组合拍卖

阳先毅

浙江大学计算机科学与技术学院

879946238@qq.com

2025 年 5 月 23 日

目录

目录

引入

组合拍卖：同时拍卖多个物品，允许买家对不同的物品组合由不同的价值。

一个情形是出售机场的降落权：

- 一家航空公司希望运营 3 趟早班航班，则可能会对 7,9,11 点的降落权竞标。
- 另一家航空公司只需要 1 趟早班航班，因此可能对 7~ 11 点的每个时间段出价，同时表达它只需要一个时间段的航班。

其他例子：**无线电频谱拍卖、公交线路运营权拍卖、供应链反向拍卖……**

正式定义

在正式定义前，可以通过一个简单的例子来看为什么需要组合拍卖：假设我们仅仅允许对单个物品出价，考虑只有两个物品 A,B，对于买家 1 而言：

$$v_1(A) = 60$$
$$v_1(B) = 40$$
$$v_1(A, B) = 200$$

如果买家 1 确定能赢得 A，那么他对 B 的最大支付意愿就是 140，但是在尚未赢得 A 的前提下，他的最大支付意愿只有 40，超过这个数他就会面临财务风险 (financial exposure)，因而对单个物品的报价不足以表达 {A,B} 组合对 1 来说互补的情况。

正式定义

在组合拍卖 (CA) 中，有一个物品集合 G=A,B,C,...，其中包含 m 个彼此不同且不可分割的物品可供出售；同时，有一组竞标者 N=$\{$1,...,n$\}$。每个竞标者 $i \in N$ 有一个估值函数 v_i ，它对每一个物品子集 $S \subseteq G$ 都给出一个非负的估值 $v_i(S) \geq 0$。

比如买家只对 i 只对 $\{$A,B$\}$ 组合感兴趣，他的估值函数为：

$$v_i(S) = \begin{cases} 100 \text{ , } \mathit{if}\ S = \{A, B\} \\ 0 \text{ , } otherwise \end{cases}$$

要求估值函数必须满足两个性质：

- 空集合的估值为 0， $v_i(\emptyset) = 0$
- 添加物品不会降低估值，对于 $S \subseteq S'$，有 $v_i(S) \leq v_i(S')$。
- 效用假设——准线性效用： $u_i = v_i(S) - p$。

目录

设计要素

组合拍卖的设计要素主要有以下三点：

- 使用什么样的投标语言来描述投标函数？
- 胜者判定：找出一个可行的分配方案，使得总投标值最大化。
- 支付规则。

在研究投标语言之前，先介绍 VCG 机制在组合拍卖中的表示（其实都一样）。首先，所有人以某种投标语言表达了对各种组合的支付意愿，然后拍卖求解胜者判定问题，得到最优分配：$X^* = (X_1^*, X_2^*, X_3^*, ..., X_n^*)$，其中 X_i^* 是分配给 i 的物品集合。

支付规则：

$$t_{\text{vcg},i}(b) = \sum_{j \neq i} b_j(X_j^{-i}) - \sum_{j \neq i} b_j(X_j^*)$$

它所具有的性质：strategy-proof，其实和激励相容基本一致，即城市占优；配置有效率——因为最大化了总估值；预算合理性，其实就是个人理性。

设计要素

组合拍卖的设计要素主要有以下三点:

- 使用什么样的投标语言来描述投标函数?
- 胜者判定:找出一个可行的分配方案,使得总投标值最大化。
- 支付规则。

在研究投标语言之前,先介绍 VCG 机制在组合拍卖中的表示(其实都一样)。首先,所有人以某种投标语言表达了对各种组合的支付意愿,然后拍卖求解胜者判定问题,得到最优分配:$X^* = (X_1^*, X_2^*, X_3^*, \dots, X_n^*)$,其中 X_i^* 是分配给 i 的物品集合。

支付规则:

$$t_{\text{vcg},i}(b) = \sum_{j \neq i} b_j(X_j^{-i}) - \sum_{j \neq i} b_j(X_j^*)$$

它所具有的性质:strategy-proof,其实和激励相容基本一致,即城市占优;配置有效率——因为最大化了总估值;预算合理性,其实就是个人理性。

VCG 的一个例子

	A	B	C	AB	BC	AC	ABC
Bidder 1	60	50	50	200*	100	120	250
Bidder 2	50	50	75*	80	140	140	255
Bidder 3	50	60	50	90	100	70	110

对于上面这个例子，显然最优分配结果是 $X^* = \{\{A, B\}, \{C\}, \emptyset\}$，1 的支付为 180，2 的支付为 50，3 的支付为 0。

基本投标语言-XOR

原子报价 (atomic bid):(S, b_S) 构成的二元组，其中 $S \subseteq G$

XOR 投标语言：在 $\mathcal{L}_{\oplus}$ 中，竞标者可以提交多个原子报价，但只能接受其中一个原子报价（实际上表示它们是互斥的），形式是：

$$bid_i = (S_1, b_1) \oplus (S_2, b_2) \oplus ... \oplus (S_l, b_l)$$

对于某个组合而言，其出价 $b_i(S) = max\{b_{S'} : S' \in bid_i, S' \in S\}$，这里再给一个例子：
bid$_i = (A, 100) \oplus (B, 80) \oplus (AB, 140) \oplus (C, 20) \oplus (BC, 200) \oplus (AD, 150)$
那么：

$$b_i(AB) = \max(100, 80, 140) = 140$$
$$b_i(ABC) = \max(100, 80, 140, 20, 200) = 200$$

优点：表达能力非常强，可以表达任何估值函数。
缺点：当有 m 个物品时，投标组合是 $2^m - 1$，是指数级的。

基本投标语言-OR

这里的 OR 语言指的是加性 OR 语言，$\mathcal{L}_{\vee}$，它可以接受任意多个原子报价，比如说如果有 3 个物品 A,B,C，竞标者 1 对任意子集 $S \subseteq \{A, B, C\}$ 的估值是：

$$v_1(S) = |S|$$

那他完全可以用 OR 语言表示为：

$$bid_1 = (A, 1) \vee (B, 1) \vee (C, 1)$$

对于某个组合而言，计算其出价应该在满足条件——同一个物品不能够重复出现的情况下，计算各种组合出价和的最大值。一个例子如下：bid$_i = (A, 100) \vee (B, 80) \vee (BC, 100) \vee (C, 10) \vee (CD, 40)$ 那么：

$$b_i(BC) = \max(80, 10, 80 + 10, 100) = 100$$
$$b_i(BCD) = \max(80, 10, 80 + 10, 100, 40, 80 + 40) = 120$$

基本投标语言-OR

OR 语言的优点：**简洁，不需要显示列举出所有的组合以及对应估值。**

OR 语言的缺点：**不能够表示所有类型的估值函数，尤其当估值函数不是超加性的。**比如说 $v_2(A)=50, v_2(B)=50, v_2(AB)=80$，在 OR 语言中系统对会认为 2 对 AB 组合愿意支付 100，但实际上只有 80。实际上，OR 语言只能表示超加性估值：

$$v_i(S \cup S') \geq v_i(S) + v_i(S'), \quad \text{for all disjoint bundles } S \text{ and } S'$$

而不能表示次加性估值 (不取等时)：

$$v_i(S \cup S') \leq v_i(S) + v_i(S'), \quad \text{for all disjoint bundles } S \text{ and } S'$$

或其他既不是超加性也不是次加性的估值函数。

OR，XOR 组合的语言

OR of XOR 语言实际上就是将 OR 逻辑运用于多个 XOR 投标，相当于可以接受多个 $\vee$ 连接的 XOR 投标，但在一个 XOR 投标中只能接受一个。比如说：

$$\text{bid}_i = ((AB, 10) \oplus (C, 12)) \vee ((DE, 10) \oplus (F, 12))$$

XOR of OR 与前者恰恰相反，将 XOR 逻辑运用于多个 OR 投标，只可以接受一个 $\oplus$ 连接的 OR 投标，在一个 OR 投标中可以接受多个，比如说：

$$\text{bid}_i = ((A, 10) \vee (B, 10) \vee (D, 20)) \oplus ((C, 10) \vee (D, 15) \vee (DE, 12))$$

OR^* 语言

Definition 11.5 (OR* bidding language). *In the* or-star (OR*) bidding language *a bidder i submits one or more atomic bids, $bid_i = (S_1, b_1) \vee (S_2, b_2) \vee \ldots \vee (S_\ell, b_\ell)$, where an atom can include items G and dummy items $D_i = \{d_1, d_2, \ldots\}$, and any number of atomic bids may be accepted as long as no item or dummy item appears in more than one atom.*

其实就是在 OR 的基础上增加了一个“虚拟物品”(dummy item)，其实就是利用相同的虚拟物品 d 防止某些组合按照 OR 求和，增强了表达能力，比如说对于一个航空公司，它可以投标：

$$(\{9\text{am}, d_1\}, 100) \vee (\{10\text{am}, d_1\}, 110) \vee (\{11\text{am}, d_1\}, 90) \vee$$
$$(\{12{:}30\text{pm}, 2{:}30\text{pm}, d_2\}, 400) \vee (\{1{:}30\text{pm}, 3{:}30\text{pm}, d_2\}, 400).$$

投标语言对比

主要针对表达能力和简洁性。

首先是表达能力： XOR、OR-of-XOR、XOR-of-OR 以及 OR^* 投标语言 (对于一般估值函数是有表达能力的，而 OR 语言则不是。

对于 XOR，这是自然的，系统计算出来的 $b_i(S) = v_i(S)$。而 XOR 可以规约到另外 3 中语言：

- 在每个原子投标中加入同一个虚拟物品 d_1。
- 作为 OR of XOR 当中的一个子句。
- 在 XOR of OR 中，每个原子投标放到一个 OR 子句中。

对于 OR 语言，前面提到，系统计算出来的 $b_i(S)$ 可能会大于 $v_i(S)$。

投标语言对比

然后考虑简洁性：

Definition 11.8 (Succinctness of a bidding language)**.** *A bidding language* $\mathcal{L}$ *is* succinct for a domain of valuation functions V*, if and only if, for every valuation function* $v \in V$*, there is a corresponding bid in language* $\mathcal{L}$ *whose number of atoms scales polynomially in the number of items,* m*.*

We are also interested in comparing the succinctness of a pair of languages, $\mathcal{L}_1$ and $\mathcal{L}_2$, on a valuation domain in which they are both expressive. We say $\mathcal{L}_1$ is *as succinct as* $\mathcal{L}_2$ *on* V when, for every $v \in V$ and corresponding bid in $\mathcal{L}_2$, there is a corresponding bid in $\mathcal{L}_1$ that is at most polynomially larger in terms of the number of items, m.

上面讲的是简洁性至少不差于，如果更强——即严格更简洁，则需要在此基础上，存在 $v \in V$, 使得用 L_1 表示相对 m 是多项式的，而用 L_2 表示是指数的。

投标语言对比

- OR 语言在简单加性的估值函数上比 XOR 语言更简洁。

在考虑 OR^*、XOR of OR,OR of XOR 的相对简洁性时，可以引入一下两个估值函数比较：

- **单色估值函数（monochromatic valuation function）**：对于任意组合 S，有
$$v_i(S) = \max(|\text{red}(S)|, |\text{blue}(S)|),$$
其中物品分为两种类型：红色（red）和蓝色（blue），$\text{red}(S)$ 和 $\text{blue}(S)$ 分别表示组合 S 中红色和蓝色物品的数量；
- **k-预算估值函数（k-budgeted valuation function）**：对于任意组合 S，有
$$v_i(S) = \min(k, |S|),$$
其中 k 是某个固定的正整数。

投标语言对比

OR-of-XOR 投标语言对于 k-预算估值是简洁的，但对于单色估值不是简洁的。

证明.

对于 k-预算估值，假设 $k=2$ 并考虑有三个物品的情况，一个 OR-of-XOR 投标结构如下：

$$\text{bid}_i = ((A,1)\oplus(B,1)\oplus(C,1)) \vee ((A,1)\oplus(B,1)\oplus(C,1)).$$

对于一般的 k-预算估值，我们可以使用 k 个 XOR 子句，每个子句包含 m 个原子出价，来代替上面的两个 XOR 子句。
对于单色估值，在假设红色物品和蓝色物品个数分别为 x,y,x+y=m。则每个 XOR 子句需要的原子个数分别为 $z^x-1, 2^{m-x}-1$，即使 x=y，一共也需要 $2\times(2^{m/2}-1)$ 的原子数，显然不简洁。 □

投标语言对比

XOR-of-OR 投标语言对于单色估值是简洁的，但对于 k-预算估值不是简洁的。

证明.

对于单色估值函数，假设物品总数的 m，其中一半是蓝色物品，一半是红色物品，报价可以如下表示:

$$\text{bid}_i = \big((\text{blue}_1, 1) \vee \cdots \vee (\text{blue}_{m/2}, 1)\big) \oplus \big((\text{red}_1, 1) \vee \cdots \vee (\text{red}_{m/2}, 1)\big).$$

而对于 k-预算估值，XOR-of-OR 语言则不够简洁，证明在论文：Bidding and Allocation in Combinational Auctions 下面仅仅展示结果：

我们可以证明当 $K = \sqrt{m}/2$ 时，在 XOR-of-OR 投标语言中表示 K-预算估值所需的表达原子数至少为 $2^{m^{1/4}}$

□

投标语言对比

OR^* 在简洁性方面至少与 XOR，OR of XOR,XOR of OR 一样。

- 在 OR^* 中可以在每个原子投标中引入同一个虚拟物品表示 XOR 投标。
- OR of XOR 投标同样可以被 OR^* 语言表示（依然是通过引入虚拟物品），含 l 个原子的可以引入 l 个虚拟物品表示。
- XOR of OR 投标可以用 OR^* 语言表示，只需要引入 l^2 个虚拟物品。

前面已经说明了 OR^* 与这三种语言一样简洁，然而下面可以说明 OR^* 语言比剩下 3 种更简洁：

- **简单加性函数**：胜过 XOR。
- **单色估值函数**：胜过 OR-of-XOR 。
- **K-预算函数**：胜过 XOR-of-OR 。

胜者确定问题

在胜者确定问题中，由于 OR^* 语言同时具有简洁以及有表达能力的优势，所以可以假设投标使用 OR^* 语言。同时使用 OR^* 还有一个好处，比如两个人报价 $(ABd_1, 100) \vee (Cd_1, 50)$,$(ACd_1, 50) \vee (Bd_1, 80)$, 我们自然可以合成为 $(ABd_1, 100) \vee (Cd_1, 50) \vee (ACd_2, 50) \vee (Bd_2, 80)$。

胜者判定问题：输入是一个由 $q \geq 1$ 个原子出价组成的 OR^* 投标：$\{(S_k, b_k)\}_{k\in\{1,\ldots,q\}}$ 其中每个原子出价包含一组真实物品和可能的虚拟物品。该问题的目标是找出一个可行的原子出价集合（即没有任何真实物品或虚拟物品出现在多个原子中），使得总出价金额最大。
它当然可以表示成一个整数规划问题：

$$\begin{aligned} \max \quad & \sum_{k=1}^{q} b_k \cdot z_k \\ \text{s.t.} \quad & \sum_{k=1}^{q} a_{kj} \cdot z_k \leq 1, \quad \forall j \in G^* \\ & z_k \in \{0,1\}, \quad \forall k \in \{1,\ldots,q\} \end{aligned}$$

胜者确定问题

$$
\begin{aligned}
\max \quad & \sum_{k=1}^{q} b_k \cdot z_k \\
\text{s.t.} \quad & \sum_{k=1}^{q} a_{kj} \cdot z_k \leq 1, \quad \forall j \in G^* \\
& z_k \in \{0,1\}, \quad \forall k \in \{1,\dots,q\}
\end{aligned}
$$

- 决策变量 $z_k \in \{0,1\}$ 表示是否接受第 k 个原子出价。
- 参数 $a_{kj} \in \{0,1\}$ 表示原子 S_k 是否包含物品 $j \in G^*$（可能是虚拟物品）

胜者确定问题

一个简单的例子如下：
共有三位投标者和四个物品 $\{A, B, C, D\}$。OR* 投标如下：
- 投标者 1 的出价为：$(A, 2) \vee (ABD, 3)$
- 投标者 2 的出价为：$(BC, 2)$
- 投标者 3 的出价为：$(CD, 1)$

在这个例子中没有使用任何虚拟物品（dummy items）。按照原子出价的顺序编号为 z_1, z_2, z_3, z_4，则该胜者判定问题（WDP）可表示为以下整数规划（IP）模型：

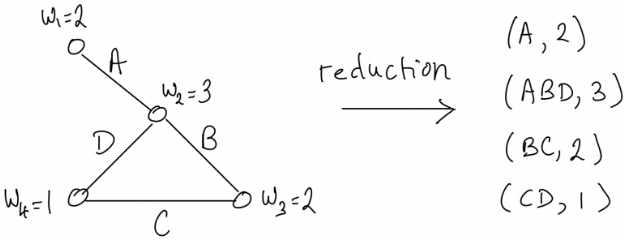
$$
\begin{aligned}
\max \quad & 2z_1 + 3z_2 + 2z_3 + 1z_4 \\
\text{s.t.} \quad & z_1 + z_2 \leq 1 \\
& z_2 + z_3 \leq 1 \\
& z_3 + z_4 \leq 1 \\
& z_2 + z_4 \leq 1 \\
& z_1, z_2, z_3, z_4 \in \{0, 1\}
\end{aligned}
$$

胜者确定问题——复杂性

前面的例子我们极其容易看出解，但是胜者确定问题其实是 NP-hard 的，我们可以把加权独立集问题规约到它。

定义 (加权独立集问题)

输入是一个无向图 (V, E)，其中每个顶点 $k \in V$ 都有一个对应的正权重 $w_k > 0$。该问题的目标是找出一个总权重最大的独立集。



{1,3} 是独立集，而 {2,4} 不是独立集。

胜者确定问题——复杂性

我们在这里考虑的是加权独立集问题的决策版本（e.g. 是否存在一个独立集，其总权重 $\geq$5?）

该问题已经被证明是 NP-hard 的。

我们现在将“加权独立集问题是否存在一个总权重大于等于 Z 的独立集?”这个问题规约到一个具有相同阈值 Z 的组合拍卖的获胜者确定问题。

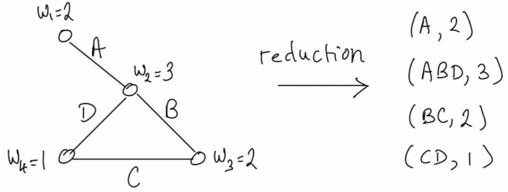
给定一个图 (V, E)，我们为每条边 $e_j \in E$ 引入一个唯一的物品 j，并为每个顶点 $k \in V$ 引入一个原子出价 (S_k, w_k)，其中:

- $w_k > 0$ 是顶点 k 的权重。
- 两个顶点 $k, k' \in V$ 被一条边连接当且仅当对应的原子 S_k 和 $S_{k'}$ 并不独立。

因此，任意一组互斥的（可同时接受的）原子出价，恰好对应图中的一个独立集。存在一个总权重至少为 Z 的独立集，当且仅当在胜者判定问题中存在一个总出价金额至少为 Z 的可行分配。

胜者确定问题——复杂性

同样有一个非常简单的例子：



上图独立集与可行分配之间存在一一对应关系。特别地，最优分配选择了原子出价 1 和 3，总价值为 $2 + 2 = 4$，并且这对应于图中的最大加权独立集。

假设 $P = NP$，那么在商品数量和原子出价数量上，不存在能在最坏情况下的多项式时间内解决组合拍卖（CAs）中的 WDP 的算法, 故考虑：

1. 设计算法以在实际有用的运行时间内解决典型实例上的 WDP；
2. 判断能够在多项式时间内求解的 WDP 特殊情况；
3. 设计提供最坏情况近似质量保证的多项式时间近似算法。

分支限界搜索算法

分支定界搜索是求解整数规划的一种典型方法，它通过多次调用求解线性规划的算法来实现。对于 q 个原子出价，其实就是松弛将 IP 模型中的 0-1 决策变量：

$$
\begin{aligned}
\max \quad & \sum_{k=1}^{q} b_k \cdot z_k \\
\text{s.t.} \quad & \sum_{k=1}^{q} a_{kj} \cdot z_k \leq 1, \quad \text{对所有} j \in G^* \\
& z_k \geq 0, \quad \text{对所有} k \in \{1, \ldots, q\}.
\end{aligned}
$$

LPR 的可行解包括分数分配，即某些 z_k 满足 $0 < z_k < 1$ 的情况，也包括所有的整数分配。LPR 的最优解为 IP 的最优解值提供了一个上界，因为它是对原问题的一个松弛。如果 LPR 的最优解恰好是 0-1 解，那么它同时也解决了原始的 IP 问题。**因而我们实际就是考虑把解限制在 $\{0,1\}$ 内。**

分支限界搜索算法

下面同样举一个例子说明，假设存在物品 A 到 F，以及以下原子出价：

$$
\begin{gathered}
(D,1),\ (AB,3),\ (BC,2.5),\ (ACD,3),\ (CDE,1.5),\\
(EF,4.5),\ (F,3.5),\ (BD,1)
\end{gathered}
$$

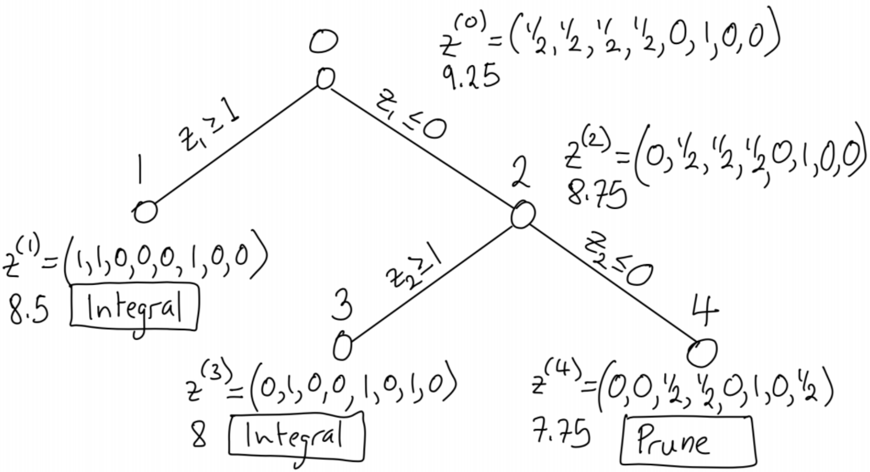
WDP 的最优 0-1 解将分配给原子出价 1、2 和 6，总价值为 8.5。

但转换为 LP 后得到一个分数解：

$$z=(1/2,\ 1/2,\ 1/2,\ 1/2,\ 0,\ 1,\ 0,\ 0)$$

对应的价值为 9.25。该解对前四个原子出价中的每一个都分配了一半的比例，第六个原子出价被完全选中。**这时候我们可以对问题进行分治，降低一个出现分数的位置用 0/1 替代，由此产生两个子节点，具体如下所示：**

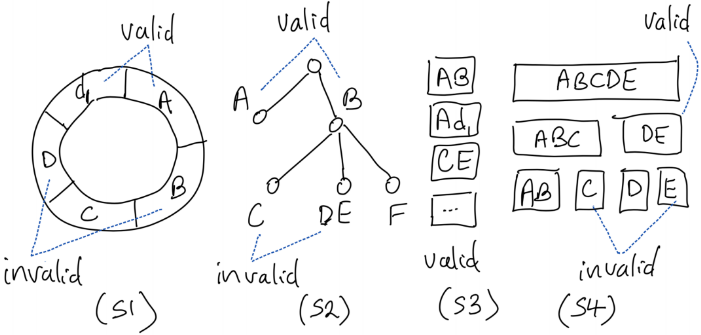
分支限界搜索算法



由于存在剪枝，所以在实际问题中，往往可以非常有效的求解。

特殊情况

下面来列举一下多项式时间（即使是最坏的情况）能够求解 WDP 的特殊情况，一共包含四种：



- 循环结构: 存在一种对物品（包括任何虚拟物品）的循环排列顺序，使得每一个 OR* 出价中的原子对应于该顺序中连续的一组物品。例如，这些物品可以代表一个岛屿沿岸待售的土地地块，或者房源的连续日期。

特殊情况

- 树形结构：存在一个物品（包括任何虚拟物品）构成的树形结构，使得每一个 OR* 出价中的原子对应于该树中连续的一组物品。每个物品唯一地与树中的一个顶点相关联，两个顶点之间的“树距离”是指它们之间路径上的边数。对于每一个原子，必须能找到一个基顶点和一个整数 $r \geq 0$，使得该原子等于所有距离基顶点不超过 r 的顶点所对应的物品的并集。例如，这些顶点可以表示某个无环道路网络连接的商业房地产开发中的不同区域。
- 成对出价：每个 OR* 出价中的每一个原子最多包含两个物品（包括任何虚拟物品）。例如，这可能适用于机场着陆时段的拍卖，其中没有航空公司希望每天运营超过两个航班。
- 层级结构：存在一个物品及捆绑包的层级结构，使得每一个 OR* 出价中的原子对应于层级结构中的某一个捆绑包（包括任何虚拟物品）。层级的底层对应于物品的一个任意划分，每一上层由下一层的一个或多个捆绑包组合而成。这种结构的效果是：对于任意两个原子，它们要么互不相交，要么其中一个完全包含于另一个之中。

目录

VCG 的缺陷

尽管 VCG 机制有很多优点，但它也有一些严重问题：

	A	B	AB	A	B	AB	A	B	AB	A	B	AB
Bidder 1	0	0	10*	0	0	10*	0	0	10*	0	0	10
Bidder 2	10	0	10	2	0	2	2	0	2	10*	0	10
Bidder 3				0	2	2				0	10*	10
		(a)			(b)			(c)			(d)	

- 收入过低：如 d 所示，即使 1 愿意以 10 元买下 AB。而且从 a 到 d 引入了新的买家却使得总收入下降了。
- 输家串谋：如 b 所示，2、3 两人可以伪造成 d 的样子，从而不用付出代价而获的物品。
- 身份伪造：如 c 所示，2 可以伪造一个 3 的身份，从而进入 d。

核心定义

核心选择拍卖通过要求足够高的价格，来修正 VCG 机制所存在的问题，但不是完美的（后面会说）。

设 (X, p) 表示一个组合拍卖（CA）的结果，其中 X 是分配结果，$p = (p_1, \ldots, p_n)$ 是支付向量，总收益为 $\mathrm{Rev} = \sum_{i \in N} p_i$，投标人 i 的价值为 $w_i = v_i(X_i)$，其效用（或收益）为 $\pi_i = v_i(X_i) - p_i$。
我们关注那些不存在"阻挠联盟"(blocking coalition) 的结果 (X, p)。所谓阻挠联盟是指某个子集 $L \subseteq N$，卖家可以不执行当前结果，而与联盟 L 进行交易，从而产生更高的收益，并使联盟中的每一个投标人 $i \in L$ 都获得更优的效用。令 $W_L = \max_{X \in \mathcal{F}} \sum_{i \in L} v_i(X_i)$ 表示对联盟 L 最优的分配的价值（称为联盟价值），其中 $\mathcal{F}$ 是所有可行分配的集合。

核心定义

核心选择拍卖通过要求足够高的价格，来修正 VCG 机制所存在的问题，但不是完美的（后面会说）。

设 (X, p) 表示一个组合拍卖（CA）的结果，其中 X 是分配结果，$p = (p_1, \ldots, p_n)$ 是支付向量，总收益为 $\mathrm{Rev} = \sum_{i \in N} p_i$，投标人 i 的价值为 $w_i = v_i(X_i)$，其效用（或收益）为 $\pi_i = v_i(X_i) - p_i$。

我们关注那些不存在“阻挠联盟”（blocking coalition）的结果 (X, p)。所谓阻挠联盟是指某个子集 $L \subseteq N$，卖家可以不执行当前结果，而与联盟 L 进行交易，从而产生更高的收益，并使联盟中的每一个投标人 $i \in L$ 都获得更优的效用。令 $W_L = \max_{X \in \mathcal{F}} \sum_{i \in L} v_i(X_i)$ 表示对联盟 L 最优的分配的价值（称为联盟价值），其中 $\mathcal{F}$ 是所有可行分配的集合。

核心定义

定义

对于估值组合 v，核心（core）是满足以下条件的所有核心结果（core outcomes）的集合 $\mathrm{Core}(v)$，当且仅当 $(X,p) \in \mathrm{Core}(v)$ 时：

- Core1 对于所有投标人 $i \in N$，有 $\pi_i \geq 0$；
- Core2 对于所有联盟 $L \subseteq N$，有

$$\mathrm{Rev} + \sum_{i \in L} \pi_i \geq W_L$$

根据核心的性质，没有单个投标人会拒绝参与该拍卖，并且不存在任何阻挠联盟。

前面的图所示的组合拍卖就存在的阻挠联盟。例如，考虑联盟 $L = \{1\}$，此时有 $\mathrm{Rev} = 0 < W_{\{1\}} = 10$。

核心定义

定理 (定理 11.20)

一定存在非空核心，且在任何核心结果中：

1. 分配是有效率的；
2. 支付至少不低于 *VCG* 支付，且未被分配的投标人的支付为零；
3. 拍卖总收入至少等于对未获胜投标人最优分配的价值。

证明.

令 X^* 表示一个有效率的分配，$N^* = \{i \in N \mid X_i^* \neq \emptyset\}$ 表示被分配的投标人集合。
为了证明存在性，考虑有效率分配 X^* 和支付 $p_i = w_i(X_i^*)$（对所有 i），这构成了一个核心结果：满足 Core1，并且对于任意联盟 $L \subseteq N$，有

$$\mathrm{Rev} + \sum_{i \in L} \pi_i = \sum_{i \in N} p_i + \sum_{i \in L} (w_i - p_i) = \sum_{i \in N} w_i \geq W_L,$$

因此也满足 Core2。

核心定义

证明.

(1) 假设分配 X 是无效率的，则对 $L = N$，Core2 被违反，因为：

$$\mathrm{Rev} + \sum_{i\in N} \pi_i = \sum_{i\in N} p_i + \sum_{i\in N}(w_i - p_i) = \sum_{i\in N} w_i < W_N.$$

(2) 考虑 $L = N \setminus \{i\}$ 的情况，此时：

$$\sum_{i\in N} p_i + \sum_{j\neq i}(w_j - p_j) \geq W_{N\setminus\{i\}},$$

由此可得：

$$p_i \geq W_{N\setminus\{i\}} - \sum_{j\neq i} w_j = t^{\mathrm{VCG},i}(v),$$

对于未被分配的投标人，由 Core1 得到 $p_i \leq 0$，同时又有 $p_i \geq t^{\mathrm{VCG},i}(v) = 0$，所以 $p_i = 0$。

核心定义

证明.

(3) 对于 $L = N \setminus N^*$，由于未被分配的投标人效用为 $\pi_i = w_i - p_i = 0 - 0 = 0$，所以:

$$\mathrm{Rev} + \sum_{i \in L} \pi_i = \mathrm{Rev} \geq W_{N \setminus N^*}.$$

$\square$

核心选择拍卖会选择一个在“揭示核心”中的结果，即给定出价组合 $b = (b_1, \ldots, b_n)$ 时，选择属于 $\mathrm{Core}(b)$ 的结果 (X, p)。

ps: 为什么叫“揭示核心”，因为我们的核心本来是在估值 v 上选择的，但在实际拍卖中估值是未知的，只能通过报价揭示。

核心选择拍卖

定义（核心选择拍卖）

给定出价 b，一个核心选择拍卖会选择“揭示核心” $\mathrm{Core}(b)$ 中的一个分配和支付向量作为结果。第一价格密封投标拍卖是核心选择的（对于揭示核心而言，设置支付等于出价类似于对真实核心设置支付等于价值），因此核心选择拍卖不一定是 stragty-proof 的。

定义 (投标人最优核心)

对于估值组合 v，当一个拍卖结果 (X, p) 属于核心（因此 X 是有效率的），并且支付向量 $(p_1, \ldots, p_n)$ 对投标人而言是帕累托最优时，则称该结果属于投标人最优核心（bidder-optimal core）。

VCG 与核心选择拍卖

VCG 机制是否能够避免前面提到的那些问题，取决于 VCG 结果是否落到核心内。为了分析这一点，我们可以使用如下等价的核心结果定义：

- 分配 X 是有效率的，未被分配的投标人支付为零，并且对任意被分配的投标人 i，有 $p_i \le v_i(X_i)$；
- 被分配到物品的人支付，满足：

$$\sum_{j\in K} p_j \ge W_{N\setminus K} - \sum_{j\in N\setminus K} w_j, \quad \text{for all } K \subseteq N^*,\ K \ne \emptyset.$$

下面同样有一个例子：

	A	B	AB
Bidder 1	0	0	10*
Bidder 2	10	0	10

(a)

	A	B	AB
Bidder 1	0	0	10*
Bidder 2	2	0	2
Bidder 3	0	2	2

(b)

	A	B	AB
Bidder 1	0	0	10*
Bidder 2	2	0	2

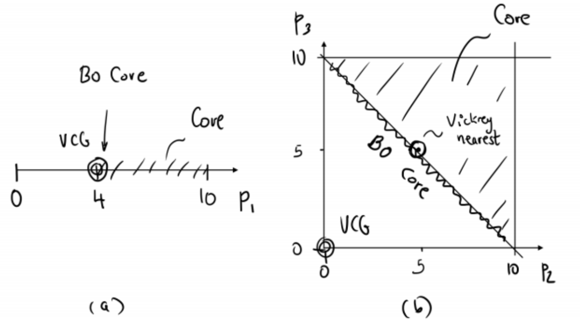
(c)

	A	B	AB
Bidder 1	0	0	10
Bidder 2	10*	0	10
Bidder 3	0	10*	10

(d)

VCG 与核心选择拍卖

其中 b,d 对应的 VCG 支付结果与 core 要求的支付范围如下图所示：



注意到对于 b 来说 VCG 的结果恰好落在了核心内，而对于 d 来说 VCG 的结果在核心之外。核心之所以具有重要意义，是因为当 VCG 结果对于所有可能的输入都落在核心内时，低收益问题、串通行为以及虚假身份投标所带来的负面影响都将消失。

最接近 Vickrey 的核心选择拍卖

定义 (Vickrey-最近核心选择拍卖)

下面我们研究 Vickrey-nearest core-selecting auction。其执行过程如下：

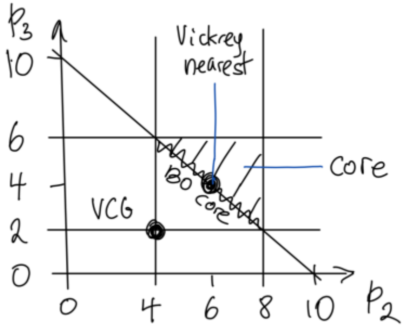
1. 分配规则与 VCG 拍卖相同；
2. 支付规则选择一个属于揭示核心 $\mathrm{Core}(b)$ 的支付向量 $p = (p_1, \ldots, p_n)$，并且该支付向量求解如下问题：

$$t^{\mathrm{core},i}(b) = \mathsf{lexmin}_p \max_{i \in N} \left| t^{\mathrm{VCG},i}(b) - p_i \right|$$

当存在多个解时，按字典序进行选择。

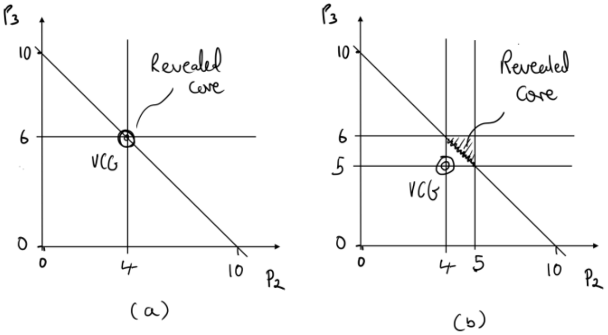
最接近 Vickrey 的核心选择拍卖——例子

	A	B	AB
bidder1	0	0	10
bidder2	8*	0	8
bidder3	0	6*	6



上图展示的是在诚实报价的前提下，VCG 的结果，核心范围、竞拍者最优核心范围、以及最接近 Vickrey 的核心。

最接近 Vickrey 的核心选择拍卖——not stragty-proof



在上图中，假定 1,3 仍保持诚实报价，假设拍卖机制平局时优先考虑投标人 2 和 3 而不是投标人 1。投标人 2 可以出价 $b_2' = (4, 0, 4)$，分别对应对物品 A、B 和 AB 的报价。如 a 所示，这种出价改变了揭示核心的形状，此时核心仅包含一个点：$p_2' = 4$、$p_3' = 6$（这也正是在此输入下的 VCG 结果）。对应的效用为 $\pi_2' = 4$、$\pi_3' = 0$。

最接近 Vickrey 的核心选择拍卖——not stragty-proof

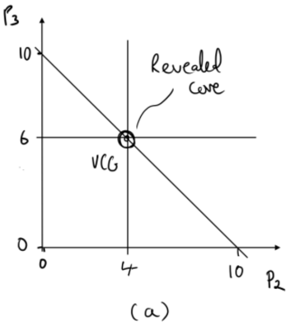


图 b 展示了如果投标人 2 出价 $b_2'' = (5, 0, 5)$ 时的情形。此时核心选择拍卖将设定支付 $p_2'' = 4.5$、$p_3'' = 5.5$，对应的效用为 $\pi_2'' = 3.5$、$\pi_3'' = 0.5$ (在此输入下对应的 VCG 支付为投标人 2 得到 4、投标人 3 得到 5，且该结果不在揭示核心中)。

最接近 Vickrey 的核心选择拍卖——not stragty-proof

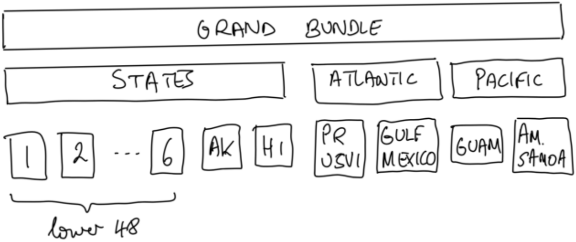
投标人 2 和投标人 3 实质上是在就如何瓜分总计为 4 的可用收益进行讨价还价。在投标人 1 如实出价的情况下，他们实际上就是在自己能够获利的范围内调整报价，使得他们报价之和等于（或微弱大于）10。

当支付之和为 10 时，实际上形成了一系列均衡：每一个结果都在真实核心（true core）上是投标人最优的（bidder-optimal)。

尽管核心选择机制不是 stargty-proof 的，但它确实可以解决 VCG 的那 3 个毛病。其中，我们可以认为 Vickrey-nearest 拍卖“更常趋近于真实报价”(truthful most often)，因为它通过尽量接近 VCG 支付方案，来最小化投标人偏离真实报价的动机。

频谱拍卖——一个小例子

2008 年，美国联邦通信委员会举办了一场无线频谱拍卖，拍卖标的是 700MHz 频段的无线频谱，这些频谱将随着模拟电视的退出而被腾出，以便进行更高效的数字化使用。



- 设置了层级结构。
- 采用 OR 出价语言。
- 采用多轮拍卖机制。

频谱拍卖——一个小例子

在每一轮结束时，新的价格是根据许可证的大小（以 *MhzPop* 为单位进行衡量）来确定的，其中 MhzPop 是频段宽度乘以地理区域内的人口数量。假设对捆绑包 1 到 6 每个都有一笔 500 万美元的出价，对 AK 和 HI 每个捆绑包各有一笔 200 万美元的出价，并且对包含五十个州的整体捆绑包有一个暂定胜出的出价 5500 万美元。此外，假设捆绑包 1 到 6 中每个的大小为 10 MhzPop，而 AK 和 HI 每个的大小为 5 MhzPop。在这种情况下，捆绑包 1 到 6 每个的新价格为：

$$5 + \left(\frac{10}{70}\right)(55 - 34) = 8 \text{ 百万美元}$$

而捆绑包 AK 和 HI 每个的新价格为：

$$2 + \left(\frac{5}{70}\right)(55 - 34) = 3.5 \text{ 百万美元}$$