基于博弈论的工程招标决策

摘 要

本研究以博弈论为基础，深入探讨了工程招标中的决策问题。首先，本文针对四个核心问题进行了分析，包括无额外信息情况下的最优报价策略、部分信息情况下的报价优化、建模优化决策下的策略选择，以及评标机制的有效性分析。接着，我们构建了相应的博弈模型，运用了多种博弈理论和数值模拟等方法，在完全信息、不完全信息以及存在串标等不同情况下，探讨了最优策略。最后，在评标机制分析中，我们通过引入新的因素和调整公式，提高了评标机制的稳定性和有效性，旨在为实际应用提供了有力的理论支持。

关键字： 建模决策　工程招标　合成标底　博弈论　评标机制　数值模拟

一、 问题重述

在 problem B 中，我们就“在工程招标中应如何依据博弈论决策”展开讨论。

1.1 约束条件

- 评标基准价计算方案(即三元组 (A, B1, B2))在 11 中方案中的选择完全随机；
- 所有投标人的投标评审价必须在 4649516 元至 4907459 元范围内；
- 投标人的数量（包括我方）在两个至十个之间（包括两个和十个）;
- 采用差价扣分制度，且扣分值与投标评审价与投标基准价的差距成正比，投标评审价大于投标基准价时的比例系数是投标评审价小于投标基准价时的两倍。

1.2 待解决问题

可以将问题简述如下：

1. 在所有投标人的评审价未知的情况下 ,我方应采取什么策略以使扣分最低的可能性最大；
2. 若已知部分投标人的评审价大致范围，如何优化我方的决策；
3. 若了解部分投标人也在通过建模优化决策，我方应采取什么策略；
4. 分析现有评标机制的有效性，并提出改进方案。

二、 模型假设

我们假定我们所讨论的问题满足以下模型条件：

- 对于完全不清楚的投标人的投标评审价，我们认为其在区间 4649516 元至 4907459 元范围内随机分布；
- 在问题 2 中，在得知部分投标人的投标标评审价的大致范围后，这些投标人的投标评审价在该区间内随机分布；
- 在问题 1、2、3 中，我们不考虑投标人之间的串标行为；
- 由于招标人的决策已经确定且公开，所以我们忽略这部分博弈；
- 题设中招标人为可靠稳定的政府机关，因此我们认为可以忽略临时因素（如招标人自身的因素）对招标结果的影响。

三、 符号说明

符号	意义
N	参与竞标人数
i	招标人编号（后简称投标人 i），我方为 1 号，其余招标人依次为 i = 2、3……N-1、N
j	评标基准价权重三元组对应方案编号（后简称方案 j），依次为 j= 1、2……10、11
A_j	方案 j 中招标人意向投标评审价（元）
B_j	方案 j 中 B1、B2 的取值（考虑到在本题环境中 B1、B2 始终保持相同）
U_i	投标人 i 的投标评审价
$\{V_N\}$	将$\{U_i\|\mathrm{i}=1,2,\ldots\ldots,\mathrm{N}\}$进行不降序排序后形成的数列，即有$V_i \leq V_{\mathrm{i+1}}$
W_j	在方案 j 获得的评标基准价
S_{ij}	投标人 i 在方案 j 下扣分情况；假定 $U_i \leq W_j$时，$S_{\mathrm{ij}} = W_j - U_i$； $U_i > W_j$时，$S_{\mathrm{ij}} = 2 * (U_i - W_j)$
$f_j(\{U_N\})$	布尔函数，表示在第 j 种参数下我方扣分是否最少/中标，即$S_{1\mathrm{j}}$是否是$\{S_{\mathrm{ij}}\}$中最小的
$F(\{U_N\})$	在所有方案下我方成功中标次数即$F(\{U_N\}) = \sum_{\mathrm{j}=1}^{11} f_j(\{U_N\})$
$[L_i, R_i]$	投标人 i 的报价区间，在该报价区间内投标人选择某一报价是等可能的
$G(\mathrm{U\_1})$	已知U_1、$U_2 - U_N$在给定区间内随机均匀取值、方案的选取完全随机时，我方的获胜权重
$H(U_1)$	问题 3 中多次迭代结果叠加值

四、 问题分析与模型建立

4.1 综合概述

投标中，所有投标人共享 100% 的成功率，一名投标人成功率的上升会带来其他投标人成功率的下降，竞争无法提高总成功率，所以这属于**零和博弈**。

投标还是典型的**静态博弈**：虽有参与者同时作出行动，没有先后次序，在本质上是平等的。

由于开标前无法知道其他投标人的报价，所以这还属于**不完全信息报价**。

在问题 1、2、3 中，我们认为投标人之间没有串标，投标人之间无法合作，所以这属于**非合作博弈**。在非合作博弈中，个人的决策是决定结果的主要因素。

在问题 4 中，有串标的情况出现，故无法再认为其是非合作博弈，因为串标者之间存在合作博弈。

4.2 判断中标

4.2.1 步骤一：获取评标基准价

在方案 j 获得的评标基准价，在本题中由于 B1=B2，有

$$W_j = B_j * \frac{U_1 + U_2 + \ldots + U_N}{N} + B_j * V_2 + \left(1 - 2 * B_j\right) * A_j$$

4.2.2 步骤二：计算每位投标人扣分情况

$U_i \le W_j$时，$S_{\mathrm{ij}} = W_j - U_i$；

$U_i > W_j$时，$S_{\mathrm{ij}} = 2 * \left(U_i - W_j\right)$

4.2.3 步骤三：比较我方与其他投标人的扣分情况

若$S_{1\mathrm{j}}$是所有$S_{\mathrm{ij}}(i = 2, 3, \ldots\ldots, N)$中最小的，则我方中标;

若$S_{1\mathrm{j}}$不是所有$S_{\mathrm{ij}}(i = 2, 3, \ldots\ldots, N)$中最小的，则我方未中标。

4.3 分别分析

4.3.1 问题 1

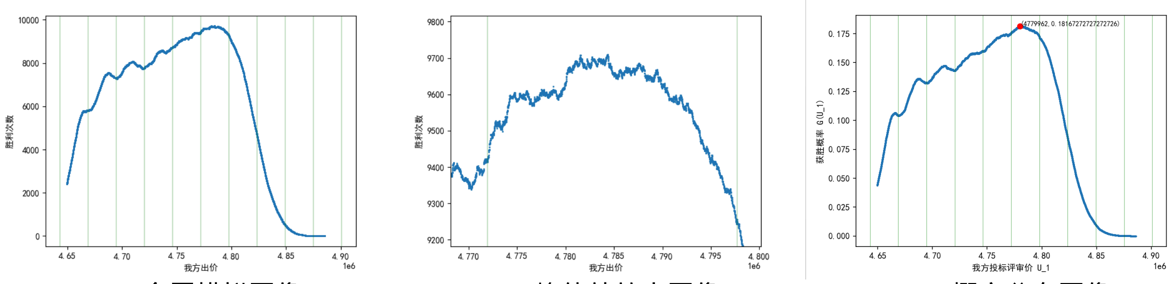
在第 1 问中，我们认为除我方外的投标者都是从默认投标评审价区间中随机选择一个数作为自己的投标评审价，其中选中每一个数的概率是相同的，属于**几何概型**。

我们需要知道，对于我方的任意报价，我们从扣分最小的可能性。我们对$F(\{U_N\})$在其他报价人的报价范围组合成的 N-1 维区域上积分，并计算

$$F(\{U_N\}) = \int_{L_2}^{R_2} \int_{L_3}^{R_3} \dots \int_{L_n}^{R_n} f_j(\{U_N\}) dU_N dU_{N-1} \dots dU_3 dU_2$$

其中，L_i和R_i分别是投标人 i 的报价区间下限和上限。由此我们便可以得到$G(U_1)$，其中$G(U_1)$与我方扣分最小的可能性成正比。**积分可以通过随机取样的方式去模拟**，通过**多次**对U_2到U_N随机取值我们进行了足够多次的模拟，从而得到$G(U_1)$的近似值，进而得到$G(U_1)$大致图像。得到$G(U_1)$的函数图像后，我们便可以找出$G(U_1)$的最大值$\max(G(U_1))$以及其对应U_1的值（记作U_1*）。

经过随机取样，我们模拟出了以下图像：（下面以 N=10 为例）



（a）全局模拟图像　（b）峰值处放大图像　（c）概率分布图像

图 1　N=10 时的模拟图像

其中（c）中图像为

我们观察到$G(U_1)$在大体趋势上存在一个极大值。在极大值左边的区间中$(U_1 < U_1*)$，$G(U_1)$总体呈现单调递增趋势（即$G'(U_1) > 0$）；在极大值右边的区间中$(U_1 > U_1*)$，$G(U_1)$总体呈现单调递减趋势（即$G'(U_1) < 0$）。

于是$G(U_1)$的最大值就是其极大值。U_1*在投标评审价默认区间的中部。最大值右方的曲线斜率比左方曲线的斜率更大，这是因为“扣分值与投标评审价与投标基准价的差距成正比，投标评审价大于投标基准价时的比例系数是投标评审价小于投标基准价时的两倍”。

当 N 变化时，$G(U_1)$图像也会变化。随着 N 的增大，出现极大值 U_1*对应的我方报价越高。

在 N 比较小时，$G(U_1)$为凸函数，$G''(U_1) < 0$；

在 N 比较大时，$G(U_1)$的右端有一段凹区间，$G''(U_1) > 0$。

在问题 1 中我方给出的**最优策略**就是：根据不同的 N，我方报价遍历所有可能的值，并模拟竞争相同且足够多次，（根据报价-中标次数）绘出$G(U_1)$的函数图像（a），然后计算出对应概率分布图像（c），以$G(U_1)$取最大值即$G(U_1*)$ 时U_1的值U_1* 作为我方的投标评审价；如图中所示，在 N=7 时，$U_1* = 4780096$ （单位：元）

4.3.2 问题 2

在问题 2 中，部分投标人（记为群体 C）的评审价报价区间发生了调整，体现为区间长度有所缩减。我们依旧认为剩余的投票人的每种报价依旧是区间[4649516, 4907459]（单位：元）上等概率的。于是我们可以沿用第 1 问的策略。

当我们模拟获得$G(U_1) - U_1$图像时，我们观察到，当其他投标者的报价区间相对集中时，会导致报价集中区域的函数值$G(U_1)$降低，同时该区域两端的函数值$G(U_1)$升高。

例如，若 C 的报价范围集中在原来$G(U_1)$的峰值U_1*附近，那么该峰值就会成为谷值，并在两端生成新的峰值。若是其他报价人报价范围集中在原来峰值的左侧，那么峰值就会往右偏移，即U_1*变大。同理,若 C 的报价范围集中在原来峰值的右侧，那么峰值就会往左偏移，即U_1*变小。

例如，当 N=10，其他 C 群体中有 6 人、报价都在 4700000 到 4800000 之间时，$G(U_1)$ 的图像如下：

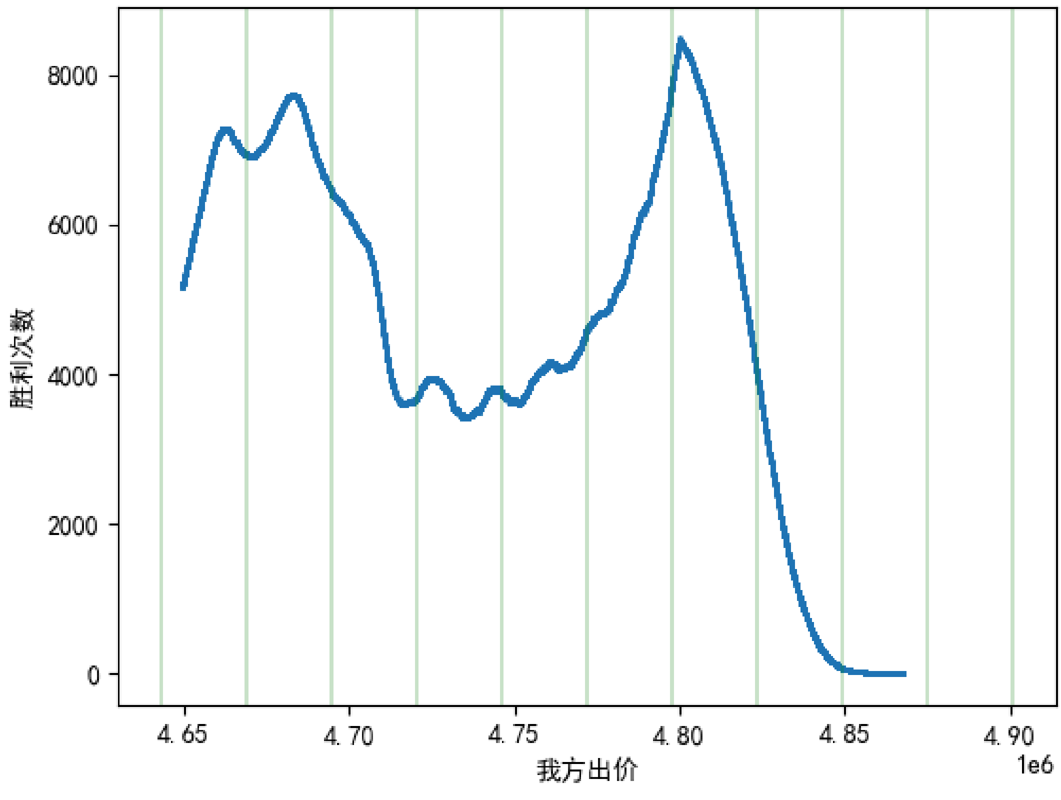


图 2　峰值分离模拟图像

而若当其他投标人报价范围不够集中时，那么这对原来峰值的影响是非常有限的。

这也印证了**我方应当权衡避免与其他投标人竞争与抢占原有峰值。**

以失火逃生中的博弈（典型的非合作博弈）打比方做解释：一间房子失火了，只有一个大门和一个小门能逃生。如果其他人都是随机选择出口进行逃生，那么显然我方应该选择逃生概率更大的大门。但是如果我们知道了其他人的选择会集中在大门，那么我

们就应该权衡门的大小和拥挤程度进行抉择。如果大门过于拥挤，那么选择小门将会有更高的逃生几率。

即使函数图像发生变化，在第 2 问中我方的最优策略依旧是选择$G(U_1)$取最大值时U_1的值U_1*作为投标评审价。

4.3.3 问题 3

在问题 3 中，存在部分投标人（记为这些人以及我方为群体 D）也在尝试通过数学建模的方法优化决策的情况。

我们认为，这些投标人和我方用的是同一个数学模型，此时只需要将问题 1 中模型获得的最优策略作为 D 和我方的报价，不断迭代，直到出现相对稳定的状态为止。

（为了避免平局都算成功带来的所有投标人的报价收敛于一点的消极情况，应该让每个投标人的报价在上下范围小幅度波动以防止平局。）

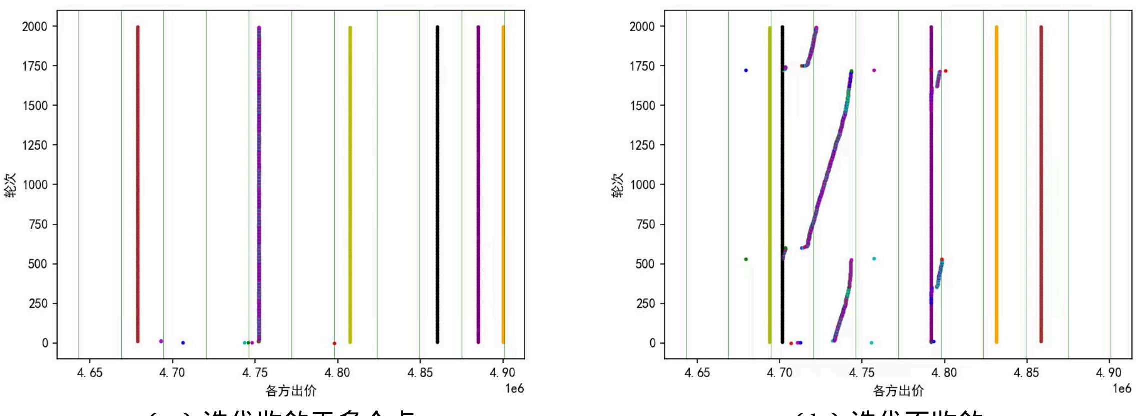
在“D 中人数较少”的情况下，最终 D 的报价会收敛于一点，该点就是原来$G(U_1)$的峰值$G(U_1*)$。这是因为此时竞争压力较小，选择峰值会带来最大的收益。此时**我方的最优策略就是选择该峰值的位置作为投标评审价。**

而当 D 中人数较多时，此时我们发现，投标人的最优决策收敛于不同的值，有时还会出现不收敛的情况。例如，两个或多个投标人报价的周期性渐变，收敛于多个值。

在问题 2 中，多峰值的出现本质上是相同的，因为此时避免竞争能带来更大的收益。而渐变则是 D 中投标人互相采取了对另外一名或几名投标人的针对性策略导致的。

例如，现在有甲乙两人参与一个猜数字的游戏。两个人分别选择一个 1-100 的整数（假定两个整数不能相同），之后随机生成一个处于 1-100 的整数，两个数中较靠近随机生成数的选择者获胜。假使甲一开始选了 30，那么对于乙，选择 31 而非 50 是最优策略；当乙选择 31 后，甲就会选择 32......直到乙选择 50，甲选择 51；此时任何一个人都不会做出改变了，因为任何改变都不会使得自己获胜的可能性增大

但并非所有所有迭代都能收敛于一点，有时会出现循环的情况。



（a）迭代收敛于多个点　　（b）迭代不收敛

图 3　迭代不收敛于一点

此时我们需要运用**纳什均衡**。

纳什均衡是指满足下面性质的策略组合：任何一位玩家在此策略组合下单方面改变自己的策略（其他玩家策略不变）都不会提高自身的收益。

纳什均衡是博弈中的稳定状态，在这种状态中，博弈者没有动力区改变自己的决策，因为改变决策不会带来收益，甚至带来损失。我们可以采取"设置未知数，求解方程"的方式获取最优策略；也可以预估均衡状态来设置初始状态，使得迭代更可能得到稳定状态；同时也可以通过设置不同的初值进行迭代，将结果进行叠加，从结果中寻找收敛点或者收敛范围，作为最优策略。

但是，考虑到采取数学建模方法的投标人（包括我方），本质上是相同的；所以我方的决策可能是任意一个收敛点。这时，我们需要运用混合策略。

所谓混合策略，就是以一定的概率随机选取策略。

打个比方，甲乙在无法沟通的情况下参与一个游戏，每人从 AB 中选择一者：假如两人选择一样，那么两人什么都得不到；假如两人选择不同，那么选择 A 的人可以获得 2 分，选择 B 的人可以获得 1 分。

在这个博弈中，假如两人选择了相同的单一对策，那么就会一起选 A，或者一起选 B，最后什么都得不到。如果最优策略是一种随机策略：假设两人选择 A 的概率分别是 p 和 q。当甲选定 p 的值后，乙再根据此 p 选定能让自己有最高的收益。甲要使得在乙作出选择后，自己的收益最高。反之同理。甲的收益期望是$2p(1-q)+(1-p)q$，乙的收益是$p(1-q)+2(1-p)q$。

代入计算，解得当 p 和 q 都等于 2/3 时，**达到了纳什均衡的状态**。若此 p 等于$\frac{2}{3}$，则乙的收益恒为 1。反之同理。时两个人分别改变策略，改变 p 或 q 的值，都无法带来收益。

在投标中，我方需要选定报价，使得其他人最大化自己的收益时，我方的收益达到最大的情况。所以投标与这个二选一的博弈具有相似性。我们注意到，$2:1=p:(1-p)$，所以作出一个选择的概率与没有竞争者时作出该选择的收益有明显的正相关性。所以我们可以通过$G(U_1)$来给混合策略中每种选择标定概率。

另外，我们通过将迭代的结果叠加，得到函数$H(U_1)$，代表迭代结果中该点作为评审价的出现频率。显然，$H(U_1)$越大，我方选择U_1作为评审价的概率也应该越大。

4.3.4 问题 4

4.3.4.1 串标行为分析

在问题 4 中，我们考虑串标（串通投标）的情况存在。

概念：串标是一种违法行为，它违反了《中华人民共和国刑法》第二百二十三条。串标一般分为两种，投标人与招标人之间串通，和投标人之间串通。因为我们认为招标人始终保持中立，所以我们这里只讨论后者。

一种常见的串标方法是集体抬高或压低报价。串标者可以约定共同抬高，压低，或轮流抬高和压低报价来改变基准价。我们注意到， 投标基准价的决定公式

$$W_j = B_j * \frac{U_1 + U_2 + \ldots + U_N}{\mathbb{N}} + B_j * V_2 + \left(1 - 2 * B_j\right) * A_j$$

中有一个因子是算术平均数$\frac{U_1+U_2+\ldots U_N}{N}$，只要串标者集体抬高或压低报价，基准价$W_j$就会相应升高或降低，这会导致$G(U_1)$峰值的位置$U_1*$偏移。投标基准价的决定公式里还有另一个因子，次低投标评审价V_2，它可以被集体压低报价所影响。此时，串标者只要根据这个偏移调整自己的决策，那么就能使自己更容易从竞标中胜出。

这种串标是相对容易被发现的，**如果出现集体的极端的报价，就可以认定为有串标的嫌疑。**

还有一种不那么明显的串标手段。根据第二问的结论，一方选中一个点作为报价，那么其他方再在该点附近区域报价的获胜概率就会下降，而相隔该点一定距离的函数图像上会出现峰值。只要串标者控制好报价的距离，使己方报价正好在对方影响下$G(U_1)$的峰值U_1*上，就能在避免竞争的同时互相增大对方获胜的概率。这种串标比较难被发现，因为在一次串标中即使不串标，也会出现投标人报价正好错开的情况。

我们可能需要用这些投标者的历史投标记录来找出这种串标：**如果多个投标人在多次投标中的投标评审价都正好错开合适的距离，那么他们就有串标的嫌疑。**

4.3.5 当前机制分析

当前机制能够在一定程度上抑制串标的行为：

1. 算术平均数能够使每一位投标人的报价都能影响结果，是基准价结果稳定在一定区域，难以被小团体撼动；
2. 次低价比最低价更能代表数据的特性，更不容易被极端出价影响；
3. 三元组参数由多组中随机选择一组，增加了基准价的随机性，更难被串标者预测；
4. 三元组中A_j的值本身不是由投标人决定的，也能防止基准价的极端化；
5. 投标报价具有趋异性；如果集体抬高或压低报价，也会在一定情况下加剧了串标者之间的竞争；如果是正常投标，报价应当尽量避开其他竞争者。

4.3.6 当前机制改进

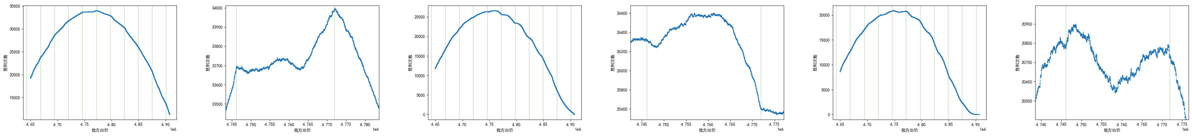
我们还可以改进这种招标方法来抑制串标行为：

- 将投标基准价的公式复杂化：在公式中，我们可以引入更多因子，如说中位数，标准差，次高价等等；
- 提高公式的随机性：如增加更多的三元组方案，使得基准价更难被预测，加大了串标的难度；
- 加大对极端报价的惩罚：如将线性扣分机制改为抛物线类扣分机制，防止基准价的极端化；
- 我们还可以在投标中引入其他环节，例如当前部分招标人采用的技术水平打分，使得投标者的评分的来源更多样化，同时抑制价格分的恶性竞争；
- 我们还可以选择更多的投标人，这样能加大串标的成本，稀释串标的影响；
- 建议有关部门加大对串标行为审查和惩罚力度，让投标人不想串、不敢串、不能串。

五、 模型求解

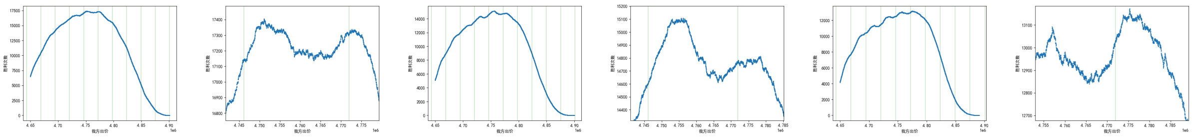
5.1 问题 1

依据上述分析，我们利用**随机取样模拟积分**，我们穷举 N = 2、3......10 分别进行模拟获得图像如下：(概率分布图像在在形状和U_1*上与全局模拟图象是一致的，因此我们大可略去概率分布图像而获得我们需要的 U_1* 值)



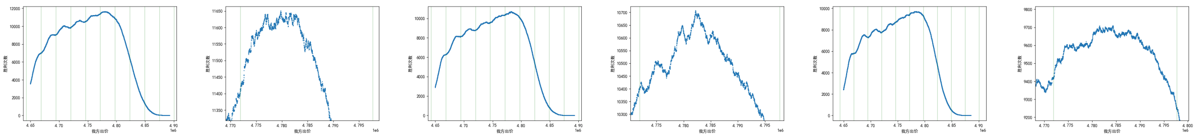
2 全局　2 峰值　3 全局　3 峰值　4 全局　4 峰值

图 4　N=2、3、4 模拟图像



5 全局　5 峰值　6 全局　6 峰值　7 全局　7 峰值

图 5　N=5、6、7 模拟图像



8 全局　8 峰值　9 全局　9 峰值　10 全局　10 峰值

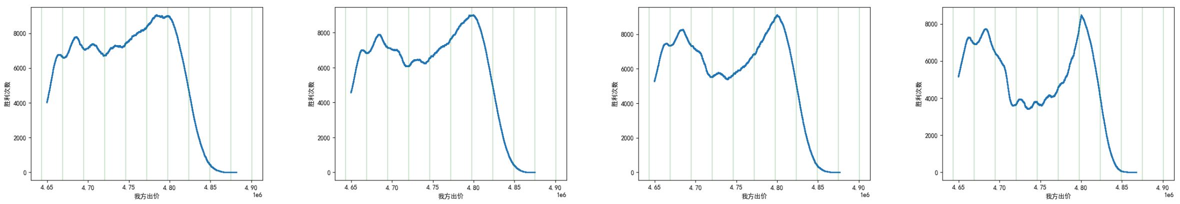
图 6　N=8、9、10 模拟图像

结合图像，我们给出我方报价为（单位：元）:

$$U_1 = \begin{cases} 4784565 & \text{if} N = 2 \\ 4787763 & \text{if} N = 3 \\ 4787666 & \text{if} N = 4 \\ 4785013 & \text{if} N = 5 \\ 4783308 & \text{if} N = 6 \\ 4781079 & \text{if} N = 7 \\ 4785914 & \text{if} N = 8 \\ 4781498 & \text{if} N = 9 \\ 4785681 & \text{if} N = 10 \\ \text{InvalidValue} & \text{if} N > 10 \end{cases} \tag{1}$$

5.2 问题 2

基于前文分析，我们同样通过函数模拟的方式来证实我们的策略正确性：(以 N=10 为例)



C 中人数为 1　　C 中人数为 2　　C 中人数为 4　　C 中人数为 6

图 7　N=10 时，不同 C 中人数的模拟图像

注：C 中人数即 我们能够根据历史经验猜测大致投标评审价格范围的投标人的数量

可以看到随着可确定范围人数的增加，我方在该范围内投标时中标的概率逐渐降低；我方的报价策略应当是：**避免与其他投标人竞争与抢占原有峰值。**

具体分以下情况讨论：

1. C 中人数相比于 N 较少(约 10 取 2)时，其对于整体趋势影响较小，我们可以在问题 1 中获取的结论基础上稍作修改即可；基于三元组方案中 A 的分布在可报价区间中偏向左侧，导致了此时峰值右侧显然高于左侧；因此**我们选择靠右的峰值处的U_1*作为我方报价;**
2. C 中人数相比于 N 中等 (约 10 取 5) 时，其对于整体趋势影响较大，依旧是选择靠右的峰值高于左侧，但是我们同时考虑到：
   - 由于扣分机制，曲线左侧变化慢于右侧，一些未知因素带来的扰动影响将更小；**因此当我们希望有更大的把握中标时，我们应该选择靠左的峰值处的U_1*作为我方报价;**
   - 能够中标的情况下，我方报价越高对于我方越有利；**因此当我们希望能够收益更大时我们应该选择靠右的峰值处U_1*作为我方报价;**
3. C 中人数相比于 N 较多 (约 10 取 8) 时，其对于整体趋势影响较大，我们可以认为相当于将整个可报价区间缩小至猜测的“大致范围”；因此只需要**将问题 1 中范围作修改后获得的 U_1*作为我方报价。**

5.3 问题 3

根据前文分析，我们同样通过模拟来获取$H(U_1) - U_1$的图像如下：

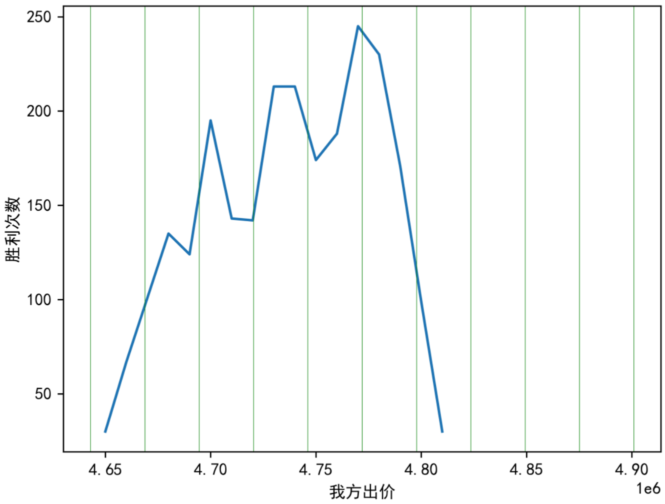


图 8　$H(U_1) - U_1$

于是我们开始考虑每种选择的概率。选择一个点的概率和迭代结果收敛于它的密集程度$H(U_1)$呈正相关，也与没有采取数学建模方法的投标者，只剩随机等概率选择报价的投标者时，我方于改点报价的获胜概率$G(U_1)$呈正相关。于是我们将两者相乘，得到一个决策函数$G * H(U_1)$，其中选取U_1的概率与该点的函数值$G * H(U_1)$成正比。

模拟$G * H(U_1) - U_1$的图像如下：

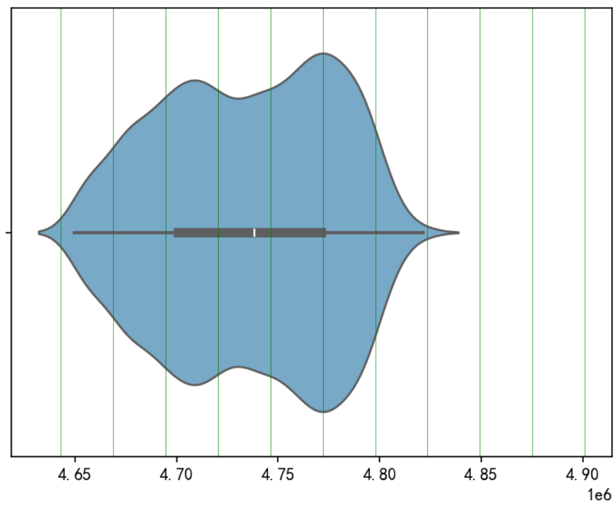


图 9　$G * H(U_1) - U_1$ 小提琴图

在函数与 x 轴围成的图形中随机等概率选择一点，然后以该点的横坐标U_1作为我方投标评审价，这就是我方的最优策略。

六、 模型评价

6.1 优点

优点：

1. 全面性问题覆盖：模型覆盖了从完全无信息到部分信息再到完全信息的情况，能够适应不同情况下的决策需求。
2. 动态调整能力：模型考虑了投标人数量的变化对最优报价策略的影响，能够动态调整报价策略以适应市场变化。
3. 模拟实际市场行为 :通过数值模拟的方法 ,模型能够模拟实际市场中投标人的行为，为决策提供更加真实的参考。
4. 优化决策过程：模型通过迭代方法寻找稳定状态下的最优报价策略，有助于在动态环境中做出更优的决策。
5. 综合评价与改进建议 :模型不仅分析了现有评标机制的有效性 ,还提出了改进建议，有助于提高评标机制的稳定性和公平性。
6. 适应性分析：模型分析了存在串标行为的情况，为预防和识别串标行为提供了理论依据。
7. 实用性：模型提供了具体的报价策略和决策依据，具有较高的实用价值，可以为实际工程招标决策提供指导。
8. 易于理解和操作：模型构建过程中使用了较为直观的数学工具和方法，易于理解和操作，有利于模型的推广和应用。

6.2 缺点

1. 在假定其他投标人的投标评审价时，将其每种策略的出价概率视为相等，这不符合现实生活中的投标报价分布。
2. 在计算函数值$G(U_1)$时，使用了多次随机取U_2到U_N的值，并将结果累加，来模拟积分的结果。这导致数据有必定会存在误差，并且在某些区间连续性较差。
3. 模型中，多个投标人扣分相同的情况被判定为共赢，这不符合现实情况。为避免平局的影响，模型让每位投标人的报价在一定区间内波动，这又导致了新的误差。
4. 本模型更适合分析纯策略，对于最优策略是混合策略的情况，模型无法直接给出准确结果，只能通过多次随机取值及迭代模拟出结果，存在误差。
5. 外部因素考虑不足：模型可能未充分考虑所有可能影响决策的外部因素，如市场趋势、政策变化等。
6. 局部最优问题：模型可能仅找到局部最优解，而非全局最优解，特别是在存在多个局部最优解的情况下。
7. 实际应用限制：模型在实际应用中可能受到法律法规、行业标准等因素的限制，需要进一步调整以适应实际操作。

参考文献

[1]杨全军. 铁路施工企业在项目投标中的博弈分析和对策研究[D].华北电力大学(河北),2008.

[2]张莹.我国招标投标的理论与实践研究[D].浙江大学,2002.

[3]郝丽萍,谭庆美,戈勇.基于博弈模型和模糊预测的投标报价策略研究[J].管理工程学报,2002,(03):94-96.

[4]黄宏飞,欧国立.博弈论在投标报价决策中的应用[J].北方交通大学学报,2000,(03):41-43+49.

[5]江伟,黄文杰.博弈论在工程招投标中的应用分析[J].工业技术经济,2004,(01):58-60.

附录 A Python code for problem 1

```python
# ide: visual studio code 1.89.1
# compiler: python 3.11.4
# libraries: matplotlib 3.7.2
import random
from collections import Counter
from matplotlib import pyplot as plt
from matplotlib.font_manager import FontProperties
from multiprocessing import Pool
plt.rcParams['font.sans-serif']=['SimHei']
plt.rcParams['axes.unicode_minus']=False
cnt=Counter()
A=[4642976]
B=[0.15]
for i in range(10):
    A.append(A[-1]+25794.3)
    B.append(B[-1]+0.01)
N=10
def f(x):
    for T in range(100):
        if T%10==0:
            print(x,T)
        U=[random.randint(4649516,4907459) for i in range(N-1)]
        V=U.sort()
        for i in range(4649516,4907459+1,10):
            average=(sum(V)+i)/(len(V)+1)
            second=sorted(V+[i])[1]
            for j in range(len(A)):
                W=(average+second)*B[j]+A[j]*(1-2*B[j])
                S=[]
                for k in V:
                    S.append(2*(k-W) if k>W else W-k)
                myscore=2*(i-W) if i>W else W-i
                if myscore<=min(S):
                    cnt.update([i])
    return cnt
if __name__ == '__main__':
    with Pool(50) as p:
        results=p.map(f,range(50))
        for result in results:
            cnt+=result
    plt.scatter(list(cnt.keys()),list(cnt.values()),s=1)
    plt.xlabel('我方出价')
    plt.ylabel('胜利次数')
    for Aj in A:
        plt.axvline(Aj,linewidth=0.3,color='g')
    plt.show()
```

附录 B Python code for problem 2

```
# ide: visual studio code 1.89.1
# compiler: python 3.11.4
# libraries: matplotlib 3.7.2
import random
from collections import Counter
from matplotlib import pyplot as plt
from matplotlib.font_manager import FontProperties
from multiprocessing import Pool
plt.rcParams['font.sans-serif']=['SimHei']
plt.rcParams['axes.unicode_minus']=False
cnt=Counter()
A=[4642976]
B=[0.15]
for i in range(10):
    A.append(A[-1]+25794.3)
    B.append(B[-1]+0.01)
def f(x):
    for T in range(100):
        if T%10==0:
            print(x,T)
        U=[random.randint(4649516,4700000) for i in range(4)]
        for m in range(5):
            U.append(random.randint(4649516,4907459))
        V=sorted(U)
        for i in range(4649516,4907459+1,10):
            average=(sum(V)+i)/(len(V)+1)
            second=i if len(V)==1 else min(V[1],i)
            for j in range(len(A)):
                W=(average+second)*B[j]+A[j]*(1-2*B[j])
                S=[]
                for k in V:
                    S.append(2*(k-W) if k>W else W-k)
                myscore=2*(i-W) if i>W else W-i
                if myscore<=min(S):
                    cnt.update([i])
    return cnt
if __name__ == '__main__':
    with Pool(50) as p:
        results=p.map(f,range(50))
        for result in results:
            cnt+=result
    plt.scatter(list(cnt.keys()),list(cnt.values()),s=1)
    plt.xlabel('我方出价')
    plt.ylabel('胜利次数')
    for Aj in A:
        plt.axvline(Aj,linewidth=0.3,color='g')
    plt.show()
```

附录 C Python code for problem 3

```python
# ide: visual studio code 1.89.1
# compiler: python 3.11.4
# libraries: matplotlib 3.7.2
#            seaborn 0.13.2
import random
from collections import Counter
from matplotlib import pyplot as plt
from matplotlib.font_manager import FontProperties
import seaborn as sns
from multiprocessing import Pool
plt.rcParams['font.sans-serif']=['SimHei']
plt.rcParams['axes.unicode_minus']=False
color=['r','g','b','c','m','y','k','orange','purple','brown']
Cnt=Counter()
ans=[]
Ans=[]
A=[4642976]
B=[0.15]
for i in range(10):
    A.append(A[-1]+25794.3)
    B.append(B[-1]+0.01)
def f(x):
    for tt in range(10):
        print(x,tt)
        U=[random.randint(4649516,4907459) for i in range(10)]
        for T in range(50):
            for ip in range(5):
                U[ip]=0
                cnt=Counter()
                for i in range(4649516,4907459+1,100):
                    average=(sum(U)+i)/len(U)
                    second=sorted(U+[i])[2]
                    for j in range(len(A)):
                        W=(average+second)*B[j]+A[j]*(1-2*B[j])
                        S=[]
                        for k in U:
                            S.append(2*(k-W) if k>W else W-k)
                        myscore=2*(i-W) if i>W else W-i
                        if myscore<=min(S):
                            cnt.update([i])
                U[ip]=max(cnt,key=cnt.get)+random.randint(-10,10)
        for ip in range(5):
            Cnt.update([round(U[ip],-4)])
            ans.append(U[ip])
    ans.append(Cnt)
    return ans
if __name__ == '__main__':
    with Pool(50) as p:
        anss=p.map(f,range(50))
        for ans in anss:
            Cnt+=ans[-1]
            Ans.extend(ans[:-1])
    plt.figure(1)
    sns.lineplot(x=list(Cnt.keys()),y=list(Cnt.values()),markers='o')
    plt.xlabel('我方出价')
    plt.ylabel('胜利次数')
```

```
for Aj in A:
    plt.axvline(Aj,linewidth=0.3,color='g')
plt.savefig('240518C1.png',dpi=600)
plt.figure(2)
sns.violinplot(data=Ans,palette="Blues",orient='h')
for Aj in A:
    plt.axvline(Aj,linewidth=0.3,color='g')
plt.savefig('240518C2.png',dpi=600)
plt.show()
```