务公里.txt','r');
aa = textscan(fid,'%s');
fclose(fid);
fid = fopen('会员点位公里数.txt','r');
bb = textscan(fid,'%s');
fclose(fid);
% 通过这几行读取 TXT 文件

aa = reshape([aa{1,1}], [8,835]);
bb = reshape([bb{1,1}], [8,1877]);
%将 cell 数据的引号去掉

task_price = xlsread('附件一：已结束项目任务数据.xls','D2:D836'); %任务价格
task_fin = xlsread('附件一：已结束项目任务数据.xls','E2:E836'); %任务完成情况
peie=xlsread('附件二：会员信息数据.xlsx','D2:D1878'); %会员配额
%读取 excel 文件

r = 2.5;%半径
R = 2.5;
m = 835;%任务数量
n = 1877;%会员数量
task_x = zeros(m,1);
task_y = zeros(m,1);
VIP_x = zeros(n,1);
VIP_y = zeros(n,1);

%%以浮点数形式记录位置坐标
for i = 1:m
    task_x(i) = str2num(aa{6,i});
    task_y(i) = str2num(aa{8,i});
end
for i = 1:n
    VIP_x(i) = str2num(bb{6,i});
    VIP_y(i) = str2num(bb{8,i});
end

%%构建关系矩阵 A 和 B
```


A = zeros(m); %任务之间的关系矩阵 行表示某一任务 后面跟着的向量记录与他相邻的任务编号

B = zeros(m,n);%构建 (m, n) 的矩阵 行表示某一任务 后面跟着的向量表示与他相邻的会员编号

```
tic;
for i = 1:m
    for j = 1:m
        if i ~= j
            if abs(task_x(i) - task_x(j)) < r && abs(task_y(i) - task_y(j)) < r
                %当两个任务相距小于 r 平方公里范围时 考虑为附近情况
                A(i,j) = 1;%表示任务 i 与任务 j 是有联系的
            end
        end
    end
end
end
toc;
```

```
tic;
for i = 1:m
    for j = 1:n
        if abs(task_x(i) - VIP_x(j)) < R && abs(task_y(i) - VIP_y(j)) < R
            %当任务与会员相距小于 R 平方公里范围时 考虑为附近情况
            B(i,j) = 1;%表示任务 i 与任务 j 是有联系的
        end
    end
end
end
toc;
```

```
nearTaskprice = zeros(m,1); %附近任务价格均值
nearTask_num = zeros(m,1); %附近任务数量
nearFTask_num = task_fin; %附近完成任务量的比例
nearVIP = zeros(m,1); %附近会员数量
nearVIP_Ca = zeros(m,1); %附近会员配额值的和（衡量购买力能力）
```

%%计算附近任务总价 任务完成量

```
tic;
for i = 1:m
    for j = 1:m
        if A(i,j) ~= 0 && i ~= j
            nearTaskprice(i) = nearTaskprice(i) + task_price(j,1);
            nearFTask_num(i) = nearFTask_num(i) + task_fin(j,1);
        end
    end
end
```

```

        end
    end
    toc;

    %%计算附近任务数 附近 VIP 数量 修正完成率
    for i = 1:m
        nearTask_num(i) = sum(A(i,:));
        nearVIP(i) = sum(B(i,:));
        if nearTask_num(i) ~= 0;
            nearTaskprice(i) = nearTaskprice(i) / nearTask_num(i);
            nearFTask_num(i) = nearFTask_num(i) / (nearTask_num(i) + 1);
        else
            nearTaskprice(i) = 0;
        end
    end
end

%%计算任务完成情况
tic;
for i = 1:m
    for j = 1:n
        if B(i,j) ~= 0
            nearVIP_Ca(i) = nearVIP_Ca(i) + peie(j,1);
        end
    end
end
end
toc;

```

第二题程序：

```

fid = fopen('已完成的任务公里.txt','r');
aa = textscan(fid,'%s');
fclose(fid);
fid = fopen('会员点位公里数.txt','r');
bb = textscan(fid,'%s');
fclose(fid);
% 通过这几行读取 TXT 文件

aa = reshape([aa{1,1}], [8,835]);
bb = reshape([bb{1,1}], [8,1877]);
%将 cell 数据的引号去掉

task_price = xlsread('附件一： 已结束项目任务数据.xls','D2:D836'); %任务价格
task_fin = xlsread('附件一： 已结束项目任务数据.xls','E2:E836'); %任务完成情况
peie=xlsread('附件二： 会员信息数据.xlsx','D2:D1878'); %会员配额
%读取 excel 文件

```

```

r = 2.5;%半径
m = 835;%任务数量
n = 1877;%会员数量
task_x = zeros(m,1);
task_y = zeros(m,1);
VIP_x = zeros(n,1);
VIP_y = zeros(n,1);

%%以浮点数形式记录位置坐标
for i = 1:m
    task_x(i) = str2num(aa{6,i});
    task_y(i) = str2num(aa{8,i});
end
for i = 1:n
    VIP_x(i) = str2num(bb{6,i});
    VIP_y(i) = str2num(bb{8,i});
end

%%构建关系矩阵 A 和 B
A = zeros(m); %任务之间的关系矩阵 行表示某一任务 后面跟着的向量记录与他相邻的任务编号
B = zeros(m,n);%构建 (m, n) 的矩阵 行表示某一任务 后面跟着的向量表示与他相邻的会员编号
C = zeros(n,m);%构建 (n, m) 的矩阵 行表示会员 后面跟着的向量表示与它邻近的任务编号

for i = 1:m
    for j = 1:m
        if i ~= j
            if abs(task_x(i) - task_x(j)) < r && abs(task_y(i) - task_y(j)) < r
                %当两个任务相距小于 r 平方公里范围时 考虑为附近情况
                A(i,j) = 1;%表示任务 i 与任务 j 是有联系的
            end
        end
    end
end

for i = 1:m
    for j = 1:n
        if abs(task_x(i) - VIP_x(j)) < r && abs(task_y(i) - VIP_y(j)) < r
            %当任务与会员相距小于 R 平方公里范围时 考虑为附近情况
            B(i,j) = 1;%表示任务 i 与任务 j 是有联系的
        end
    end
end

```

```

        end
    end

    for i = 1:n
        for j = 1:m
            if abs(VIP_x(i) - task_x(j))^2 + abs(VIP_y(i) - task_y(j))^2 < r^2 * peie(i)
                %不同会员由于配额差异 辐射面也不一样
                C(i,j) = 1;
            end
        end
    end
end

```

%%对于每一个任务求出它所辐射范围内用户给出的期望价格

```

d = zeros(m,n);
p = 0.1; %范围是（0 ， 1） p 越大满意度越高
for i = 1:m

```

```

    for j = 1:n
        %表示第 i 个任务 第 j 个会员
        if B(i,j) == 1
            num_v = sum(C(j,:));%该会员辐射范围内任务数
            v = zeros(num_v,1);
            k = 1;
            for ii = 1:m
                if C(j,ii) == 1
                    %找出与会员 j 有关联的任务 ii
                    v(k) = task_price(ii);
                    k = k + 1;
                end
            end
            %会员能够给出的期望价格
            if num_v ~= 0
                d(i,j) = p*max(v) + (1 - p)*sum(v)/num_v;
            else
                d(i,j) = 0;
            end
        end
    end
end
end

```

```

t = 1;
target = zeros(11,1);
e = zeros(11,1);
cost = zeros(11,1);
tic;

```

```

for q = 0:0.1:1
    %%通过前面的期望报价矩阵 d 求出平台想给出的价格 c 以及完成情况矩阵 ch
    ch = zeros(m,n);
    c = zeros(m,1);
    for i = 1:m
        [ii,jj] = find(d(i,:)'~=0);
        %找出期望价格矩阵中 对第 i 个任务有报价的会员编号 jj
        [row,col] = size(jj);
        %求出 jj 的数量 row 当 col 为 0 时表示没有人报价
        if col > 0
            u = zeros(col,1);
            for k = 1:col
                u(k) = d(i,jj(k));
            end
            c(i) = q*max(u) + (1-q)*sum(u)/col;
        else
            c(i) = 0;
        end
    end
end

for i = 1:m
    for j = 1:n
        if d(i,j) == 0
            ch(i,j) = 0;
        else
            ch(i,j) = choose(d(i,j),c(i));
        end
    end
end
sum(sum(ch))

cost(t) = 0;%总的支出
for i = 1:m
    if sum(ch(i,:)) > 0
        %表示有人对他报价
        cost(t) = cost(t) + c(i);
    end
end

%%计算完成率
e(t) = 0;
for i = 1:m
    [ii,jj] = find(d(i,:)'~=0);
    [row,col] = size(jj);

```

```

        if col > 0
            e(t) = e(t) + sum(ch(i,:))/col;
        end
    end
    e(t) = e(t) / m;
    target(t) = e(t) / cost(t);
    t = t + 1;
end
toc;

% t = 0:0.01:1;
% plot(t,target)
% title('定价参数和优化目标的关系')
% xlabel('定价参数')
% ylabel('优化目标')

```

第三题程序：

```

fid = fopen('已完成任务公里.txt','r');
aa = textscan(fid,'%s');
fclose(fid);
fid = fopen('会员点位公里数.txt','r');
bb = textscan(fid,'%s');
fclose(fid);
% 通过这几行读取 TXT 文件

aa = reshape([aa{1,1}], [8,835]);
bb = reshape([bb{1,1}], [8,1877]);
%将 cell 数据的引号去掉

task_price = xlsread('附件一：已结束项目任务数据.xls','D2:D836'); %任务价格
task_fin = xlsread('附件一：已结束项目任务数据.xls','E2:E836'); %任务完成情况
peie=xlsread('附件二：会员信息数据.xlsx','D2:D1878'); %会员配额
%读取 excel 文件

r = 2.5;%半径
m = 835;%任务数量
n = 1877;%会员数量
task_x = zeros(m,1);
task_y = zeros(m,1);
VIP_x = zeros(n,1);
VIP_y = zeros(n,1);

%%以浮点数形式记录位置坐标
for i = 1:m

```

```

        task_x(i) = str2num(aa{6,i});
        task_y(i) = str2num(aa{8,i});
    end
    for i = 1:n
        VIP_x(i) = str2num(bb{6,i});
        VIP_y(i) = str2num(bb{8,i});
    end

    %%构建关系矩阵 A 和 B
    A = zeros(m); %任务之间的关系矩阵 行表示某一任务 后面跟着的向量记录与他相邻的
    任务编号
    B = zeros(m,n);%构建 (m, n) 的矩阵 行表示某一任务 后面跟着的向量表示与他相邻
    的会员编号
    C = zeros(n,m);%构建 (n, m) 的矩阵 行表示会员 后面跟着的向量表示与它邻近的任
    务编号

    for i = 1:m
        for j = 1:m
            if i ~= j
                if abs(task_x(i) - task_x(j)) < r && abs(task_y(i) - task_y(j)) < r
                    %当两个任务相距小于 r 平方公里范围时 考虑为附近情况
                    A(i,j) = 1;%表示任务 i 与任务 j 是有联系的
                end
            end
        end
    end

    for i = 1:m
        for j = 1:n
            if abs(task_x(i) - VIP_x(j)) < r && abs(task_y(i) - VIP_y(j)) < r
                %当任务与会员相距小于 R 平方公里范围时 考虑为附近情况
                B(i,j) = 1;%表示任务 i 与任务 j 是有联系的
            end
        end
    end

    for i = 1:n
        for j = 1:m
            if abs(VIP_x(i) - task_x(j))^2 + abs(VIP_y(i) - task_y(j))^2 < r^2 * peie(i)
                %不同会员由于配额差异 辐射面也不一样
                C(i,j) = 1;
            end
        end
    end
end

```

```

%%对于每一个任务求出它所辐射范围内用户给出的期望价格
d = zeros(m,n);
p = 0; %范围是（0 ， 1） p 越大满意度越高
for i = 1:m
    for j = 1:n
        %表示第 i 个任务 第 j 个会员
        if B(i,j) == 1
            num_v = sum(C(j,:)); %该会员辐射范围内任务数
            v = zeros(num_v,1);
            k = 1;
            for ii = 1:m
                if C(j,ii) == 1
                    %找出与会员 j 有关联的任务 ii
                    v(k) = task_price(ii);
                    k = k + 1;
                end
            end
            %会员能够给出的期望价格
            if num_v ~= 0
                d(i,j) = p*max(v) + (1 - p)*sum(v)/num_v;
            else
                d(i,j) = 0;
            end
        end
    end
end

t = 1;
target = zeros(11,1);
e = zeros(11,1);
cost = zeros(11,1);
tic;
for q = 1:1
    %%通过前面的期望报价矩阵 d 求出平台想给出的价格 c 以及完成情况矩阵 ch
    ch = zeros(m,n);
    c = task_price;

    for i = 1:m
        for j = 1:n
            if d(i,j) == 0
                ch(i,j) = 0;
            else
                ch(i,j) = choose(d(i,j),c(i));
            end
        end
    end
end

```



```

        end
    end
end

cost(t) = 0;%总的支出
for i = 1:m
    if sum(ch(i,:)) > 0
        %表示有人对他报价
        cost(t) = cost(t) + c(i);
    end
end

%%计算完成率
e(t) = 0;
for i = 1:m
    [ii,jj] = find(d(i,:)'~=0);
    [row,col] = size(jj);
    if col > 0
        e(t) = e(t) + sum(ch(i,:))/col;
    end
end
e(t) = e(t) / m;
target(t) = e(t) / cost(t);
t = t + 1;
end
toc;

```

第四题程序：

```

fid = fopen('已完成的任务公里.txt','r');
aa = textscan(fid,'%s');
fclose(fid);
fid = fopen('会员点位公里数.txt','r');
bb = textscan(fid,'%s');
fclose(fid);
fid = fopen('新任务点位公里数.txt','r');
cc = textscan(fid,'%s');
fclose(fid);
% 通过这几行读取 TXT 文件

aa = reshape([aa{1,1}], [8,835]);
bb = reshape([bb{1,1}], [8,1877]);
cc = reshape([cc{1,1}], [8,2066]);
%将 cell 数据的引号去掉

```

```

task_price = xlsread('附件一：已结束项目任务数据.xls','D2:D836'); %任务价格
task_fin = xlsread('附件一：已结束项目任务数据.xls','E2:E836'); %任务完成情况
peie=xlsread('附件二：会员信息数据.xlsx','D2:D1878'); %会员配额
%读取 excel 文件

r = 2.5;%半径
m = 835;%任务数量
n = 1877;%会员数量
L = 2066;%新的任务数量
task_x = zeros(m,1);
task_y = zeros(m,1);
VIP_x = zeros(n,1);
VIP_y = zeros(n,1);
newtask_x = zeros(L,1);
newtask_y = zeros(L,1);

%%以浮点数形式记录位置坐标
for i = 1:m
    task_x(i) = str2num(aa{6,i});
    task_y(i) = str2num(aa{8,i});
end
for i = 1:n
    VIP_x(i) = str2num(bb{6,i});
    VIP_y(i) = str2num(bb{8,i});
end
for i = 1:L
    newtask_x(i) = str2num(cc{6,i});
    newtask_y(i) = str2num(cc{8,i});
end

%%构建关系矩阵 A 和 B
A = zeros(m); %任务之间的关系矩阵 行表示某一任务 后面跟着的向量记录与他相邻的
任务编号
B = zeros(m,n);%构建 (m, n) 的矩阵 行表示某一任务 后面跟着的向量表示与他相邻
的会员编号
C = zeros(n,m);%构建 (n, m) 的矩阵 行表示会员 后面跟着的向量表示与它邻近的任
务编号
D = zeros(L,n);%构建 (L, n) 的矩阵 行表示新任务 后面跟着的向量表示与它邻近的会
员编号
E = zeros(L,L);%构建 (L, L) 的矩阵 行表示新的任务 纵表示与它相邻新任务编号

for i = 1:m
    for j = 1:m
        if i ~= j

```

```

        if abs(task_x(i) - task_x(j)) < r && abs(task_y(i) - task_y(j)) < r
            %当两个任务相距小于 r 平方公里范围时 考虑为附近情况
            A(i,j) = 1;%表示任务 i 与任务 j 是有联系的
        end
    end
end

for i = 1:m
    for j = 1:n
        if abs(task_x(i) - VIP_x(j)) < r && abs(task_y(i) - VIP_y(j)) < r
            %当任务与会员相距小于 R 平方公里范围时 考虑为附近情况
            B(i,j) = 1;%表示任务 i 与任务 j 是有联系的
        end
    end
end

for i = 1:n
    for j = 1:m
        if abs(VIP_x(i) - task_x(j))^2 + abs(VIP_y(i) - task_y(j))^2 < r^2 * peie(i)
            %不同会员由于配额差异 辐射面也不一样
            C(i,j) = 1;
        end
    end
end

for i = 1:L
    for j = 1:n
        if abs(newtask_x(i) - VIP_x(j)) < r && abs(newtask_y(i) - VIP_y(j)) < r
            %当新任务与会员小于一定范围时 考虑为附近情况
            D(i,j) = 1;
        end
    end
end

for i = 1:L
    for j = 1:L
        if i ~= j
            if abs(newtask_x(i) - newtask_x(j)) < r && abs(newtask_y(i) - newtask_y(j)) <
r
                %当两个任务相距小于 r 平方公里范围时 考虑为附近情况
                E(i,j) = 1;%表示任务 i 与任务 j 是有联系的
            end
        end
    end
end

```

```

        end
    end

    %%对于每一个任务求出它所辐射范围内用户给出的期望价格
    d = zeros(L,n);
    p = 0.2; %范围是（0 ， 1） p 越大满意度越高
    for i = 1:L
        for j = 1:n
            %表示第 j 个会员对第 i 个新任务的报价
            if D(i,j) == 1
                num_v = sum(C(j,:));%该会员辐射范围内历史任务数
                v = zeros(num_v,1);
                k = 1;
                for ii = 1:m
                    if C(j,ii) == 1
                        %找出与会员 j 有关联的任务 ii
                        v(k) = task_price(ii);
                        k = k + 1;
                    end
                end
                %参考历史价值，该会员能够给出的期望价格
                if num_v ~= 0
                    d(i,j) = p*max(v) + (1 - p)*sum(v)/num_v;
                else
                    d(i,j) = 0;
                end
            end
        end
    end

    %%通过前面的期望报价矩阵 d 求出平台想给出的价格 c 以及完成情况矩阵 ch
    q = 0.5;
    ch = zeros(L,n);
    c = zeros(L,1);
    for i = 1:L
        [ii,jj] = find(d(i,:)==0);
        %找出期望价格矩阵中 对第 i 个任务有报价的会员编号 jj
        [row,col] = size(jj);
        %求出 jj 的数量 row 当 col 为 0 时表示没有人报价
        if col > 0
            u = zeros(col,1);
            for k = 1:col
                u(k) = d(i,jj(k));
            end
        end
    end

```

```

        c(i) = q*max(u) + (1-q)*sum(u)/col;
    else
        c(i) = 0;
    end
end
end

```

```

for i = 1:L
    for j = 1:n
        if d(i,j) == 0
            ch(i,j) = 0;
        else
            ch(i,j) = choose(d(i,j),c(i));
        end
    end
end
end

```

```

cost = 0;%总的支出
for i = 1:L
    if sum(ch(i,:)) > 0
        %表示有人对他报价
        cost = cost + c(i);
    end
end
end

```

```

%%计算完成率
e = 0;
for i = 1:L
    [ii,jj] = find(d(i,:)==0);
    [row,col] = size(jj);
    if col > 0
        e = e + sum(ch(i,:))/col;
    end
end
e = e / L;

```

```

%%计算饱和系数
baohe = zeros(L,1);
for i = 1:L
    for j = 1:n
        if ch(i,j) ~= 0
            baohe(i) = baohe(i) + peie(j);
        end
    end
end
if baohe(i) ~= 0 && sum(E(i,:)) ~= 0

```

```
        baohe(i) = baohe(i) / sum(E(i,:));  
    else  
        baohe(i) = 0;  
    end  
end  
target = e / cost;
```