

纳什均衡与计算复杂性

1900010681 李东晨

2023 年 7 月 19 日

1 博弈论的历史

“博弈”的英文是 *game*，和“游戏”同义。博弈，顾名思义，就是竞争和对抗。在一个给定的游戏规则下，竞争的各方各自具有不同的利益，并力图选择对自己最有利的方案，这就是游戏中的博弈行为。博弈在我们的日常生活中非常常见：小到下棋打牌，大到国际关系，无不是博弈行为的表征。博弈论则是为了描述和预测给定规则下各方的合理行动方案而创造的理论。

早期博弈论的研究主要集中于对某些具体游戏中玩家行为的探讨。例如，17 世纪中叶，Pascal 和 Huygens 在通讯中对若干个赌博的可能结果进行了深入的研究，并提出了“数学期望”这一概念；1713 年，Waldegrave 在解释纸牌游戏中玩家的行为时发现了最大最小原理；1913 年，Zermelo 提出了 Zermelo 定理，即双人有限完全信息确定博弈中要么某一方存在必胜策略，要么双方都存在不败策略¹，为现代博弈论的出现开辟了道路。

区别于早期博弈论，现代博弈论不再聚焦于具体的游戏，而是将博弈的概念抽象成数学模型，并且试图推导一些更一般的性质，如博弈的均衡、可计算性、稳定性等。现代博弈论在 1928 年由 Von Neumann 提出。他开创性地使用了不动点定理研究博弈中的策略均衡。然而，他的分析仅限于零和博弈。1951 年，Nash 将他们对均衡的定义推广，建立了有关博弈的标准模型，并且在冯诺依曼的方法的基础上，证明了博弈中纳什均衡的存在性。

博弈的标准模型一经建立，许多实际博弈问题的建模与分析便如同雨后春笋一般出现，如重复博弈、扩展形式博弈（序贯博弈）等。在这一时期，博弈论开始展现出模拟人类在博弈中的复杂动态行为的强大威力。1970 年，为了分析生物的进化，约翰·梅纳德·史密斯引入了进化博弈论。直至今日，博弈论仍然是一个活跃的领域，许多学者都因博弈论相关的研究获得了诺贝尔经济学奖，如 Kenneth Arrow(1972), Gérard Debreu(1983), John Harsanyi & John Forbes Nash(1994), William Vickrey(1996), George Akerlof & Michael Spence(2001), Robert J. Aumann & Thomas C. Schelling(2005) 等。这也证明了博弈论在现代经济学中的重要性。

¹该定理的直觉是，有限确定完全信息博弈中，我们可以合将两个玩家任意时刻的所有可能状态构造为一个集合，并且在其中通过规则提供的合法行动定义状态之间的后继关系。那么，我们可以对集合中的所有状态分为三类：玩家 1 必胜的状态，玩家 2 必胜的状态和双人必然平局的状态。显然，玩家 1 在当前状态必胜，当且仅当存在一个该状态的后继状态，使得玩家 1 在该状态必胜；玩家 2 在当前状态必胜，当且仅当存在一个该状态的后继状态，使得玩家 2 在该状态必胜；当前状态必然平局，当且仅当该状态的所有后继状态都是平局。这样，理论上，通过规则判断出所有“终局”的状态类型，再通过该递推关系逐渐反向求出该集合中所有状态类型，我们就能完全确定出该博弈的一个必胜/平局策略。一个常见的这种博弈就是井字棋，由于它的状态非常有限，我们很容易证明先手玩家存在不败策略：占据中央位置。当然，Zermelo 定理只证明了这样策略的存在性，对于大部分游戏来说，因为可能的状态数量太多，几乎不可能给出这样的策略。

2 双人零和博弈

2.1 双人零和博弈的均衡

为了引入纳什均衡的概念，我们沿着 1928 年 Von Neumann 的推导过程，从博弈的一种非常特殊的情况，双人零和博弈出发。很多常见的游戏都可以理解为双人零和博弈，如石头剪刀布、象棋、物品分配等等。直观上，零和博弈是指在游戏中两个玩家的收益总和固定，因此一个玩家对自己的利益的提升总是伴随着对另一个玩家的损害。如果我们认为在石头剪刀布中，获胜的收益为 1，失败的收益为 0，平局的收益为 $1/2$ ，那么在任意可能结局中，两个玩家的收益总和均为 1；在物品分配中，如果我们认为玩家的收益和拿到的物品数量成正比，那么在物品总数固定时，玩家的收益之和也是固定的。以石头剪刀布为例，我们可以使用两个矩阵来表示两个玩家在双方策略下的结果：

玩家 1 的支付矩阵 A		玩家 2 的支付矩阵 B	
		玩家 2	
玩家 1	石头	剪刀	布
	1/2	1	0
	0	1/2	1
	1	0	1/2

注意到，对任意 $i, j \in \{1, 2, 3\}$, $A_{ij} + B_{ij} = 1$ ，因此这是一个零和博弈。这样，我们可以定义一般的双人博弈和双人零和博弈：

Definition 2.1. 双人博弈

一个有两个玩家的博弈称为双人博弈。如果玩家 1, 2 初始拥有 m 和 n 个策略，则玩家 1, 2 的支付矩阵可以分别用矩阵 $A, B \in \mathbb{R}^{m \times n}$ 定义。其中 A_{ij}, B_{ij} 表示玩家 1 选择第 i 个策略，玩家 2 选择第 j 个策略时各自的收益。特别的，如果存在常数 c ，使得对任意 i, j , $A_{ij} + B_{ij} = c$ ，则称该博弈为一个零和博弈。由于双人博弈的结果完全被两个玩家的支付矩阵 (A, B) 所确定，因此我们也用 (A, B) 表示对应的博弈。

但是，仅仅使用固定的策略不足以描述我们的实际行为。我们平时玩石头剪刀布的时候，往往不会固定使用某一个策略，而是在出手时随机选择某一个策略，以防止对手在重复几次后学习到自己的策略，并作出针对性的策略选择。在支付矩阵的框架下，我们需要在博弈中引入随机性：

Definition 2.2. 混合策略

在博弈中，玩家初始拥有的策略称为纯策略。假设玩家 1, 2 初始有 m 和 n 个纯策略。那么，玩家在决策时可以以某个概率分布使用自己的纯策略，这样的策略称为混合策略。我们一般用两个列向量 $x \in \Delta_m, y \in \Delta_n$ 表示此分布，其中 $\Delta_k \triangleq \{(x_1, \dots, x_k) | x_1, \dots, x_k \geq 0, x_1 + \dots + x_k = 1\}$ 为 $(k-1)$ -维单纯形。

对于策略对 (x, y) ，玩家 1 和 2 的期望收益分别为 $x^T A y = \sum_{i=1}^m \sum_{j=1}^n x_i A_{ij} y_j$ 和 $x^T B y = \sum_{i=1}^m \sum_{j=1}^n x_i B_{ij} y_j$ 。注意到，如果博弈是零和的，那么 $x^T A y + x^T B y \equiv c$ ，因此混合策略同样也具有零和性质。注意到，纯策略是一种特殊的混合策略。之后我们提及“策略”，如无说明，均指混合策略。

下面，我们分析玩家在零和博弈中的行为。我们假设博弈的所有玩家都是完全理性的，即他们总是试图最大化自己的收益，并且他们知道彼此是理性的；并且假设他们具有完全信息，即每个玩家都知道所有人的支付矩阵。

Definition 2.3. 最佳反应:

在双人博弈 (A, B) 中, 如果一个玩家知道对手的策略是 t , 那么他可以使得自己利益达到最大值的策略称为他针对策略 t 的最佳反应。对策略 t 的所有最佳反应的集合记为 $\text{br}(t)$ 。

注意: 一个策略可以有多个最佳反应。例如, 在石头剪刀布博弈中, 如果对手的策略是以各 $1/3$ 的概率出石头剪刀布, 那么此时自己的任何策略的期望收益都是 $1/2$, 所以此时任何策略都是其最佳反应。

假如玩家 1 知道玩家 2 将会使用策略 y^* , 根据理性原则, 他使用的策略 x^* 应当使得最大化自己的收益, 也就是 $x^* = \arg\max_{x \in \Delta_m} x^T A y^*$, 亦即, 使得 x^* 是 y^* 的最佳反应。同理, 如果玩家 2 知道玩家 1 将会使用策略 x^* , 那么他使用的策略 $y^* = \arg\max_{y \in \Delta_n} x^{*T} B y$, 亦即, y^* 是 x^* 的最佳反应。这样看来, 博弈的结果可能和玩家产生策略的先后顺序有关, 而且后手可以额外知道对手使用的策略信息, 从而做出最佳反应占据优势。但是事实上是这样吗?

我们假设玩家 1 先出手, 玩家 2 可以获得玩家 1 的策略后再出手。那么, 玩家 2 必定选择最大化自己的利益, 而根据完全理性原则, 玩家 1 知道玩家 2 是理性的, 那么他在出手时就应该选择一个使得玩家 2 可以占到最少便宜的策略。也就是说, 假如他使用策略 x , 那么他知道玩家 2 一定会选择 x 的一个最佳反应 y 。根据零和博弈的性质, 实际上就是使得 $x^T A y$ 最小的 y 。因此, 他应该选择能够最大化自己收益的策略 $x^* = \arg\max_{x \in \Delta_m} \min_{y \in \Delta_n} x^T A y$ 。此时, 他的期望收益是 $\max_x \min_y x^T A y$; 同理, 如果玩家 2 先手, 则玩家 1 的期望收益为 $\min_y \max_x x^T A y$ 。事实上, 下面的 Minimax 定理表明, 事实上它们是相等的!

Theorem 2.1. Minimax 定理

对于矩阵 $A \in \mathbb{R}^{m \times n}$, 有:

$$\max_x \min_y x^T A y = \min_y \max_x x^T A y$$

我们先给出几个引理:

Lemma 2.1. 凸集分离定理:

若凸集 $B \subset \mathbb{R}^d, x \notin B$, 则存在向量 $\alpha \in \mathbb{R}^d$ 和常数 a , 使得 $\alpha \cdot x = a < \alpha \cdot y, \forall y \in B$, 亦即, 凸集 B 和 x 被超平面 $\alpha \cdot t = a$ 分离。

Lemma 2.2. A 为 $m \times n$ 维矩阵, 其 m 个 n 维行向量记为 $a_i, i = 1, \dots, m$, 则下列二者之一成立:

- 0_n 位于 $\{a_1, \dots, a_m, e_1, \dots, e_n\}$ 的凸包²中。 $e_i, i = 1, \dots, n$ 是 n 个 n 维单位行向量, $0_n = (0, \dots, 0)$ 。
- 存在一个向量 $x_0 \in \Delta_n$, 使得 $a_i \cdot x_0 > 0, \forall i = 1 \dots m$ 。

证明. 如果情形 1 不成立, 则可对 0_n 和这些向量的凸包使用凸集分离定理, 即可得到情形 2. \square

证明. 对于矩阵 $A \in \mathbb{R}^{m \times n}$, 有: 我们先证明:

$$\max_x \min_y x^T A y \leq \min_y \max_x x^T A y$$

不妨设 $\max_x \min_y x^T A y = x_0^T A y_0, \min_y \max_x x^T A y = x_1^T A y_1$. 那么, 根据定义, 我们有: $\forall y, x_0^T A y_0 \leq x_0^T A y$, 以及 $\forall x, x_1^T A y_1 \geq x^T A y_1$.

特别的, 我们有: $x_0^T A y_0 \leq x_0^T A y_1 \leq x_1^T A y_1$, 命题得证!

下面考虑上面引理 2 的两种情形:

²向量 x_1, \dots, x_k 的凸包 $\text{conv}(x_1, \dots, x_k) \triangleq \{x | x = \sum_{i=1}^k a_i x_i + \dots + a_k x_k, (a_1, \dots, a_k) \in \Delta_k\}$, 容易验证这是一个凸集。

情形 1: 根据凸包的定义, 存在和为 1 的非负实数 s_1, \dots, s_{n+m} , 使得 $\sum_{j=1}^m a_{ij}s_j + s_{m+i} = 0, \forall i = 1, \dots, n$. 注意到, s_1, \dots, s_n 不全为 0, 从而可以定义 m 维列向量 \bar{y} : $\bar{y}_i = s_i / \sum_{i=1}^n s_i$, 则 $y \in \Delta_m$, 且 $\sum_{j=1}^n a_{ij}\bar{y}_j = -s_{n+i} / \sum_{i=1}^n s_i \leq 0, \forall i$, 从而, $\min_y \max_x x^T A y = \min_y \max_{i=1}^m \{\sum_{j=1}^n a_{ij}y_j\} \leq \max_{i=1}^m \{\sum_{j=1}^n a_{ij}\bar{y}_j\} \leq 0$.

情形 2: 同理, 我们有: $\max_x \min_y x^T A y \geq \min_y x_0^T A y = \min_{i=1}^n \{a_i \cdot x_0\} \geq 0$.

综上, 我们证明了对于 0, 要么 $\max_x \min_y x^T A y \leq \min_y \max_x x^T A y \leq 0$, 要么 $0 \leq \max_x \min_y x^T A y \leq \min_y \max_x x^T A y$, 也就是两个式子的值必然在其同侧。由于该结论对任意的矩阵 A 成立, 对任意的常数 c , 令 $B = A - c$, 则说明两个式子的值必然也在 c 同侧。从而, 两个式子的值必须相等, 命题得证! \square

Definition 2.4. 双人零和博弈的均衡

在零和博弈中, 我们将求解 $V = \max_x \min_y x^T A y$ 所得到的策略 (x^*, y^*) 称为一个均衡。因此, 任意一个零和博弈必定存在一个均衡。并且, 任意均衡策略中玩家 1 和玩家 2 的收益分别由 V 和 $c - V$ 给出。 V 称为该博弈的价值。

一个策略 (x^*, y^*) 是均衡的当且仅当对任意的策略 x, y , $x^T A y^* \leq x^{*T} A y^*, x^{*T} A y \geq x^{*T} A y^*$, 也就是说, 两个玩家都无法通过单方面更改自己的策略提升自己的收益。

根据 Minimax 定理, 双人零和博弈中理性玩家的收益实际上与先后手无关, 理性玩家的期望收益为一个定值。从某种意义上来说, 这其实相当于两个玩家达成了某种“默契”, 通过各自理性的制约, 最终走向了一个均衡的策略。我们通过一个例子来更深入地理解这个概念。

Example 2.1. 对于石头剪刀布博弈:

- (1) (石头, 剪刀) 不是一个均衡。(之后均默认一对策略 (a, b) 中 a 表示玩家 1 的策略, b 表示玩家 2 的策略)
- (2) 该博弈不存在纯策略均衡。
- (2) 该博弈的唯一一个混合策略均衡是 (各以概率 1/3 出石头剪刀布, 各以概率 1/3 出石头剪刀布)。

证明. (1) 对于玩家 2 来说, 他可以通过将策略从“剪刀”更改为“布”将自己的收益从 0 提升到 1, 因此这对策略不是均衡。

(2) 在该博弈中, 策略的制胜关系为: “石头 > 剪刀 > 布 > 石头”。因此, 对于该博弈的任意一对纯策略 (a, b) , 玩家 1 会偏移使得 a 恰好胜过 b 的策略, 而玩家 2 会偏移使得 b 恰好胜过 a 的策略。但是制胜关系的循环周期为 3, 因此这两个关系不能同时满足, 从而不存在纯策略均衡。

(3) 我们直接计算 $\max_x \min_y x^T A y$:

$$x^T A y = \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix} \begin{pmatrix} 1/2 & 1 & 0 \\ 0 & 1/2 & 1 \\ 1 & 0 & 1/2 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \frac{1}{2}(x_1 y_1 + x_2 y_2 + x_3 y_3) + (x_1 y_2 + x_2 y_3 + x_3 y_1).$$

$$\begin{aligned} \max_x \min_y x^T A y &= \max_x \left(\min_y \left(\frac{1}{2}(x_1 + x_3)y_1 + \frac{1}{2}(x_2 + x_1)y_2 + \frac{1}{2}(x_3 + x_2)y_3 \right) \right) \\ &= \max_x \min \left\{ \frac{1}{2}(x_1 + x_3), \frac{1}{2}(x_2 + x_1), \frac{1}{2}(x_3 + x_2) \right\} = 1/3 \end{aligned}$$

倒数第二步是因为: 对任意参数 $a = (a_1, \dots, a_n)$ 和向量 $y = (y_1, \dots, y_n) \in \Delta_n$, 根据 y_i 非负且和为 1, 有: $a_1 y_1 + \dots + a_n y_n \geq \min\{a_1, \dots, a_n\}(y_1 + \dots + y_n) = \min\{a_1, \dots, a_n\}$; 并且假设 $j = \operatorname{argmin}_i \{a_i\}$, 则取 $y = e_j$ 时, 取得等号。因此, $\min_{y \in \Delta_n} \{\sum_{i=1}^n a_i y_i\} = \min\{a_1, \dots, a_n\}$.

最后一步是因为 $\min\{\frac{1}{2}(x_1 + x_3), \frac{1}{2}(x_2 + x_1), \frac{1}{2}(x_3 + x_2)\} \leq \frac{1}{3}(x_1 + x_2 + x_3) = 1/3$ ，且等号取得当且仅当 $x_1 = x_2 = x_3 = 1/3$ 。从而，玩家 1 的均衡策略必为以 1/3 的概率选取各个策略。同理，玩家 2 的均衡策略也为以 1/3 的概率选取各个策略。根据取等条件，这是该博弈的唯一一个均衡。□

Remark 2.1. 常见疑问：

在石头剪刀布中看起来后手具有必胜策略，例如，如果通过晚出观察到对手的出手后再出手，那么似乎是必胜的。

解释：这个问题实际上违反了博弈的定义。在博弈的决策过程中，两个人必须先给出自己的策略，然后再获得博弈的结果。策略的不确定性出现在博弈的结果出现之前。由于石头剪刀布中，作出决策的人和执行结果的人是同一个，所以可能会引发混淆。如果我们将过程改为：两个人各自告诉计算机一个策略，并且由计算机执行结果足够多次使得两个各自获得期望收益，就不会混淆了。

2.2 双人零和博弈均衡的计算

在提出了零和博弈的均衡概念以及存在性之后，一个很自然的问题是，给定玩家的支付矩阵，能否高效地计算该矩阵的一个均衡？这一部分，第一节，我们将介绍利用线性规划的多项式时间精确算法；第二节，我们将引入预测理论的一些概念，并给出迭代的高效近似算法。

2.2.1 线性规划算法

一个比较自然的想法是从定义出发。我们之前给出了判定策略 (x^*, y^*) 是均衡的充要条件：对任意的策略 x, y ， $x^T A y^* \leq V, x^{*T} A y \geq V$ ，其中 V 为该博弈的价值。这说明，玩家 1 的某个策略 x^* 是均衡策略当且仅当对玩家 2 的任意策略 y ， $\sum_{i=1}^m x_i^* A_{ij} \geq V$ 。这是因为，一方面，这使得 $x^{*T} A y \geq V, \forall y$ ，另一方面，根据 V 的定义， $V = \max_x \min_y x^T A y \geq \min_y x^{*T} A y$ ，而上面的分析表明玩家 1 可以使用策略 x^* 保证自己得到的收益不小于 V ，因此，总有 $\min_y x^{*T} A y \geq V$ 。因此， $V = \min_y x^{*T} A y$ 。如果取 $y^* = \operatorname{argmin}_y x^{*T} A y$ ，那么 $x^* A y^* \geq V = \max_x x^T A y^* \geq x^T A y^*$ ，对任意策略 x 成立。

根据这个命题，我们可以提出一个对应的线性规划问题：

$$\begin{aligned} \forall 1 \leq j \leq n, \quad \sum_{i=1}^m x_i A_{ij} - v &\geq 0 \\ \forall 1 \leq i \leq m \quad x_i &\geq 0 \\ \sum_{i=1}^m x_i &= 1 \\ \text{Maximize} \quad &v \end{aligned} \tag{2.1}$$

注意，这里巧妙地把规划问题的目标函数设定为了 v 。这是因为，一方面上面的推导过程表明，当 $v = V$ 时，该线性规划程序有解 x^* ，且必然是一个均衡策略；另一方面， V 的定义表明，该问题在 $(V, +\infty)$ 上无解，否则我们可以得到一个策略 x^* ，使得对任意 y ， $x^{*T} A y \geq v > V$ 。但是 $V = \min_y \max_x x^T A y \geq \min_y x^{*T} A y \geq v > V$ ，矛盾！

使用以上程序求出 x^* 和 V 后，根据充要条件， y^* 是对应的均衡策略当且仅当对任意策略 x ， $x^T A y^* \leq V$ ，因此我们同样可以直接给出关于 y 的线性规划：

$$\begin{aligned} \forall 1 \leq i \leq m, \quad \sum_{j=1}^n y_j A_{ji} - V &\geq 0 \\ \forall 1 \leq j \leq n \quad y_j &\geq 0 \\ \sum_{j=1}^n y_j &= 1 \\ \text{Maximize} \quad &y_1 \end{aligned}$$

这里的目标函数是随意选取的。

这样，由于线性规划的求解具有多项式时间复杂度，我们事实上就得到了求解一对零和博弈均衡策略的多项式时间算法。

Exercise 2.1.

1. 证明：双人零和博弈的均衡具有可交换性：

假设零和博弈 (A, B) 的纳什均衡为 (γ_1, γ_2) 和 (σ_1, σ_2) ，那么 $(\gamma_1, \sigma_2), (\sigma_1, \gamma_2)$ 也是纳什均衡，且这四个均衡中玩家 1 的收益完全相等。

2. 写出线性规划算法 2.1 的对偶规划，并证明：该规划给出了玩家 2 的一个均衡策略。

Remark 2.2.

1. 根据习题 1 的结论，将线性规划算法 2.1 和习题 2 的对偶算法的解组合起来也是一个均衡策略。
2. 这两个对偶算法事实上就是在优化中求解问题 $\max_x \min_y x^T A y$ 和 $\min_y \max_x x^T A y$ 的标准程序。

2.2.2 预测论与虚拟博弈

除了线性规划的思想，另一个比较自然的想法是模拟现实中两个玩家做重复博弈，最终相互学习和妥协，逐渐走向一个均衡的过程。由于不同的模拟可以导出不同的算法，所以这是一类统称“虚拟博弈” (Fictitious Play) 的算法。它们模拟了“学习”一个均衡的过程，而且效率很高，所以在信息论、在线算法 (Online Algorithm)、机器学习的生成对抗网络 (GAN) 等领域中有广泛的应用和讨论。

虚拟博弈的思想来源于预测论 (Prediction Theory)。预测论试图通过对某个序列的结果进行预测。亦即：对于在时刻 $t = 1, 2, \dots$ 不断生成的序列 $\{y_t\}_{t=1}^n$ ，在每个时刻 t ，预测者要基于之前的观察 $\{y_1, \dots, y_{t-1}\}$ 和某些信息来预测 $\{y_t\}$ ，并试图达到最大的正确率。

预测论听起来很像统计学。但是，统计学往往假设 $\{y_t\}$ 产生自一个固定的分布族 F ，并且希望通过若干统计量（如样本期望，样本方差等）来确定数据具体来源于哪个分布。与其不同，预测论不对序列 $\{y_t\}$ 做任何假设，而是假设存在若干个“专家”，他们在每个时刻会提出各自的建议。预测者的目的是基于这些建议做出最正确的预测。我们通过下面的例子展示预测论的一些基本概念和思想：

Example 2.2. 假设预测者需要预测一个序列 $\{y_t\}_{t=1}^\infty, y_t \in \{0, 1\}$ 。从时刻 1 开始，在每个时刻 t ， N 个专家会分别给他一个建议 $f_{t,i}, i = 1, \dots, N$ ，他需要基于这些建议给出一个预测 $\hat{p}_t \in \{0, 1\}$ 。做出预测后，他会被告知 y_t 的真实值。他的目的是给出某个决策程序，使得自己犯错的总次数尽可能小。考虑下面的两种情况：

- (1) 若已知至少有一名专家的预测是一定正确的，给出一个犯错总次数有限的决策程序。
- (2) 若已知存在一名专家，在前 m 次的预测中保证至多犯 km 次错， k 为某个常数，给出一个在前 m 次犯错总比例仅与 k 相关的决策程序。

证明. (1) 一个很自然的想法是维护一个“可信专家集”，并且决策仅依赖他们的建议，例如，在他们的建议里面随机选择一个采纳。一开始，这个集合包含所有的专家，但是某个专家一旦犯错就将他移出这个集合。这样，就能保证这位预测一定正确的专家总是留在可信集内。采用这种策略，可以得到一个自然的错误上界 $N - 1$ ：因为预测者每次犯错就意味着至少有一个专家变得不可信，而一开始 N 个专家都可信，最终至少还剩一个专家，所以犯错次数至多为 $N - 1$ 。

但是，事实上我们可以对决策过程略微改进一下，大幅降低犯错次数的上界。我们考虑“少数服从多数”的原则，每次从可信专家集的建议中采纳出现次数较多的建议 0/1。这样，一旦犯错，那么至少有一半的专家都变得不可信。从而，犯错的次数至多是 $\lfloor \log_2 N \rfloor$ 次。

(2) 由于我们现在无法保证一定有一个专家是正确的了，所以不能再采用之前一旦错误就不信任的方法。然而，我们注意到其实信任与不信任的本质区别在于，不信任的专家的建议在决策中的权重为 0；因此我们自然地考虑对 N 个专家各自维护一个权重 w_i ，并且专家若犯错一次，则将其权重更新为 βw_i ，亦即之后减少他的策略的权重。我们注意到之前的方法正式 $\beta = 0$ 的特殊情形。

这样，我们构造的策略是：在第一步时对所有专家赋予一个权重 $w_i = 1$ ，之后每一轮将上一次预测错误的专家的权重更新为 βw_i 。在做决策时，我们比较所有给出预测 1 的专家的总权重和给出所有预测为 0 的专家的总权重，哪个总权重更高，我们就使用哪个预测。这样，我们可以保证，我们一旦犯错，那么至少有占据一半权重的专家犯错。

因此，假设我们在前 m 次中总共犯了 m^* 次错，那么，每次犯错会导致总权重从 W 至少减少到 $W/2 + \beta W/2$ ，而一开始的总权重为 N ，那么最终的总权重至多为：

$$\left(\frac{1+\beta}{2}\right)^{m^*} N$$

但是，根据条件，至少有一名专家，他至多会犯 km 次错，故权重至多会被减少 km 次，故他的权重至少是 β^{km} 。综合以上两个式子，有：

$$\begin{aligned} \beta^{km} &\leq \left(\frac{1+\beta}{2}\right)^{m^*} N \\ \text{解得: } \frac{m^*}{m} &\leq \left\lfloor \frac{\log_2 N}{m \log_2 \frac{1+\beta}{2}} + k \frac{\log_2(1/\beta)}{\log_2 \frac{1+\beta}{2}} \right\rfloor \end{aligned}$$

从而得到了一个错误比例仅与 k 有关的决策。特别的，当决策总次数 $m \rightarrow \infty$ 时，该比例趋近于 $k \frac{\log_2(1/\beta)}{\log_2(2/(1+\beta))}$ ， $\beta \rightarrow 1$ 时取到最小值 $2k$ 。而且，即使是 m 比较小的情况，我们的决策也能保证受到专家总数量 N 的影响很小。 \square

从这个例子里，我们可以看到很多决策论的基本思想，例如，为不同的建议加权，总是选择能够引入最多信息的决策等等。下面，我们给出专家建议决策的一个正式的数学模型。这个模型和上面例子的区别主要是引入了损失函数。因为，在决策空间比 $\{0, 1\}$ 更大的情况下，不再能够使用“对/错”简单地描述决策的结果。

Definition 2.5. 专家建议决策：

参数：决策空间 D ， D 是凸向量空间，结果空间 Y ，损失函数 $l: D \times Y \rightarrow \mathbb{R}$ ，专家集合 E 。

对于每一轮 $t = 1, 2, \dots$:

- (1) 专家们提供建议 $\{f_{e,t} \in D : i \in E\}$.
- (2) 预测者选择预测 $\hat{p}_t \in D$.
- (3) 环境揭示结果 $y_t \in Y$.
- (4) 决策者遭受损失 $l(\hat{p}_t, y_t)$ ，每个专家 e 遭受损失 $l(f_{e,t}, y_t)$.

从而可以定义：

玩家的累积损失： $\hat{L}_n \triangleq \sum_{t=1}^n l(\hat{p}_t, y_t)$.

专家 e 的累积损失： $L_{e,n} \triangleq \sum_{t=1}^n l(f_{e,t}, y_t)$

预测者的目标：

最小化自己的累积损失和每个专家 e 的累积损失之差，即没有遵守专家的建议导致的损失，称

为后悔 (regret), 即:

$$\text{Minimize: } R_{e,n} \triangleq \hat{L}_n - L_{e,n}$$

定义第 t 轮中对专家 e 的后悔为 $r_{e,t} = l(\hat{p}_t, y_t) - l(f_{e,t}, y_t)$.

那么, 将上面例题的思想略作推广, 就可以得到在关于专家建议决策问题的一个决策机制:

我们只考虑专家总数有限的情况, 因此不妨设共 N 个专家, 编号为 $1, 2, \dots, N$ 。在时刻 t , 假设这些专家的后悔分别为 $R_{1,t}, \dots, R_{N,t}$ 。根据定义, 一个专家的后悔值越高, 说明他的建议越本该被采纳。仿照上面的思想, 我们希望构造关于后悔值的非负权重函数 w , 使得后悔值越高, 权重就越大。这样, 在第 t 轮时, 这些专家的权重分别为 $(w(R_{1,t}, \dots, R_{N,t}))$, 因此预测者在这一轮的预测应该为:

$$\hat{p}_t = \frac{\sum_{i=1}^N w(R_{i,t-1}) f_{i,t}}{\sum_{j=1}^N w(R_{j,t-1})}$$

Lemma 2.3. 如果损失函数 l 对于第一个参数是凸的, 那么:

$$\sup_{y_t \in Y} \sum_{i=1}^N r_{i,t} w(R_{i,t-1}) \leq 0$$

证明. 使用 Jensen 不等式, 有:

$$\begin{aligned} l(\hat{p}_t, y) &= l\left(\frac{\sum_{i=1}^N w(R_{i,t-1}) f_{i,t}}{\sum_{j=1}^N w(R_{j,t-1})}, y\right) \leq \frac{\sum_{i=1}^N w(R_{i,t-1}) l(f_{i,t}, y)}{\sum_{j=1}^N w(R_{j,t-1})} \\ \text{即: } \sum_{i=1}^N w(R_{i,t-1}) (l(\hat{p}_t, y) - l(f_{i,t}, y)) &= \sum_{i=1}^N w(R_{i,t-1}) r_{i,t} \leq 0 \end{aligned}$$

□

这个引理的给出了这种加权结构的一个优良性质, 即: 不论序列中下一个出现的元素是什么, 采用加权方法给出的新策略都能保证这一轮中所有的后悔值的加权和的上界为 0。而且, 更重要的是, 这种形式和方向导数很接近。我们下面说明: 通过构造特定势函数, 那么这种结构可以进一步转换为更加有利于分析的形式。

将时刻 t 对所有专家的后悔值记为一个向量 $r_t = (r_{1,t}, \dots, r_{N,t})$, 该向量在时间 t 上的累积为 $R_t = \sum_{j=1}^t r_j$, 那么, 定义关于 n 维向量 u 的势函数 $\Phi(u) = \psi\left(\sum_{i=1}^N \phi(u_i)\right)$, 如果我们取 $w = \phi'$, 则:

$$\nabla \Phi(u) = ((\psi'(\sum_{i=1}^N \phi(u_i)) \phi'(u_1), \dots, (\psi'(\sum_{i=1}^N \phi(u_i)) \phi'(u_N))$$

从而, 上面引理的等价形式为: $\sup_{y_t \in Y} \sum_{i=1}^N r_t \cdot \nabla \Phi(R_{t-1}) \leq 0$. (Blackwell 条件)

为了保证现在的权重函数 ϕ' 满足一开始的三个条件, 即: 非负、单调递增, 我们需要 ϕ 是非负单调递增的凸函数。

根据方向导数小于等于 0 的条件, 利用泰勒展开, 我们立即得到:

$$\begin{aligned} \Phi(\mathbf{R}_t) &= \Phi(\mathbf{R}_{t-1} + \mathbf{r}_t) \\ &= \Phi(\mathbf{R}_{t-1}) + \nabla \Phi(\mathbf{R}_{t-1}) \cdot \mathbf{r}_t + \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \frac{\partial^2 \Phi}{\partial u_i \partial u_j} \Big|_{\xi} r_{i,t} r_{j,t} \leq \Phi(\mathbf{R}_{t-1}) + \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \frac{\partial^2 \Phi}{\partial u_i \partial u_j} \Big|_{\xi} r_{i,t} r_{j,t} \end{aligned}$$

我们对余项进行估计：

$$\begin{aligned}
& \sum_{i=1}^N \sum_{j=1}^N \frac{\partial^2 \Phi}{\partial u_i \partial u_j} \Big|_{\xi} r_{i,t} r_{j,t} \\
&= \psi'' \left(\sum_{i=1}^N \phi(\xi_i) \right) \sum_{i=1}^N \sum_{j=1}^N \phi'(\xi_i) \phi'(\xi_j) r_{i,t} r_{j,t} \\
&+ \psi' \left(\sum_{i=1}^N \phi(\xi_i) \right) \sum_{i=1}^N \phi''(\xi_i) r_{i,t}^2 \\
&= \psi'' \left(\sum_{i=1}^N \phi(\xi_i) \right) \left(\sum_{i=1}^N \phi'(\xi_i) r_{i,t} \right)^2 + \psi' \left(\sum_{i=1}^N \phi(\xi_i) \right) \sum_{i=1}^N \phi''(\xi_i) r_{i,t}^2 \\
&\leq \psi' \left(\sum_{i=1}^N \phi(\xi_i) \right) \sum_{i=1}^N \phi''(\xi_i) r_{i,t}^2 \quad (\text{假设 } \psi \text{ 是凹函数})
\end{aligned}$$

ldc: [这个计算过程没什么技巧，也许可以留作习题?] 迭代地使用这个结果，即可得到定理：

Theorem 2.2. 若 ψ 为二阶可导凹函数， ϕ 为非负单调递增的二阶可导凸函数，则其定义的函数 $\Phi(u) = \psi \left(\sum_{i=1}^N \phi(u_i) \right)$ 使得：

$$\begin{aligned}
\Phi(R_t) &\leq \Phi(0) + \frac{1}{2} \sum_{j=1}^t C(r_j) \\
\text{其中, } C(\mathbf{r}_t) &= \sup_{\mathbf{u} \in \mathbb{R}^N} \psi' \left(\sum_{i=1}^N \phi(u_i) \right) \sum_{i=1}^N \phi''(u_i) r_{i,t}^2
\end{aligned}$$

使用这个定理，我们可以对许多不同类型的加权函数进行分析，如使用 p-范数加权，指数加权等。这里我们给出指数加权的分析。

注意到： $\ln(x)/\lambda$ 是二阶可导凹函数，而 $e^\lambda x$ 是非负单调递增的二阶可导凸函数，我们可以给出一个满足条件的势函数

$$\Phi_\eta(\mathbf{u}) = \frac{1}{\eta} \ln \left(\sum_{i=1}^N e^{\eta u_i} \right)$$

从而，有 $w_{i,t-1} = \nabla \Phi_\eta(\mathbf{R}_{t-1})_i = \frac{e^{\eta R_{i,t-1}}}{\sum_{j=1}^N e^{\eta R_{j,t-1}}}$ ，预测者的预测为：

$$\hat{p}_t = \frac{\sum_{i=1}^N \exp \left(\eta \left(\hat{L}_{t-1} - L_{i,t-1} \right) \right) f_{i,t}}{\sum_{j=1}^N \exp \left(\eta \left(\hat{L}_{t-1} - L_{j,t-1} \right) \right)} = \frac{\sum_{i=1}^N e^{-\eta L_{i,t-1}} f_{i,t}}{\sum_{j=1}^N e^{-\eta L_{j,t-1}}}$$

从而，权重函数可以化简为 $w_{i,t-1} = e^{-\eta L_{i,t-1}}$ 。

这表明指数加权方法有着非常好的性质。一方面，预测者的预测仅与各个专家的总后悔值有关，和预测者自己的后悔值无关；另一方面，两轮之间权重的变化更新非常容易，因为根据定义， $L_{i,t} = L_{i,t-1} + r_{i,t}$ ，因此，在新的轮次中，我们只需要将每个专家的权重更新为原来权重和 $e^{-\eta r_{i,t}}$ 的乘积。

下面我们用上面的估计定理对玩家的累计后悔值给出一个上界：

Corollary 2.1. 指数加权上界：

假设损失函数关于第一个参数凸，并且取值范围为 $[0, 1]$ ，则对任意 n 和 $\eta > 0$ ：

$$\hat{L}_n - \min_{i=1, \dots, N} L_{i,n} \leq \frac{\ln N}{\eta} + \frac{n\eta}{2}$$

证明. 代入 $\phi(x) = e^{\eta x}, \psi(x) = (1/\eta) \ln x$, 有:

$$\psi' \left(\sum_{i=1}^N \phi(u_i) \right) \sum_{i=1}^N \phi''(u_i) r_{i,t}^2 \leq \eta \max_{i=1, \dots, N} r_{i,t}^2 \leq \eta.$$

从而, $C(\mathbf{r}_t) \leq \eta$, 则

$$\max_{i=1, \dots, N} R_{i,n} \leq \Phi_\eta(\mathbf{R}_n) \leq \frac{\ln N}{\eta} + \frac{n\eta}{2}$$

其中第一个不等式是由于琴生不等式: $\phi_\eta(R_n) = \frac{\ln(\sum_{i=1}^N e^{\eta R_{i,n}})}{\eta} \geq \frac{\sum_{i=1}^N \eta R_{i,n}}{\eta}$ □

那么, 这个结论在虚拟博弈中的对应是什么呢? 我们先给出下面的算法, 然后讨论该算法和上面介绍的决策论的相似性。

光滑虚拟博弈

给定参数族 $\{\eta_0^t : t = 1, 2, 3, \dots\}$

对 t 从 1 开始迭代:

1. 玩家 1 选择策略 x^t : 使用第 i 个策略的概率与 $e^{\eta_0^t \pi_i^t}$ 成正比, 其中 π_i^t 表示策略 i 在玩家 2 使用混合策略 $\hat{y}^{t-1} = \frac{1}{t-1} \sum_{s=1}^{t-1} y^s$ 的期望收益
2. 玩家 2 选择策略 y^t : 使用第 j 个策略的概率与 $e^{\eta_0^t \pi_j^t}$ 成正比, 其中 π_j^t 表示策略 j 在玩家 1 使用混合策略 $\hat{x}^{t-1} = \frac{1}{t-1} \sum_{s=1}^{t-1} x^s$ 的期望收益
3. 两个玩家分别收到更新后的 x^t, y^t

Theorem 2.3. 设零和博弈为 $(A, -A)$, $A \in \mathbb{R}^{n \times n}$, 且 A 的所有分量恰好在 $[0, 1]$ 内, 则对任意 $\epsilon > 0$, 以上算法若选取 $\eta_0^t = \sqrt{2t \ln n}$, 则可以在 $T = O(\log(n)/\epsilon^2)$ 的时间内得到一个 ϵ -近似均衡。一个策略对是 ϵ -近似的, 若其中任一玩家的最佳反应和当前策略的期望收益差不超过 ϵ 。

证明. 我们注意到, 在该虚拟博弈中, 两个玩家的交互行为实际上对每个玩家来说, 都相当于一个决策问题: 在这里, 专家就是玩家的各个纯策略, 而玩家对各个策略的权重由该策略应对对方的累积收益决定; 每一轮开始, 两个玩家要各自选择一个混合策略, 而在结束时了解到本轮对方选择的策略信息, 并收获当前轮次的收益; 玩家要参考之前的收益, 预测这一轮使得自己收益最大的策略, 也就是对方将要使用的策略 y_t 的最佳反应。

对玩家 1, 该决策问题的参数为:

$N = n$: 每个纯策略对应一个专家;

$f_{i,t} = i$: 每个专家的建议就是该纯策略;

x_t, y_t : 第 t 轮中玩家 1 和对手的决策;

$\hat{y}_t = \frac{\sum_{j=1}^t y_j}{t}$: 前 t 轮中对手的平均策略;

$l(i, y) = -e_i^T A y$: 损失函数为当前纯策略在应对对手策略 y 的负效用, 因此, 一个纯策略带来的效用越高, 损失就越小。 l 关于第一个参数是线性的, 所以必然是凸函数。

$L_{i,t} = \sum_{j=1}^t l(i, y_j) = -e_i^T A \left(\sum_{j=1}^t y_j \right)$: 第 i 个策略的累积损失为前 t 轮的效用和的相反数。

$w_{i,t} = e^{-\eta L_{i,t}} = e^{\eta e_i^T A (\sum_{j=1}^t y_j)} e^{t\eta e_i^T A x_t}$.

如果取参数 $\eta = \frac{\eta_0}{t}$, 则我们构造出来的决策问题和光滑虚拟博弈完全相同。代入 $\eta_0 = \eta_0^t = \sqrt{2t \ln n}$, 有 $\eta = \eta_0^t = \sqrt{2 \ln n / t}$ 。代入推论 2.1, 有: $\hat{L}_t - \min_{i=1, \dots, n} L_{i,t} \leq \sqrt{2t \ln n}$ 。

我们注意到，根据线性性，纯策略 $1, \dots, n$ 中必然有应对 \hat{x}_t 的最佳策略，从而 $\min_{i=1, \dots, n} L_{i,t}$ 的值等于前 t 轮可能采取的最佳策略的后悔值。因此， $\hat{L}_t - \min_{i=1, \dots, n} L_{i,t} = R_t$ 。从而，如果将博弈进行 t 轮，则前 t 轮博弈的平均后悔值小于等于 $\sqrt{2 \ln n / t}$ 。我们立即得到一个推论：对于任意 $\epsilon > 0$ ，如果我们希望前 t 轮的平均后悔值小于 ϵ ，我们只需要取 $t = 4 \ln n / \epsilon^2$ 。

最后，我们证明，最终得到的 (\hat{x}_t, \hat{y}_t) 即为所求的 ϵ -近似均衡。

定义玩家 1 的平均收益为 $v = \frac{1}{t} \sum_{j=1}^t x_j^T A y_j$ 。那么，根据上面关于后悔的结论（对两个玩家分别使用），有：

$$\begin{aligned} v &\geq \left(\max_x \frac{1}{t} \sum_{t=1}^t x^T A y_t \right) - \frac{\epsilon}{2} = \left(\max_x x^T A \hat{y}_t \right) - \frac{\epsilon}{2} \\ v &\leq \left(\min_y \frac{1}{t} \sum_{t=1}^t (x^t)^T A y \right) + \frac{\epsilon}{2} = \left(\min_y \hat{x}_t^T A y \right) + \frac{\epsilon}{2} \end{aligned}$$

因此，对于策略 (\hat{x}_t, \hat{y}_t) 的任何偏离，例如 (x, \hat{y}_t) ，根据式 1，有 $x^T A \hat{y}_t \leq v + \frac{\epsilon}{2}$ ，故其比起 \hat{x} 至多能获得 $\frac{\epsilon}{2}$ 的收益；对于 y 的偏离同理。综上，命题得证！ \square

Exercise 2.2.

1. ldc: [上面的计算？]

2. 利用定理 2.2 证明以下结论：如果损失函数对一个参数凸，且取值范围为 $[0, 1]$ ，选取势函数

$$\Phi(u) = \|u_+\|_p^2 \triangleq \left(\sum_{u_i > 0, 1 \leq i \leq N} (u_i)^p \right)^{2/p}$$

，则

$$\hat{L}_n - \min_i L_{i,n} \leq \sqrt{n(p-1)N^{2/p}}$$

3. 已知计算 $n \times n$ 矩阵和 n 向量乘法的时间复杂度为 $O(n^2)$ ，证明：光滑虚拟博弈算法的时间复杂度为 $O(n^2 \log n / \epsilon^2)$ 。

4. 为了展示势函数的作用，定理 2.2 的证明思路和例题并不相同。事实上，用例题的证明思路，定义 $W_t = \sum_{i=1}^N w_{i,t}$ ，利用 $\frac{W_n}{W_0}$ 的上下界，也可以证明定理 2.2。按照如下提示，给出证明：

(1) 证明：

$$\ln \frac{W_n}{W_0} \geq -\eta \min_{i=1, \dots, N} L_{i,n} - \ln N.$$

(2) 利用 Hoeffding 不等式：对任意随机变量 X 取值范围 $[a, b]$ ， $\ln \mathbb{E}[e^{sX}] \leq s\mathbb{E}X + \frac{s^2(b-a)^2}{8}$ ，证明 $\ln \frac{W_t}{W_{t-1}} \leq -\eta \ell(\hat{p}_t, y_t) + \frac{\eta^2}{8}$

(3) 结合以上两式，可得： $\hat{L}_n - \min_{i=1, \dots, N} L_{i,n} \leq \frac{\ln N}{\eta} + \frac{n\eta}{8}$

Remark 2.3. 由于迭代方式有很多种，虚拟博弈也有很多种类。例如，经典虚拟博弈算法为：在第 t 轮，两个玩家分别给出对方策略在前 t 轮的平均的最佳反应。这个算法看起来非常简单，然而，由于一个策略可以有多个最佳反应，如何选择最佳反应对于算法的结果影响很大。2014 年，[16] 证明了在对抗情况下（最坏地选取最佳反应）该算法需要指数轮才能收敛到 ϵ -均衡；而一般情况下经典虚拟博弈是否在多项式轮次内收敛的问题至今仍然未被解决。

3 纳什均衡的定义与存在性

上文中的“均衡”概念很容易推广到一般的博弈中。1951 年，纳什推广了上面的概念，定义了纳什均衡，并证明了任意博弈都存在纳什均衡。

Definition 3.1. 纳什均衡：

对于一般的双人博弈，假设两个玩家分别有 m 和 n 个策略，并且支付矩阵分别为 A 和 B ，我们称一对混合策略 (x^*, y^*) 为纳什均衡，若其满足：

$$\begin{aligned} x^T A y^* &\leq x^{*T} A y^*, \forall x \in \Delta_m \\ x^{*T} B y &\leq x^{*T} B y^*, \forall y \in \Delta_n \end{aligned}$$

我们之后同样用 (A, B) 直接指代对应的博弈。

容易验证， (x^*, y^*) 是纳什均衡当且仅当 x^* 是 y^* 的最佳反应， y^* 也是 x^* 的最佳反应。

上一节我们讨论的双人零和博弈的均衡就是零和博弈的纳什均衡。直观上来说，纳什均衡可以理解为两个玩家之间的一个特别的约定。在这个约定中，双方都没有动机在选取策略时偏离约定，因为假设对方遵守约定，那么遵循理性原则，自己也应该遵守约定。

这个概念在哲学上的重要性在于，它表明，即使是完全理性（自私）的玩家之间，也必然能达成某种不会背叛的合作关系。

下面，我们证明纳什均衡的存在性。由于纳什的初始证明比较复杂，这里我们展示的是 Geanakoplos 等人 [20] 用 Brouwer 不动点定理给出的一个非常自然的证明。³

Theorem 3.1. Brouwer 不动点定理：

如果 C 是 \mathbb{R}^d 的一个紧致凸子集，函数 $f : C \rightarrow C$ 连续，则其必然存在一个不动点 $x \in C$ 使得 $f(x) = x$ 。

该定理非常经典，而且证明比较复杂（参考尤承业《基础拓扑学讲义》第八章：映射度与不动点），与我们的主题也关系不大，我们在此省略。

现在，我们考虑如何把纳什均衡转化为不动点问题：

Theorem 3.2. 纳什均衡的存在性：任意一个双人博弈 (A, B) 都存在一个纳什均衡 (x^*, y^*) 。

证明. 定义 $C = \Delta_m \times \Delta_n$ 。由于任意两个概率分布的凸组合仍然是一个概率分布，并且 C 有界，因此 C 是 \mathbb{R}^{n+m} 的一个紧致凸子集，满足使用 Brouwer 不动点定理的条件。

我们知道，纳什均衡的本质是两个人互相达到了最佳反应。如果我们能定义函数 $h : C \rightarrow C$ ，其直接将策略对映射为其最佳反应对，亦即：

$$\begin{aligned} h_1(x_1, y_1) &= \text{br}(y_1) = \underset{x'_1 \in \Delta_m}{\operatorname{argmax}} (x'_1)^T A y_1 \\ h_2(x_1, y_1) &= \text{br}(x_1) = \underset{y'_1 \in \Delta_n}{\operatorname{argmax}} x_1^T A (y'_1) \end{aligned} \tag{3.2}$$

那么 h 的不动点就是纳什均衡。可惜的是，由于多个可能最佳反应的存在，这个函数并不是良定义的。所以，我们考虑对函数做一些“正则化”，使得它良定义，亦即，取 argmax 的目标函数恰好有一个最大值点。我们自然联想到，严格凹函数就恰好具有这样的性质。

定义函数 $f_1 : C \rightarrow \Delta_m$ ，满足：

$$\begin{aligned} f_1(x_1, y_1) &= \underset{x'_1 \in \Delta_m}{\operatorname{argmax}} \{ (x'_1)^T A y_1 - \|x'_1 - x_1\|_2^2 \} \\ f_2(x_1, y_1) &= \underset{y'_1 \in \Delta_n}{\operatorname{argmax}} \{ x_1^T A (y'_1) - \|y'_1 - y_1\|_2^2 \} \end{aligned}$$

³初始证明用到了 Kakutani 不动点定理。事实上可以证明，Kakutani 定理和 Brouwer 定理是等价的，但是这里我们用 Brouwer 定理，因为它的形式更加简单。本节的练习题中介绍了 Kakutani 不动点定理和如何用它证明 Nash 均衡的存在性。

下面，我们证明 $f : C \rightarrow C : f(x, y) = (f_1(x), f_2(y))$ 是一个连续函数，并且它的不动点是一个纳什均衡。

注意到，关于 x'_1 ， $(x'_1)^T A x_2$ 是线性的，而 $\|x'_1 - x_1\|_2^2$ 是严格凸的，因此，这个添加距离惩罚项的操作让目标函数从线性函数变成了严格凹函数，因此保证了最大值点的唯一性，因此 f 是良定义的。并且，函数 f 是连续的。（证明留作习题）

最后，策略 (x, y) 是 f 的不动点当且仅当其为纳什均衡：

(x, y) 是纳什均衡，当且仅当 x 是 y 的最佳反应且 y 是 x 的最佳反应。

x 是 y 的最佳反应当且仅当对任意 x' ，有 $(x')^T A y \leq x^T A y \Leftrightarrow (x')^T A y - \|x' - x\|_2^2 \leq x^T A y - \|x - x\|_2^2$ ，等价于 $f_1(x, y) = x$ ；同理， x 是 y 的最佳反应当且仅当 $f_2(x, y) = y$ ，因此， (x, y) 是 f 的不动点当且仅当其为纳什均衡。 \square

Exercise 3.1.

1. Kakutani 不动点定理：

X 为欧氏空间中非空紧凸集， $C : X \rightarrow 2^X$ 有闭图像（对任意序列 $\{x_n\}, \{y_n\} \subset X, x_n \rightarrow x, y_n \rightarrow y$ ，若对任意 i ， $y_i \in C(x_i)$ ，则 $y \in C(x)$ ），且任给 $x \in X$ 有 $C(x)$ 为非空闭凸集。则存在 $x^* \in X$ ，使得 $x^* \in C(x^*)$ 。

对函数 h （式 3.2）使用 Kakutani 不动点定理，证明纳什均衡的存在性。

4 纳什均衡的常见算法

虽然纳什均衡的存在性证明非常简单，而且我们在上一节也轻松给出了零和博弈的多项式时间精确算法，但是一般地计算一个纳什均衡则相当困难。我们在下一节计算复杂性的部分会给出纳什均衡的计算复杂性类，并且给出其不存在多项式时间算法的证据。因此，这一节中我们给出的纳什均衡算法没有多项式时间保证。为了方便讨论，在这一节中，我们假设 $m = n$ 。

4.1 近似纳什均衡

我们首先严格定义近似纳什均衡。

注意到，均衡的定义实际上仅和各个策略之间的相对大小有关，亦即纳什均衡具有某种意义上的不变性：

Theorem 4.1. 纳什均衡的不变性：

如果 (x^*, y^*) 为博弈 (A, B) 的纳什均衡，那么它也是博弈 $(A + c, B + c)$ 和博弈 (kA, kB) 的纳什均衡。其中 $A + c \triangleq (A_{ij} + c)$ ， a, c 为正实数。

证明. 只需注意 $x^T A \leq x^{*T} A$ 等价于 $x^T (A + c) \leq x^{*T} (A + c)$ ，等价于 $x^T (kA) \leq x^{*T} (kA)$. \square

Corollary 4.1. 博弈的标准化：

根据上面的定理，在仅涉及到纳什均衡的讨论时，我们可以不妨对矩阵 A, B 做标准化，使得 A, B 的所有分量都恰好位于区间 $[0, 1]$ 中（即：所有分量位于 $[0, 1]$ 内，且如果不是常矩阵，则恰好存在某两个分量取得 0 和 1）。同理，在零和博弈中仅涉及到均衡的讨论时，我们可以不妨令 $A + B = 0$ 。事实上，这也是我们将零和博弈称为零和博弈而不是常和博弈的原因。之后我们涉及的讨论，若无说明，均假设矩阵 A, B 的所有分量恰好位于区间 $[0, 1]$ 中。

在设定了输入矩阵 A, B 的取值范围之后，我们就可以定义近似纳什均衡了。

Definition 4.1. ϵ -近似纳什均衡：

对于一般的双人博弈 (A, B) ，策略对 (\hat{x}, \hat{y}) 称为 ϵ -近似纳什均衡，如果：

$$\begin{aligned}\hat{x}^T A \hat{y} &\geq x^T A \hat{y} - \epsilon \quad \forall x, \\ \hat{x}^T B \hat{y} &\geq \hat{x}^T B y - \epsilon \quad \forall y\end{aligned}$$

显然， ϵ 越小，近似纳什均衡就越接近纳什均衡。特别的，当 $\epsilon = 0$ 时，得到的就是纳什均衡。而且，可以看到，近似纳什均衡的近似度事实上和矩阵 A 和 B 的放缩是相关的。如果 (x, y) 是博弈 (A, B) 的 ϵ -近似纳什均衡，那么 (x, y) 是 (kA, kB) 的 $k\epsilon$ -近似纳什均衡。因此，如果对支付矩阵的规模不做限定，就不能使用这个定义来分析算法的一般情况。

4.2 LMM 采样算法

2003 年，Lipton, Markakis 和 Mehta[27] 提出了一个对任意 $\epsilon > 0$ ，时间复杂度为 $n^{O(\log n/\epsilon^2)}$ 的算法，下面简称 LMM 采样算法。我们介绍这个算法是因为当 ϵ 为常数时，这个算法虽然不是近似算法，但是可以达到拟多项式 (quasi-polynomial，即 $2^{O(\log^c n)}$ ， $c > 0$ 为某个常数) 级别，实际效率相当高。

注意到，任意混合策略 x 是在所有纯策略上的一个概率分布。因此， $\text{supp}(x) = \{i | x_i \neq 0\}$ 给出了 x 使用的纯策略的数量。 $|\text{supp}(x)|$ 称为 x 的样本数量。我们称一个策略 x 是 k -一致的，如果其样本数量小于等于 k 。

该算法的思路非常直接：首先证明存在样本数量为 $O(\log n/\epsilon^2)$ 的 ϵ -近似纳什均衡，然后枚举所该类型的策略，找到该近似纳什均衡。

因此，我们分两步证明：

Theorem 4.2. 小样本近似纳什均衡的存在性：

在标准双人博弈 (A, B) 中，对于任意的纳什均衡 (x^*, y^*) 和 $\epsilon > 0$ ，存在一对 $\lceil 12 \ln n/\epsilon^2 \rceil$ -一致的策略 (x', y') ，使得： x', y' 是一个 ϵ -近似纳什均衡。且：

$$\begin{aligned}\left| x'^T A y' - x^{*T} A y^* \right| &< \epsilon \\ \left| x'^T B y' - x^{*T} B y^* \right| &< \epsilon\end{aligned}$$

证明. 证明由概率方法给出。大致思想类似大数定律：以 (x^*, y^*) 为概率在纯策略空间中做采样，并且以策略在采样中的出现频率给出 (x', y') 的分布，并证明：在采样了足够多次后， (x', y') 的性质充分接近 (x^*, y^*) 。

固定 $k \geq 12 \ln n/\epsilon^2$ ，以 x^* 为概率分布独立地采样 k 次，得到一个可重集合 U ；以 y^* 为概率分布独立地采样 k 次，得到一个可重集合 V 。对于集合 U, V ，我们直接计算各个策略在其中的出现频率，得到概率分布 x', y' ，即： $x' = \sum_{u \in U} u/k, y' = \sum_{v \in V} v/k$ 。

定理中要求的条件可以拆分为下列事件：

$$\begin{aligned}\phi_1 &= \left\{ \left| x'^T A y' - x^{*T} A y^* \right| < \epsilon/2 \right\} \\ \pi_{1,i} &= \left\{ e_i^T A y' < x'^T A y' + \epsilon \right\}, \quad (i = 1, \dots, n) \\ \phi_2 &= \left\{ \left| x'^T B y' - x^{*T} B y^* \right| < \epsilon/2 \right\} \\ \pi_{2,j} &= \left\{ x'^T B e_j < x'^T B y' + \epsilon \right\}, \quad (j = 1, \dots, n)\end{aligned}$$

那么, 我们寻找的满足条件的 (x', y') 应该在集合 $GOOD = \phi_1 \cap \phi_2 \bigcap_{i=1}^n \pi_{1,i} \bigcap_{j=1}^n \pi_{2,j}$ 中。因此, 只需要证明 $\Pr[GOOD] > 0$ 。

进一步, 我们考虑事件

$$\begin{aligned}\phi_{1a} &= \{|x'^T Ay' - x'^T Ay^*| < \epsilon/4\}, \\ \phi_{1b} &= \{|x'^T Ay^* - x^{*T} Ay^*| < \epsilon/4\}\end{aligned}$$

显然 $\phi_{1a} \cap \phi_{1b} \subset \phi_1$, 从而 $\phi_1^c \subset \phi_{1a}^c \cup \phi_{1b}^c$ 。之后我们要估计这两个事件的发生概率。

Hoeffding 不等式: 若 X_1, \dots, X_n 为独立的随机变量, 且 $X_i \in [a, b], i = 1, \dots, n$ 。则:

$$\forall t > 0, \quad \Pr(|\bar{X} - E[\bar{X}]| \geq t) \leq 2 \exp\left(-\frac{2n^2 t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right)$$

由于 y' 是 y^* 的 k 次采样的频率, 事实上我们可以将 ϕ_{1a} 中的事件看做是以期望为 $X = x'^T Ay^*$ 独立采样了 k 个随机变量 $X_v = x'^T Av, v \in V$, 而其均值根据定义正是 $x'^T Ay'$ 。并且, 根据博弈的标准性, X_v 的取值范围总是 $[0, 1]$ 。的那么, 根据 Hoeffding 不等式, $\Pr(\phi_{1a}^c) = \Pr(|x'^T Ay' - x'^T Ay^*| \geq \epsilon/4) \leq 2 \exp\left(-\frac{k\epsilon^2}{8}\right)$ 。同理, 可以证明 $\Pr(\phi_{1b}^c) \leq 2 \exp\left(-\frac{k\epsilon^2}{8}\right)$, $\Pr[\psi_{1,i}^c] \leq e^{-k\epsilon^2/2}, \Pr[\psi_{2,j}^c] \leq e^{-k\epsilon^2/2}$ 。

综上, 代入 $k\epsilon^2 \geq 12 \ln n$:⁴

$$\begin{aligned}\Pr[GOOD^c] &\leq \Pr[\phi_1^c] + \Pr[\phi_2^c] + \sum_{i=1}^n \Pr[\pi_{1,i}^c] + \sum_{j=1}^n \Pr[\pi_{2,j}^c] \\ &\leq 8e^{-k\epsilon^2/8} + 2n \left[e^{-k\epsilon^2/2} + 4e^{-k\epsilon^2/8} \right] \leq 16n^{-2/3} + 2n^{-5} \leq 1.\end{aligned}$$

□

Theorem 4.3. 存在一个时间复杂度为 $n^{O(\ln n/\epsilon^2)}$ 的算法找到一个 ϵ -近似纳什均衡。

证明. 我们考虑直接枚举对于任意可能生成的 (x, y) , 以寻找上面已经证明存在的近似均衡。

上面的采样相当于在两个玩家所有策略中可重复地选取 k 次, 得到 U 和 V 。因此, 所有可能结果的数量为 $\binom{n+k-1}{k}^2 = O(n^k)$; 而验证一个选取的策略 (x', y') 是否为近似纳什均衡, 只需要分别计算偏离到各个纯策略的收益和当前收益之差是否小于 ϵ 即可, 至多只需要 $O(n^2)$ 次计算。

综上, 该算法的计算复杂度为 $n^{O(k)} = n^{O(\ln n/\epsilon^2)}$ 。 □

Exercise 4.1.

1. 证明: 如果在定理 4.2 中去掉对 x', y' 是近似纳什均衡的要求, 那么, 可以使用至多 $O(1/\epsilon^2)$ 次采样找到这样的策略对 (x', y') 。

Remark 4.1. 事实上, 对于任意的 n 和 ϵ , 我们可以给出一个构造, 表明我们给出的采样数量的界 $O(\log n/\epsilon^2)$ 是紧的。但是构造过程非常复杂 (见 [tightness nlogn])。

4.3 Lemke-Howson 算法

上一节我们介绍的 LMM 采样算法虽然使用了枚举法, 但理论效率并不低。不过, 我们也希望能够找到类似于零和博弈中线形规划方法的直接求解精确算法。1964 年, Lemke-Howson 等人提出的算法则是一个非常好的例子。他们的思路与线性规划问题中的单纯形法非常类似, 即寻找纳什均

⁴这里的系数 12 并不重要, 但是从下面的放缩中可以看出, 阶 $\ln n/\epsilon^2$ 是必要的

衡的组合结构，将其转化为图上的搜索问题。这个搜索问题实际上和纳什均衡计算的组合结构是紧密相关的，我们将会在后文复杂性分析的部分深入探讨这个问题。

一个自然的想法是把纳什均衡的性质做一些抽象。在下面的讨论中，我们把两个玩家的策略合并讨论，即：用分量 $1 \sim m$ 标记玩家 1 的策略，用分量 $m+1 \sim m+n$ 标记玩家 2 的策略。注意到，策略对 (x^*, y^*) 是纳什均衡要求对任意策略 $x, x^T A y^* \leq x^{*T} A y^*$ 。但是，本身 x^* 作为 $\text{supp}(x^*)$ 上的一个混合策略，其期望收益就是这些纯策略的加权平均。此时又要求对其中任意一个纯策略，收益都小于等于 x^* 的收益，因此必须有 $\forall i \in \text{supp}(x^*), e_i^T A y^* = x^{*T} A y^*$ 。此外，还有 $x^{*T} A y^* = \max_x x^T A y^* = \max_k e_k^T A y^*$ 。同理可以对 y^* 做类似推导。综上，我们得知，一个策略对 (x^*, y^*) 是纳什均衡当且仅当：

$$\begin{aligned} \forall 1 \leq i \leq m, x_i > 0 &\Rightarrow (A y^*)_i = \max_k (A y^*)_k \\ \forall m+1 \leq j \leq m+n, y_j > 0 &\Rightarrow (x^{*T} B)_j = \max_k (x^{*T} B)_k \end{aligned}$$

为了抽象这些概念，我们定义：

$$\begin{aligned} P &= \left\{ (u, \mathbf{x}) \mid x_i \geq 0, \sum x_i = 1, \mathbf{x}^T B \leq u \cdot \mathbf{1} \right\} \\ Q &= \left\{ (v, \mathbf{y}) \mid y_j \geq 0, \sum y_j = 1, A \mathbf{y} \leq v \cdot \mathbf{1} \right\} \end{aligned}$$

P, Q 为两个多面体。这两个多面体的顶点将是我们研究的主要对象，因为多面体的顶点必然使得若干不等式取等，如 $x_i = 0$ 则说明策略 i 没有出现， $(x^T B)_j = u$ 则说明策略 j 是 \mathbf{x} 的最佳反应。并且，根据上面纳什均衡的定义，任意纳什均衡 (x^*, y^*) ， x^*, y^* 必定分别为 P, Q 的顶点（即要求部分不等号取等）。

为了方便讨论，考虑下列变换：

$$\begin{aligned} \bar{P} &= \{ \mathbf{x} \mid x_i \geq 0, \mathbf{x}^T B \leq \mathbf{1} \} \\ \bar{Q} &= \{ \mathbf{y} \mid y_j \geq 0, A \mathbf{y} \leq \mathbf{1} \} \end{aligned}$$

那么，除了零点之外，存在 P 与 \bar{P} ， Q 与 \bar{Q} 的顶点的一一对应：事实上，对 P 中任意顶点 (u, \mathbf{x}) ， \mathbf{x}/u 是 \bar{P} 中顶点；对任意 \bar{P} 的非零顶点 \mathbf{x} ， $(1/\sum x_i, \mathbf{x}/\sum x_i)$ 是 P 的一个顶点。 Q 同理。

下面，我们更加清晰地刻画 \bar{P}, \bar{Q} 的顶点和纳什均衡的关系。构造图 G_1, G_2 ，其顶点分别为 \bar{P}, \bar{Q} 的顶点，而两个顶点连边当且仅当对应顶点在多边形中相邻。每个顶点 x 唯一对应一组紧约束 $L(x) = \{i \mid x_i = 0\} \cup \{j \mid (x^T B)_j = 1\}$ ， y 同理。这里，我们假设讨论的纳什均衡是非退化的，亦即对任意策略 u 及其最佳反应 v ， $\text{supp}(u) \leq \text{supp}(v)$ 。这样，我们能够保证对玩家 1 的任意策略 x ， $|L(x)| \leq m$ ；对玩家 2 的任意策略 y ， $|L(y)| \leq n$ ；对于顶点， $|L(x)| = m, |L(y)| = n$ 。⁵

这时候，我们把纳什均衡的定义转移到这个图上：我们注意到，顶点 (x, y) 是一个纳什均衡当且仅当 $L(x) \cup L(y) = \{1, 2, \dots, m+n\}$ 。（证明留作习题）

这样，纳什均衡的计算就转化成了在图中搜索特定顶点的组合问题。基于该转化后的问题，Lemke 和 Howson 提出了 Lemke-Howson 算法：

取图 G_1 和 G_2 的积 $G = G_1 \times G_2$ ，即： $V(G) = V(G_1) \times V(G_2)$ ， $E(G) = \{((u_1, v_1), (u_2, v_2)) : (u_1, u_2) \in E(G_1), (v_1, v_2) \in E(G_2)\}$ 。我们定义顶点 (u, v) 的标签为 $L(u, v) \cup L(v)$ 。考虑图 G 的子图 $U_k = \{v \in V(G) \mid L(v) \supseteq \{1, 2, \dots, m+n\} \setminus \{k\}\}$ ，那么，图 G 具有以下性质：

⁵可以证明，退化的博弈经过恰当的扰动可以成为非退化的博弈。细节可以参考 [31] 第 3.6 章

对于零向量，定义其 supp 为所有分量的集合。则：对于任意 $k \in \{1, 2, \dots, m+n\}$ ，零点 $(0, 0)$ 和所有的纳什均衡都属于 U_k ，并且其度数在图 U_k 中恰好为 1；而其它顶点的度数则恰好为 2。

严格的证明比较复杂，我们给出一个比较直观的解释：对前者而言，通过去掉策略 k 在多边形中可以“松弛”到的方向只有 G_1, G_2 中的恰好一个；而对后者而言，由于 $|L(x) \cap L(y)| = 1$ ，所以在 G_1, G_2 中都可以松弛，且松弛到的方向恰好有两个。由于任意图中奇数度的顶点必须有偶数个，我们可以得到一个简单的推论：非退化博弈中恰好有奇数个纳什均衡。

据此，我们可以得到 Lemke-Howson 算法：

1. 选择一个标签 $k \in \{1, 2, \dots, m+n\}$ 作为枢轴，令 $(x, y) = (0, 0)$ ，重复步骤 2、3 直到算法终止：
2. 从顶点 (x, y) 出发，在 (x, y) 中扔掉标签 k ，得到一个新的顶点 (x', y')
3. 由于 $(x', y') \in U_k$ ，因此要么 (x', y') 是一个纳什均衡，算法结束；要么 (x', y') 度数为 2，记 (x, y) 以外的一个邻居为 (x'', y'') ，更新 (x, y) 为 (x'', y'') 。（由于顶点 $(0, 0)$ 度数为 1，根据我们的寻径方式，不可能回到 $(0, 0)$ ）

我们注意到， L 的定义实际上给出了需要满足的若干个线性约束，而纳什均衡就是这些线性约束的解。但是，直接同时求解这些线性约束是不可能的，因为它同时涉及到 x 和 y 两个向量，所以我们的思路是每次只对一个线性约束求解。这个“松弛”和“更新”的过程就是去掉/加回第 k 个约束的过程，反应在多边形上则是在多边形的顶点上沿着路径寻找纳什均衡的过程。这种寻径的方法与单纯形法非常类似，都称为枢轴（pivoting）方法。

Exercise 4.2. 1. 证明：图 $G_1 \times G_2$ 的顶点是纳什均衡当且仅当 $L(x) \cup L(y) = \{1, 2, \dots, m+n\}$ 。

2. (Lemke Howson 算法实例) Lemke Howson 算法的求解过程比较复杂，下面，我们用一个例子和图示来说明算法执行的过程：

考虑矩阵 $A = \begin{bmatrix} 3 & 3 \\ 2 & 5 \\ 0 & 6 \end{bmatrix}$ ， $B = \begin{bmatrix} 3 & 2 \\ 2 & 6 \\ 3 & 1 \end{bmatrix}$ ，则：

$$\overline{P} = \{x | x_1 \geq 0(1), x_2 \geq 0(2), x_3 \geq 0(3), 3x_1 + 2x_2 + 3x_3 \leq 1(4), 2x_1 + 6x_2 + x_3 \leq 1(5)\}$$

$$\overline{Q} = \{y | 3y_4 + 3y_5 \leq 1(1), 2y_4 + 5y_5 \leq 1(2), 6y_5 \leq 1(3), y_4 \geq 0(4), y_5 \geq 0(5)\}$$

$\overline{P}, \overline{Q}$ 的顶点及其根据不等式取等条件的编号如图 1 所示：

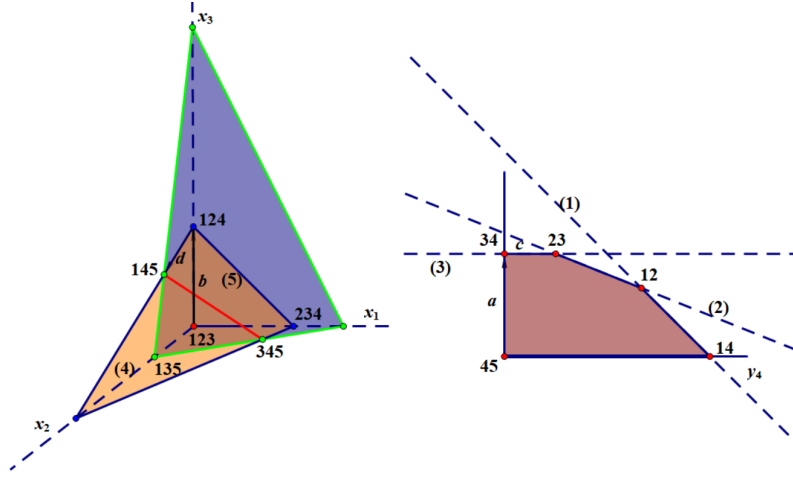


图 1: 多边形 \bar{P}, \bar{Q}

编号的计算：以左边编号 345 的点为例：其使得编号为 (3), (4), (5) 的三个不等式取等，因此编号为 345，右边编号为 23 的点为例：其使得两个不等式取等：(2), (3)，因此编号为 23。

由于我们要还原顶点之间的移动，而顶点其实就不等式的极值情况。因此，枢轴转换从一个顶点移动到另一个顶点的过程其实就是让枢轴变量从取值范围的一个极值移动到另一个极值，同时保持其他标签不变（也就是其他被标签标记的不等式的取等情况不变）的过程。

注意：下面说提及的点都是指 $G_1 \times G_2$ 中的点，其标签为两个集合中标签的并集。例如，零点 $(0, 0, 0, 0, 0)$ 分为 \bar{P} 中的 $(0, 0, 0)$ 和 \bar{Q} 中的 $(0, 0)$ 两部分。 $(0, 0, 0)$ 在 \bar{P} 中的标签为 123， $(0, 0)$ 在 \bar{Q} 中的标签为 45，因此零点的标签为 $123 \cup 45$ 。从零点出发，以 5 为枢轴，就是在 \bar{Q} 中以 y_5 为枢轴。目前，零点在 \bar{Q} 中的坐标为 $(y_4, y_5) = (0, 0)$ ，标签为 45。因此，枢轴变换就相当于从 $(0, 0)$ 的取值出发，保持编号为 4 的不等式取等不变（即 $y_4 = 0$ ），让 y_5 从一个极值移动到另一个极值。在 \bar{Q} 的约束条件下，容易求出 $0 \leq y_5 \leq 1/6$ 。那么，我们令顶点由 $(0, 0, 0, 0, 0)$ 移动到 $(0, 0, 0, 0, 1/6)$ ，即从 45 到 34 的路径，标记为 a 。新的顶点的标记为 $123 \cup 34$ ，于是标签 5 被“松弛”掉了。

下一步则是标签 5 的更新。我们刚才在 \bar{Q} 中更新，再移动就要回去了。所以，这一步，我们把目光移到 \bar{P} 中。（这一步是之前描述的非均衡点度数为 2 的直观表现：重复标签同时出现在 \bar{P}, \bar{Q} 两个集合中，所以两边的枢轴给出了两个消去重复标签的可行方向。）由于目前重合的是标签 3，所以这次以标签 3，即 x_3 为枢轴。容易计算，在 $x_1 = x_2 = 0$ 和 \bar{P} 的条件下， $0 \leq x_3 \leq 1/3$ ，所以这次应该把 x_3 移动到值为 $1/3$ 处，也就是沿着从 123 到 124 的路径 b ： $(0, 0, 0, 0, 1/6) \rightarrow (0, 0, 1/3, 0, 1/6)$ 的移动。

现在，我们的标签中，4 又重复了。我们再移动到 \bar{Q} 中做枢轴变换。在点 34 处，保持编号 3 的不等式 $6y_5 = 1$ 不动，求出 $0 \leq y_4 \leq 1/12$ ，因此这次沿着从 34 到 23 的路径 c 移动：从 $(0, 0, 1/3, 0, 1/6) \rightarrow (0, 0, 1/3, 1/12, 1/6)$ 。

现在，2 重复了。我们在 \bar{P} 中做变换。在点 124 处，保持 $x_1 = 0, 3x_1 + 3x_2 + 3x_3 \leq 1$ 不动，则 $0 \leq x_2 \leq 1/3$ ，于是沿着 124 到 145 的路径 d 移动：从 $(0, 0, 1/3, 0, 1/6)$ 到 $(0, 1/3, 1/3, 1/12, 1/6)$ 。

此时，我们发现目前的标签集是 $145 \cup 23$ ，所以我们现在得到了一个纳什均衡！把当前各个点的坐标代入验证： $x_1 = 0, x_2 = 1/3, x_3 = 1/3, y_4 = 1/12, y_5 = 1/6$ ，确实是一个纳什均衡。

参考上面给出的 Lemke Howson 算法的实例，给出：

(1) 对于一般的博弈 $A, B \in \mathbb{R}^{m \times n}$ ，给出 Lemke Howson 算法实现的详细文字描述。

(2) 对于实例 $A = B^\top = \begin{bmatrix} 3 & 3 & 0 \\ 4 & 0 & 1 \\ 0 & 4 & 5 \end{bmatrix}$ ，以 3 为初始枢轴，用 *Lemke Howson* 算法计算该博弈的

纳什均衡。

Remark 4.2. 单纯形法和 *Lemke Howson* 算法具有路径查找算法的通病，即最坏复杂度和平均复杂度间隔很大。

Idc: [需要提一下单纯形法的历史吗?] 单纯形法虽然在一般情况下的运行都非常快，速度往往超过有多项式最坏时间保证的内点法；然而，在最坏情况下，单纯形法的复杂度是指数级别的。

同理，*Lemke Howson* 算法是当前计算纳什均衡的最常用算法，在大部分情况下运行结果都不错；然而，可以证明该算法的最坏复杂度是 *PSPACE-complete*（多项式空间完全）的。

这个现象从直观上理解，就是这两个算法的流程都相当于在一个巨大的二度图上查找所需要的顶点。单纯形法的顶点数量的上界是 $O(2^n)$ ，而可以证明 *Lemke Howson* 算法定义的 U_k 的顶点数量上界为 $O(\binom{n+m}{n})$ 。由于图的度数为 2，那么如果使用路径查找方法，就是从这个图中的某条链的一端（零点）移动到另一端。虽然一般情况下需要走过的路径很短，但是在非常差的情况下，这条链的长度是可以接近整个图的顶点数量的，所以造成了单纯形法和 *Lemke Howson* 算法的最坏复杂度非常之高。虽然这种非常差的情况的出现概率接近 0，但仍然是可能出现的。

平均时间复杂度只能刻画问题在所有情况下的平均复杂性，并不足以描述这种现象。有没有办法准确地描述这种局部极端性呢？一个比较自然的想法是考虑对这种非常差的情况做“扰动”后是否可以移动到比较正常的情况。2004 年，Daniel Spielman 和滕尚华在 [38] 中提出了刻画这种现象的平滑分析法（smoothed analysis）。他们证明了约束的数量为 n ，向量空间为 d 的二阶段投影顶点在尺度为 σ 的扰动下复杂度是 $\text{poly}(n, d, 1/\sigma)$ 的，从而准确地刻画了这种现象。可是，对于 *Lemke Howson* 算法，这种平滑性是否成立呢？这将是我们在接下来关于复杂性理论的部分即将探讨的问题。

5 纳什均衡的计算复杂性

在第 4.1 中，我们利用 Brouwer 定理证明了纳什均衡的存在性，这似乎揭示了它们之间的某种联系。令人惊奇的是，我们事实上可以借助计算复杂性理论证明它们有着某种关联的组合结构，并且借助这种结构证明它们的计算复杂性是相等的。

5.1 复杂性类：TFNP, PPAD 和 PPA

我们首先容易注意到，纳什均衡是属于 NP 的。这是因为显然纳什均衡的解就是一个多项式验证：对于给定的一组概率分布，我们很容易通过定义在多项式时间内验证其是否是一个纳什均衡。但是，纳什均衡是否是 NP-困难的呢？从直观上来看，似乎不是。这是因为纳什均衡保证了解的存在性，而对于一般的 NP 问题，解是不一定存在的。所以，直观上看，纳什均衡的求解比一般的 NP 问题更“容易”。

为了解释这类保证了解一定存在的问题的复杂性，Megiddo 和 Papadimitriou 在 1991 年提出了 TFNP 复杂性类。TFNP 类全称“全函数多项式时间类”，是基于 FNP（函数多项式时间）类概念提出的。

首先回忆一下 NP 类的定义：

判定问题 $L \in NP$, 若: 存在确定性多项式时间图灵机 M , 对任意字符串 x , $x \in L$ 当且仅当存在多项式平衡的验证字符串 $y \in L$, $M(x, y) = 1$ 。

Idc: [前面复杂性基础介绍一定要 clarify 语言和问题, 判定问题和函数问题的区别: 判定问题对应的是 $\Sigma^* \rightarrow \{0, 1\}$ 的函数的复杂性, 而函数问题则对应 $\Sigma^* \rightarrow \Sigma^*$ 的复杂度。细节参考 https://en.wikipedia.org/wiki/Function_problem?useskin=vector]

现在, 将这种存在关系提取出来: 判定问题 $L \in NP$ 当且仅当: 存在多项式平衡、多项式可验证的二元关系 $R(x, y)$, 使得 $x \in L$ 当且仅当存在 y , $R(x, y)$ 成立。

对于 NP 问题与二元关系 R 的这种对应性, 我们可以定义 NP 问题对应的函数 (functional) 复杂性类 (注: function 是一个误称, 因为实际上是对二元关系的分类) 为 FNP 类:

称一个二元关系 R 是 FNP 的, 若对于任意 x 和任意长度为关于 x 的多项式的 y , $R(x, y)$ 可以在确定性多项式时间内计算出。

因此, 根据 NP 问题的定义, 对于每个 NP 问题, 都存在其对应的二元关系 R , 使得 R 是 FNP 的。反之, 给定的 FNP 的二元关系 R , 语言 $L = \{x : \exists y, R(x, y)\}$ 一定属于 NP。例如, 对哈密顿回路问题而言, 判定图中是否存在哈密顿回路的复杂度是 NP 的, 而寻找这样的图和回路的关系的二元问题的复杂度就是 FNP 的。而且, 由于可能有多个解, 一个 NP 问题可以对应多个 FNP 函数问题。不过, 由于 NP 和 FNP 所描述的对象是不同的, 虽然我们可以说如果一个问题对应的二元关系问题如果是 FNP 的, 那么它就是 NP 的, 但是我们很难在这两个类间建立起通用的某种归约。

对于给定的 FNP 关系 R , 其上的搜索问题定义为: 对给定 x , 寻找 y , 使得 $R(x, y)$ 成立, 或者断言这样的 y 不存在。可以证明, $P=NP$ 当且仅当对任意 FNP 关系 R , 搜索问题可多项式求解。

基于该函数复杂性类, 我们额外要求其中的二元关系一定保证对任意 x , $R(x, y)$ 成立, 即可得到函数复杂性类 TFNP。同理, 我们同时定义 TFNP 为通过关系 R 可以在确定性多项式时间内搜索到 x 对应的 y 的复杂性类。显然, TFNP 是属于 NP 的, 但是尚不知道 TFNP 是否是 NP 完全的。不过, 最近的一个工作 [21] 表明, 如果证明了任意一个 TFNP 问题是 NP 完全的, 那么必然有 $NP=co-NP$ 。

由于解的存在性, TFNP 类问题往往与搜索和优化类问题紧密关联, 例如求解某个函数的全局极小值、不动点等等。我们讨论的纳什均衡的计算问题就属于 TFNP。我们只要验证, 一方面, 纳什均衡计算是属于 NP 的; 另一方面, 纳什均衡对任意输入 x (博弈的支付矩阵), 对应的解 y 均存在。⁶

然而, TFNP 类的定义存在致命的缺陷。这是因为, TFNP 类是语义 (semantic) 类, 而常见的复杂性类, 如 P, NP 等, 都是语法 (syntactic) 类。语法类的定义要求其成员 (membership) 是可以通过计算模型判定的, 也就是说, 存在一个图灵机去判定任意一个语言是否属于该语法类; 而语义类则是尚未发现这种图灵机的复杂性类。这种特质进一步使得语法类往往具有完全问题, 而目前还尚未发现任何语义类的完全问题。这是因为, 我们可以使用语法类语言的判定机来构造一个“通用”的完全问题: 我们只需要寻找一个属于该语法类, 但是可以描述这个“通用判定机”的问题, 那么这个问题就是该复杂性类的完全问题。回忆一下 SAT 问题是 NP 完全的证明, 本质是就是把 NP 完全问题的“通用判定机” (输入语言和实例, 多项式时间内停机的 NTM) 利用 SAT 问题描述, 进而推导出 SAT 问题是 NP 完全的。

例如, P, NP 都是语法类, 因为它们都是可枚举的, 更不必说可以判定某个语言是否在其中了 (直接按照枚举出来的列表查找即可)。但是, 对 TFNP 类这种语义类而言, 目前只能通过数学证明说明某些问题属于 TFNP 类。试想, 如果存在某个图灵机判定一个语言是否属于 TFNP 类, 那么这

⁶注意, 并不是任意一个搜索问题都是 TFNP 问题。TFNP 问题首先要是多项式可判定的。例如, 著名的最大割问题, 它是 NP 困难的, 解也一定存在, 但是现在尚未证明它是 TFNP 问题, 因为对于给出的一个解, 目前尚不知道如何在多项式时间内验证它到底是不是最大割。

个机器应该得学会复杂逻辑证明，直觉上这样的机器并不存在。

完全问题是我们研究复杂性类的重要工具。因此，为了解决这个问题，1995 年，Papadimitrou [32] 定义了复杂性类 PPA 和 PPAD。他的想法是：把解决 TFNP 类中的某些问题所需要的数学证明模式直接“告诉”机器。这样，机器就不需要具备逻辑证明的能力。通过这种方式，可以定义了 TFNP 类的子类 PPA 和 PPAD。

PPA：给定一个确定性多项式时间图灵机 M ：对于任意给定 x ，定义 $C(x)$ 为长度是 $|x|$ 多项式倍的字符串集合。对任意 $c \in C(x)$ ， M 在关于 n 的多项式时间内输出 $M(x, c)$ ，其为至多 $C(x)$ 中字符串的集合。定义以 $C(x)$ 为顶点集的图 $G(x)$ ，任意 $c, c' \in C(x)$ 连边当且仅当 $c' \in M(x, c)$ 且 $c \in M(x, c')$ 。这样，图 $G(x)$ 是顶点度数至多为 2 的无向图。我们另外要求 M 满足 $M(x, 0 \cdots 0) = \{1 \cdots 1\}$ ， $0 \cdots 0 \in M(x, 1 \cdots 1)$ ，使得 $0 \cdots 0$ ，以使得 $0 \cdots 0$ 始终是图 $G(x)$ 的一个叶子。我们定义下列搜索问题 A ：对任意 x ，找到 $G(x)$ 除 $0 \cdots 0$ 的另一个叶子节点。⁷

这个定义很冗长，不过，这是为了在问题的定义中引入存在性模式。这个问题对应的证明模式就是“度数至多为 2 的无向图中有偶数个叶子节点”。我们注意到，这个证明模式其实与之前 Lemke Howson 算法中 U_k 的结构非常相似。

PPAD 则是将 PPA 中 $G(x)$ 的定义换成有向图后得到的复杂性类，具体构造方式是：有边 c 指向 c' 当且仅当 c' 是 $M(x, c)$ 的第二个分量，而 c 是 $M(x, c')$ 的第一个分量。

Exercise 5.1. 1. 定义因子分解问题如下：给定一个正整数，输出它的质因子分解。证明：因子分解问题属于 TFNP。

2. (PPA 是语法类) 为 PPA 中的关系构造一个判定图灵机 N ：对任意关系 R ， N 接受 R 当且仅当 R 属于 PPA。

3. PPA 和 PPAD 的定义模拟了关于无向图和有向图的定理“度数至多为 2 的无向图中有偶数个叶子节点”。模仿它们的定义，给出模拟定理“每个有向无环图都有一个汇（出度为 0 的点）”的复杂性类 PLS(Polynomial Local Search) 的定义。

5.2 PPAD 的完全问题：Sperner

PPAD 的结构在上一节的定义中已经明确了，其完全问题呼之欲出：

Definition 5.1. (EOL: End of the Line) EOL 问题定义如下：令 $V = \{0, 1\}^n$ ， $N = |V| = 2^n$ 。图 G 由前驱和后继函数 $pred$ 和 $succ$ 给出，使得 $pred(succ(u)) = succ(pred(u)) = u$ ，并且 $pred(0 \cdots 0)$ 为空。试找出一个度数为 1 的顶点。⁸

显见，EOL 是 PPAD 完全问题。一方面，任何属于 PPAD 的问题，根据 PPAD 的定义都具有上述结构。另一方面，EOL 本身把结构转化成字符串后也是 PPAD 中的问题。之后，我们对 PPAD 问题的完全性的讨论都基于 EOL 的归约。

回忆 Lemke Howson 算法 (Section 4.3)，它的结构和 EOL 很像。从它的结构可以自然地推出下面的定理：

Theorem 5.1. 精确纳什均衡的计算问题属于 PPAD。

⁷注：PPA 的定义中其实引入了 M 作为 oracle，但我们简单理解为只能通过给定的 M 来获取关于 G 的信息，不过多涉及这部分内容。

⁸这里为了清晰起见省略了一些技术上的细节。实际定义的时候会利用技巧把对 $pred$ 和 $succ$ 的要求移动到对解的要求中，使得问题的定义与语义无关，从而迫使问题不是语义类，详情可参考 [33] 第 2.6 节。

证明. 只需注意到 Lemke Howson 算法在 $m = n$ 的情况下, 顶点总数至多为 $O(\binom{2n}{n}) = 2^{O(n)}$, 因此, 精确纳什均衡的计算问题可以通过该算法归约到 EOL 上。□

接下来, 我们把注意力集中在证明纳什均衡是 PPAD 完全的之上。一旦证明这个结论, 我们就完全明确了纳什均衡的复杂性层次, 将它锁定在 PPAD 类之中。但是, 纳什均衡的结构和 EOL 距离太远, 为了逐步把 EOL 归约到纳什均衡的计算问题上, 我们需要建立一系列 PPAD 完全问题。

n 维单纯形 $\Delta(x_0, x_1, \dots, x_n) = \{v \mid v = \sum_{i=0}^n \alpha_i x_i, \alpha_i \geq 0, \sum_{i=0}^n \alpha_i = 1\}$ 上的面定义为 $\Delta(x_{i_1}, x_{i_2}, \dots, x_{i_k})$, 其中 $0 \leq i_1 < i_2 < \dots < i_k \leq n$. 对 $v \in \Delta$, 定义 $\chi(v) = \{i \mid \alpha_i > 0\}$, 定义 Δ 的一个单纯剖分为单纯形集合 $\Delta_i, i = 1, \dots, m$, 使得 $\Delta = \bigcup_i \Delta_i$, 且对任意 i, j , $\Delta_i \cap \Delta_j$ 要么是空集, 要么是 i 和 j 的一个公共面。

Theorem 5.2. (Sperner 定理, 1928[37]) 对于一个单纯形和它的单纯剖分, 可以定义其顶点集 V 上的一个染色 $\lambda: V \rightarrow \{0, 1, \dots, n\}$. 称一个染色是合法的, 若对任意 $v \in V$, $\lambda(v) \in \chi(v)$. 那么, 对任意一种合法染色, 必然存在剖分中的一个单纯形 Δ_i , 其所有顶点的染色恰好遍历 $\{0, 1, \dots, n\}$, 称为全色单纯形。

我们先证明这个定理, 然后讨论它和 Brouwer 以及纳什均衡的关系。

证明. 我们实际上证明更强的结论: 有奇数个这样的全色单纯形。对 n 进行归纳:

$n=1$ 时单纯形即为线段和顶点。单纯剖分即将线段 x_0x_1 分为若干个线段。根据合法性条件, x_0 的颜色必定为 1, x_1 的颜色必定为 0, 假如现在我们向线段 x_0x_1 上逐个将染色顶点加入, 那么初始全色线段的条数为 1, 而每次加入染色为 0/1 的顶点时, 全色线段数量的变化都是偶数 (加入两端点同色线段: $+0/+2$, 加入异色线段: $+0$), 因此最终全色线段数量也是奇数。

我们着重证明 $n=2$ 的情况, 因为之后的归纳都与该情况类似。此时, 单纯形为多边形、线段和顶点。对于多边形, 不妨做三角剖分, 使得最终得到一个三角剖分。

由于此时单纯形的三条边都分别是合法的, 随意挑选其中一条 (图中为 x_1x_2 所在边), 运用归纳假设知, 这条边中有奇数个双色线段。现在, 我们从其中一个双色线段出发, 搜索一个全色三角形。注意到, 以该线段为边的三角形的另一个顶点有两种情况:

1. 与这两个顶点异色, 则我们已经找到了一个双色三角形
2. 与其中一个顶点同色, 那么这个三角形中恰好有两个异色线段, 我们移动到新的异色线段上, 继续循环上面的过程。

注意到, 上面的搜索过程至多经过每个三角形一次, 且移动出边界时, 根据合法性, 只能从下边界移出。(证明留作习题) 因此, 上面的过程要么以找到一个全色三角形终止, 要么在进入和移出单纯形的过程中将两条异色线段配对。但是, 异色线段的数量根据归纳假设是奇数, 所以必然有一个异色线段出发的搜索过程终止于一个全色三角形, 命题得证。

下图为一个实例, 图中标出了所有算法的路径。

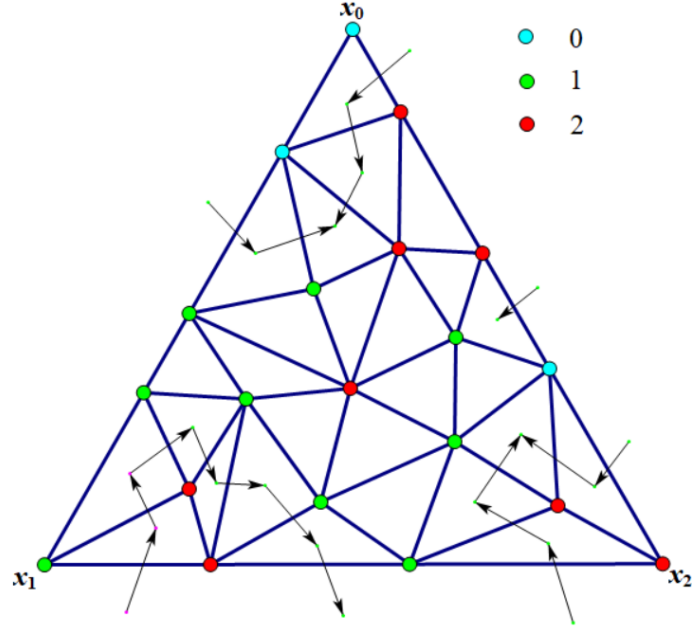


图 2: 3 维 Sperner 实例

对于更高维的情况，可以做类似的讨论，证明留作习题。 \square

我们注意到，Sperner 定理定义的路径已经和 PPAD 非常接近。下面，我们给出其对应的计算问题 m -Sperner。为了保证输入的规范性（ n 位字符串表示），我们定义 n 阶标准 Sperner 剖分为顶点集为 $\{(\alpha_0, \dots, \alpha_m) \in \Delta : n\alpha_i \in \mathbb{Z}\}$ 的单纯剖分。⁹

定义 m -Sperner 计算问题如下：

给定单纯形 $\Delta(x_0, \dots, x_m)$ 和一个图灵机 M ，其输入 n 阶标准剖分中任何一个顶点的坐标，输出该顶点的颜色。给出一个全色单纯形的重心的坐标。（注意，该定义仅与 n 有关，与具体的剖分方式无关。）

Theorem 5.3. 计算问题 m -Sperner 属于 PPAD

证明。我们可以使用恰当的标准剖分，保证三角形的数量是 $O(2^n)$ 个。从而，任何染色方式根据我们上面定理的描述都构造了一个有向图 G ，且结合 M 和证明的过程可以给出 G 的路径移动规则。由于数量为 $O(2^n)$ 个，图中的任意一个单纯形都可以被长度为 $O(n)$ 的字符串编码。因此，求解这个 Sperner 的过程归约为了查找图 G 的另一个叶子节点的过程，也就是 EOL 问题。由于 EOL 是 PPAD 完全的，我们证明了 m -Sperner 问题属于 PPAD。 \square

要注意的是，在这种情况下，我们需要把从一条双色边指向另一条双色边的路径消去，否则算法可能会在双色边处终止。不过，我们事实上可以证明不同的路径之间是没有交叉的（证明留作习题），因此，我们只需要添加由第偶数个异色边到第奇数个异色边的路径，即可保证最后算法一定进入一个全色三角形。

⁹实际上，当 $m \geq 3$ 的时候，类似二维的标准单纯剖分是不存在的（例如，无法简单地把一个三维单纯形（三棱锥）分割为八个全等的小三棱锥），但是 [32] 指出，可以通过修改规则类似地进行处理，例如：1. 另外定义一个不是标准单纯剖分的合理规则（保证各个子单纯形的直径比较接近即可）2. 在单纯形中嵌入一个超立方体，以超立方体为基础进行讨论。所以，这里我们忽略该细节，只讨论比较本质的复杂性类的关系。

同理，对高维的情况可做类似讨论。

下面，我们再把 EOL 问题归约到 Sperner 上，证明下面的定理：

Theorem 5.4. 3-Sperner 问题是 PPAD 完全的。¹⁰

证明. 由于已经证明了 m-Sperner 是属于 PPAD 的，只需给出一个 EOL 到 3-Sperner 的归约。归约的思路非常直接，其实就是用 3 维 Sperner 的某种恰当的染色来构造任意 EOL 问题对应的图。我们下面只给出证明的大致思路，忽略一些技术性的细节：

假设 EOL 问题的参数为 n ，也就是 $|V| = 2^n$ ，下面取 3 维 Sperner 问题的规范分割的阶数为 $N = 2^{3n}$ 。我们对其做染色如下：

1. 大部分顶点都染成颜色 0。

2. 为了保证合法性，几个面、棱和顶点直接染成合法的颜色 (1/2/3)。

3. 由于顶点至多为 2 的图中只存在长链，因此我们让 123 三种颜色构成“管道”来模拟长链在单纯形内游走。在这三种颜色的“保护”之下，四种颜色同时出现在一个单纯形中的情况只能发生在管道断掉，也就是长链到达另一个端点的情况。

4. 我们将 G 的顶点放置在单纯形的一条边附近，将 G 的所有可能有序顶点对放置在另一条边附近。由于顶点对的总数不超过 N^2 ，所以可以放下。这种“放置”是用一条短管道表示的。之后，对于图 G 中的任意有向边 $i \rightarrow j$ ，令对应 i 的短管道的终点向对应 (i, j) 的短管道的起点连接一条长管道；再令对应 (i, j) 的短管道的终点向对应 j 的短管道的起点连接一条长管道。这样，图中的管道就确定地建模了 G 中的边。

一个直观示例如下图所示：

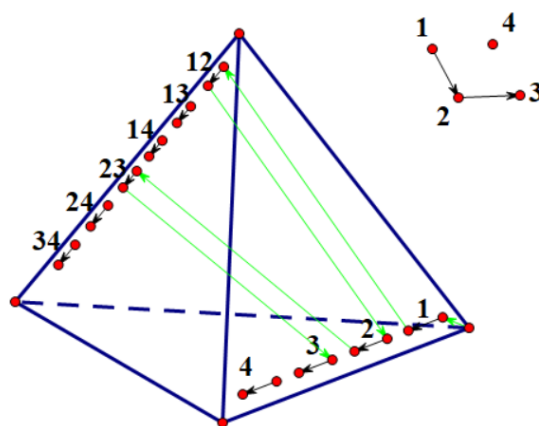


图 3: EOL 到 Sperner 的归约图示

□

Exercise 5.2. 1. 证明：Sperner 定理构造的路径移出边界时，只能从下边界移出。

2. 证明：Sperner 定理构造的路径相互不交叉。

3. 仿照 2 到 3 维的归纳方法，证明 Sperner 定理的高维情形。

¹⁰ [8] 证明了 2-Sperner 也是 PPAD 完全的，但是方法比较复杂)

5.3 PPAD 的完全问题：Brouwer

我们在最初证明纳什均衡存在性时就使用过 Brouwer 定理，这说明它和纳什均衡的结构很接近。这一节，我们先用 Sperner 定理证明 Brouwer 定理，明确它们之间的关联，然后再定义 Brouwer 的计算问题并给出 Sperner 到 Brouwer 的计算问题的归约，最终证明 Brouwer 的计算问题是 PPAD 完全的。

Theorem 5.5. (Brouwer 定理, 单纯形版本) $f: \Delta(x_0, \dots, x_m) \rightarrow \Delta(x_0, \dots, x_m)$ 连续, 则 f 必然存在一个不动点 x , 使得 $f(x) = x$.

证明. 考虑对 $\Delta(x_0, \dots, x_m)$ 的 i 阶标准剖分. 对任意 $i \in \{0, 1, \dots, m\}$, 记顶点 x_i 在单纯形中向对面的垂直投影为 l_i . 在每个标准剖分中, 为每个顶点染色, 并且对任意顶点 x , 要求其若染色为 i , 则 $x - f(x)$ 必须与 l_i 的内积是正数. 显然, 一个顶点不能染色当且仅当其为不动点; 并且这样的染色是合法的. 因此, 根据 Sperner 定理, 必定存在一个全色单纯形. 定义 i 阶单纯形的全色单纯形的重心为 p_i , 那么, 最终构成的序列 $\{p_i\}_{i=1}^\infty$ 根据 Bolzano-Weierstrass 定理, 必然收敛到某个定点 p^* . 根据收敛和全色单纯形的定义, 对任意 i , $(f(p^*) - p^*) \cdot l_i \geq 0$, 从而由 l_i 构成线性空间的一组基知必有 $f(p^*) = p^*$, 即得到一个不动点, 命题得证. \square

Brouwer 的计算问题的定义比较困难, 因为图灵机没有无限精度, 自然也不可能真正地表示一个连续函数. 但是, 不论是理论和实践中, 我们都往往并不需要完全精确地表示一个函数, 而只是调用它在某些特定位置的函数值, 甚至还会假设它具备若干附加的性质. 这就增加了近似或者简化表示的空间. 从 Brouwer 的问题来看, 计算问题和定理的区别就在于, 如果 f 可以取到任意的连续函数, 那么可能导致一些奇异行为的发生 (见本节习题), 影响实际的计算. 所以, 我们退而求其次, 求解一个近似问题. 由于近似有多种方式, 因此问题的参数会比较多.

Definition 5.2. (Brouwer 计算问题 $(\|\cdot\|, d, \mathcal{F}, \epsilon)$)

参数: 范数 $\|\cdot\|$, 维数 d , 函数 f 的性质 \mathcal{F} , 求解精度 ϵ (这些参数均可以与函数 f 的输入长度 n 相关)

输入一个满足性质 \mathcal{F} 的函数 $f: [0, 1]^d \rightarrow [0, 1]^d$

输出一个解 x , 使得 $\|f(x) - x\| \leq \epsilon$

例如, 该计算问题的一个实例是要求 \mathcal{F} 为多项式函数, 维数为 1, 范数取二范数, 求解精度为 $1/2^n$. 此时, 根据函数的这个性质, 我们可以要求输入为一个数组, 其中第 i 项代表 x^i 的系数. 这样, 问题的直接描述就是: 输入一个 n 阶多项式 $f(x) = \sum_{i=1}^n a_i x^i$, 输出一个解 x_0 , 使得 $|f(x_0) - x_0| \leq O(1/2^n)$. 那么, 一个自然的算法就是通过二分法求多项式函数 $g(x) = f(x) - x$ 的一个零点. 由于二分法的每次迭代耗时为常数, 可以缩短一半的区间长度, 因此, 在 $O(n)$ 时间内我们就能得到一个精度为 $O(1/n)$ 的解. 因此, 这个实例下问题属于 P 类.

当然, 复杂性理论的奥妙就在于寻找简单和复杂之间的恰当界限. 假如要求太高, 例如要求 $\epsilon = 0$, 那么就是多项式求精确解的问题, 通过伽罗瓦理论我们已经知道五次以上的方程都不存在代数数解, 因此即使解一定存在, 我们也不可能用有限的计算单元精确表示; 假如要求太低, 例如 ϵ 是常数, 那么这样定义出来的问题就没有任何价值. 现在, 我们的目标是证明 Brouwer 是 PPAD 完全的, 那就要求我们定义的复杂性类必须在简单和复杂之间寻找适当的平衡: 一方面足够复杂, 涵盖了足够规模的问题, 使得任何 EOL 的实例都能找到 Brouwer 中的对应; 另一方面又足够简单, 不至于出现要求的图灵机根本就不存在的情况.

对于 Brouwer 问题的各种近似情况已经有很多研究. 现在已知的结论有:

1. ([32]) 当 $d = 3$, \mathcal{F} 是 $O(1)$ -Lipschitz (即存在常数 C , 对任意的 x_1, x_2 , $\|f(x_1) - f(x_2)\| \leq C \cdot \|x_1, x_2\|$), $\|\cdot\|$ 是无穷范数, $\epsilon = 1/2^{O(n)}$ 时, Brouwer 的计算问题是 PPAD 完全的。
2. ([34]) 当 $d = O(n)$, \mathcal{F} 是 $O(1)$ -Lipschitz (即存在常数 C , 对任意的 x_1, x_2 , $\|f(x_1) - f(x__2)\| \leq C \cdot \|x_1, x_2\|$), $\|\cdot\|$ 是二范数, ϵ 是充分小的常数时, Brouwer 的计算问题是 PPAD 完全的。

下面, 我们简述结论 1 的证明思路。

首先是 Brouwer 属于 PPAD 的证明思路。根据 Theorem 5.3, 我们只需要构造一个 Brouwer 到 Sperner 的归约。事实上, 我们为函数 f 加入的 Lipschitz 连续性正是为了保证近似纳什均衡和全色单纯形的重心的距离比较接近。这样, 当阶数足够高时, 使用 Sperner 的计算问题找到的全色单纯形的坐标就是我们需要的近似纳什均衡。详细证明留作习题。

其次是 Brouwer 是 PPAD 困难的证明思路。我们只需要构造一个 EOL 到 Brouwer 的归约。归约的思路和 Sperner 比较就近, 实际上就是把任何一个 EOL 的实例转化成一个 Brouwer 函数。只不过, 在 Sperner 问题中, 这种转化是构造一个离散顶点的染色, 而在 Brouwer 中则是构造一个满足 Brouwer 条件的连续函数形成的向量场。由于 Brouwer 条件比较复杂, 这种构造相对更难一点, 但是总体而言仍然可以用离散的办法解决, 即构造一个线性插值函数, 其在网格顶点上的函数值由类似 Sperner 的染色模拟给出 (例如, 不同颜色对应不同 $f(x) - x$ 的方向, 即场的方向), 而在网格内部的函数值由该网格的顶点插值给出。这样, 我们可以用比较少的信息量表示出这个函数, 但同时又满足了模拟 EOL 问题的需求。(思路来源: [13])

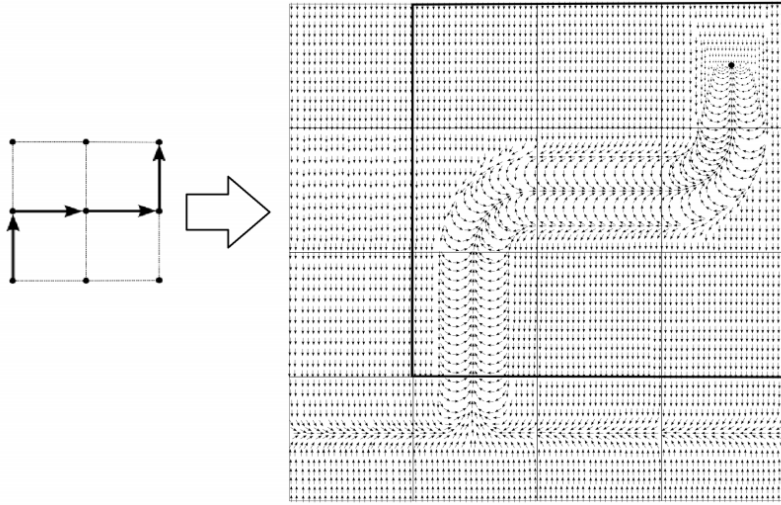


图 4: EOL 到 Brouwer 的归约图示 ([33], Section 3.1.2)

- Exercise 5.3.**
1. 一般的 Brouwer 不动点定理只适用于凸紧集上的连续函数 f 。试给出非凸和非紧的集合上 Brouwer 定理不成立的反例。
 2. 对于 Brouwer 定理的证明, 举例说明, p^* 可能并不属于任何一个重心收敛到它的一系列 i 阶全色单纯形。
 3. 补充 Brouwer 定理的证明细节: 在 $d = 2$, 其它条件与结论 1 相同的情况下, 选取恰当的阶数, 构造 Brouwer 到 Sperner 的归约。有兴趣的读者可以从构造恰当的标准剖分入手, 考虑 $d=3$ 甚至更高维的情况。提示已在证明部分给出。

4. 参考讲义提供的思路和4的图示, 构造一个从顶点数量为 $O(n)$ 的 EOL 问题到 $Brouwer(\|\cdot\|_2, d = 2, \mathcal{F} = O(1)\text{-Lipschitz}$ 连续函数, $\epsilon = 1/n$) 的归约

5.4 纳什均衡是 PPAD 完全的

此前, 为了行文的连贯性, 我们模糊了对纳什均衡的计算问题的定义。因此, 我们首先要明确这个概念。由于 Lemke Howson 算法给出的是精确纳什均衡的求解, 定理5.1实际上只说明了精确纳什均衡的求解属于 PPAD。但是, 即使我们证明了精确纳什均衡的求解问题是 PPAD 困难的, 其价值也比较有限。因为人们在实践中往往是采用虚拟博弈等近似算法来求解纳什均衡的, 即使使用 Lemke Howson 算法求精确解也不可避免地会有舍入误差的问题。¹¹ 因此, 如果我们能够证明近似纳什均衡的求解也是 PPAD 完全的, 那么就有足够的理由说明纳什均衡是“难解”的。

从近似纳什均衡的定义出发, 我们直接定义计算问题如下:

Definition 5.3. (双人博弈的 ϵ -近似纳什均衡的计算问题) 对于双人博弈, 输入两个玩家的 $n \times n$ 大小的支付矩阵, 输出一个 ϵ -近似纳什均衡。(可以和 n 有关)

历史上, Daskalakis, Goldenberg 和 Papadimitriou 在 2006 年首先给出了三人以上博弈的精确纳什均衡计算是 PPAD 完全的证明 [13], 同年, 陈汐、邓小铁和滕尚华证明了双人博弈的精确纳什均衡计算是 PPAD 完全的 [8], 随后, 他们将证明改进, 证明了 $\epsilon = 1/n^{O(1)}$ 时的近似纳什均衡计算亦是 PPAD 完全的。[9] [8]

下面, 我们简述 [13] 中证明三人博弈的精确纳什均衡计算是 PPAD 完全的思路。三人博弈的定义与双人博弈类似, 只是三个玩家现在有三个支付矩阵, 且每个矩阵的维数为 $n * n * n$, 标志着每个玩家分别选取第 i, j, k 个策略时各自的收益。根据上一节的证明, 我们只需要把恰当精度的 PPAD 困难的 Brouwer 计算问题归约为三人博弈的纳什均衡的计算问题。这种归约实际上分为两步, 第一步是把恰当的 Brouwer 的计算问题归约到三人的图博弈上, 第二步是把三人的图博弈归约到双人博弈的计算问题上。这里涉及到一个新的概念, 即图博弈, 引入它主要是为了使用其图结构来模拟 Brouwer 计算问题中涉及的算术电路。[8] 中双人博弈的证明则采用了类似的思路, 只是将中间过渡的三人图博弈改为了三人二部图博弈。

首先, 我们需要叙述这里使用到的 Brouwer 计算问题: 这里的计算问题比较特殊, 为了模拟离散化的情况, 与 Sperner 的计算问题很类似。

取 $d = 3$, 该 Brouwer 问题的函数 f 的函数值由 $[0, 1]^2$ 中的 2^{3n} 个小立方体 $K_{ijk} = \{(x, y, z) : i \cdot 2^{-n} \leq x \leq (i+1) \cdot 2^{-n}, j \cdot 2^{-n} \leq y \leq (j+1) \cdot 2^{-n}, k \cdot 2^{-n} \leq z \leq (k+1) \cdot 2^{-n}\}$ 的中心决定。对每个中心 c_{ijk} , 要求 $f(c_{ijk}) = c_{ijk} + \delta$, 其中 $\delta \in \{\delta_1 = (\alpha, 0, 0), \delta_2 = (0, \alpha, 0), \delta_3 = (0, 0, \alpha), \delta_0 = (-\alpha, -\alpha, -\alpha)\}, \alpha = 2^{-2n}$ 。为了保证 f 不越过边界, 要求其满足类似 Sperner 的边界条件。其余点处 f 的函数值由线性插值定义。

这样, 可以要求 Brouwer 的计算问题的输入为一个算术电路 C^{12} : 向 C 中输入中心点的坐标 ($3n$ 个比特), 输出 $0/1/2/3$ (2 个比特) 标志 δ 的方向; 输出为某个中心点的坐标 ($3n$ 个比特), 使得其为某个四色单纯形的顶点。根据我们在上一节的讨论, Brouwer 的不动点必然出现在四色单纯形附近, 因此这个输出同时也是不动点问题的一个比较好的近似解。

¹¹即使所有玩家的支付都是有理数, 纳什 [29] 构造的反例表明, 三人博弈的纳什均衡也可能包含无理数; 而双人博弈的纳什均衡必定是有理数

¹²算术电路即利用可以表示任意的布尔函数

由于该计算问题与 Sperner 的相似性，不难仿照 Sperner 的计算问题给出 EOL 问题到该问题的归约，从而表明其为 PPAD 困难的。下面，我们完成后半部分的证明。

5.4.1 用图博弈模拟算术电路

算术电路 (arithmetic circuit) 即节点进行算术运算 (加减乘除) 构成的有向无环图，由于与电路的基本结构相似，所以称为算术电路，其可以表示任意布尔函数。因此，为了将 Brouwer 的计算问题中输入的电路转化为博弈问题，一个自然的想法就是通过具有某种图结构的博弈来模拟算术电路的构造。

具体而言，图博弈是一种特殊的博弈。标准形式的博弈中，一个玩家的支付与所有玩家的策略都相关联；而在图博弈中，所有玩家对应于一个有向图上的顶点，而每个玩家的支付仅仅与他的前驱玩家有关联，构成一个局部的标准形式博弈。这种博弈在实际中也很常见，例如某些公司的组织架构中每个人的收益可能仅仅与下属或上级有关，从而导出一个树状结构的图博弈。可以证明，任何一种图博弈都可以多项式时间内归约为度数为 3，每个玩家的策略数量为 2 的博弈。因此，我们之后的讨论都基于这种标准形式的图博弈。

下面，我们讨论如何利用图博弈来模拟算术电路。基本思路是：利用标准图博弈的纳什均衡中玩家 v 选择策略 0 的概率 $\mathbf{p}[v] \in [0, 1]$ 来模拟算术电路的计算过程。

Example 5.1. 下面列举的博弈 $\mathcal{G}\mathcal{G}$ 模拟了算术运算 $x \rightarrow \alpha x$ (数乘)：

该博弈涉及三个玩家 v_1, v_2, w 。 α 为一非负实数，定义玩家 v_2 和 w 的支付如下：

		w plays 0	w plays 1
v_2 :	v_2 plays 0	0	1
	v_2 plays 1	1	0

		v_2 plays 0	v_2 plays 1
w :	w plays 0	0	0
	w plays 1	α	α

		v_2 plays 0	v_2 plays 1
v_1 :	v_1 plays 0	0	1
	v_1 plays 1	0	1

其对应的图结构如5所示。那么，对任意 $\epsilon < 1$ ，博弈 $\mathcal{G}\mathcal{G}$ 的任意一个 ϵ -近似纳什均衡 \mathbf{p} ($\mathbf{p}[v]$ 定义为玩家 v 选择策略 0 的概率) 满足条件： $\mathbf{p}[v_2] = \min(\alpha \mathbf{p}[v_1], 1) \pm \epsilon$ ($a = b \pm \epsilon$ 即 $b - \epsilon \leq a \leq b + \epsilon$)

证明留作习题。

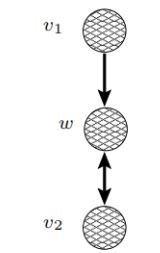


FIG. 3. $\mathcal{G}_{x\alpha}, \mathcal{G}_{=}$.

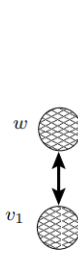


FIG. 4. \mathcal{G}_{α} .

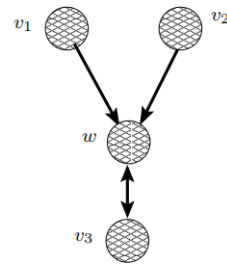


FIG. 5. $\mathcal{G}_{+}, \mathcal{G}_{*}, \mathcal{G}_{-}$.

图 5: 图博弈的

类似地，如5部分所示，我们可以构造模拟其它算术电路的图博弈 \mathcal{G}_{α} (常数)， \mathcal{G}_{+} (加法)， \mathcal{G}_{-} (减法)， $\mathcal{G}_{=}$ (复制)， \mathcal{G}_{\times} (点乘)， \mathcal{G}_{max} (求最大值)，将这些模拟算术电路的博弈元件组合起来，我

们就能高效地模拟算术电路，从而，构造算术电路到图博弈的纳什均衡的归约。

5.4.2 图博弈到标准形式博弈的归约

首先，我们证明，顶点的最大度数 d ，玩家的策略数量至多为 t 的图博弈的均衡计算问题可以多项式时间内归约为 $d^2 + 1$ 个玩家的标准形式博弈的均衡计算问题。为了方便表示，该图博弈中玩家 v 的策略分别记为 $(v, a), 1 \leq a \leq t$ ，其相邻的玩家集合记为 $\mathcal{N}(v)$ 。

1. 对 \mathcal{GG} 中的支付的分量做放缩，使得其落在区间 $[0, 1]$ 内。
2. 选取 r 为 $d^2 + 2$ 或 $d^2 + 1$ 中的偶数
3. 为图博弈中的玩家集合 V 做 r 染色，不妨设颜色集为 $\{1, 2, \dots, r\}$ ，使得相邻的顶点颜色互异，并且具有相同前驱的顶点也颜色互异。这样的染色显然存在且容易计算，因为任意一个顶点的颜色至多与 d^2 个顶点冲突，因此总能找到一个不冲突的颜色染上。不妨通过补充无用染色节点假设染每种颜色的顶点数量相同。记 $\{v_1^{(i)}, \dots, v_{n/r}^{(i)}\}$ 为染第 i 种颜色的顶点集合。
4. 对任意 $1 \leq p \leq r$ ，我们构造博弈 \mathcal{G} 的第 p 个玩家的策略集合 $S_p = \bigcup_{j=1}^{n/r} \bigcup_{a=1}^t (v_j^p, a)$ ，从而 $|S_p| = t \frac{n}{r}$
5. 记 S 为 $S_p, p \in [r]$ 的笛卡尔积。现在，我们还需要构造该博弈的支付矩阵，即对玩家 p 和博弈 \mathcal{G} 的任意一个纯策略组 $s \in S$ 的支付 u_s^p 。
 - (a) 一开始，设定所有支付为 0
 - (b) 之后，考虑任意顶点 $v_0 \in V$ 及其在博弈 \mathcal{GG} 中的异色前驱 $v_1, \dots, v_{d'}$ 。假设 v_0 的颜色为 p 。并且，如果对任意 $0 \leq i \leq d'$ ， s 包含 (v_i, a_i) ，则 $u_s^p = u_{s'}^{v_0}$ ，即 \mathcal{GG} 中玩家 v_0 在相邻玩家 $v_i, i \in [p]$ 分别使用策略 a_i 时得到的策略 s' 的效用。
 - (c) 取 $M > 2 \frac{n}{r}$ ，对奇数 $p < r$ ，对任意 i, a, a' ，如果玩家 p 使用策略 $(v_i^{(p)}, a)$ ， $p+1$ 使用策略 $(v_i^{(p+1)}, a')$ ，则将 u_s^{p+1} 减去 M 。

那么，根据该构造，我们有如下结论：

Theorem 5.6. 从博弈 \mathcal{G} 的任意纳什均衡 $N_{\mathcal{G}} = \left\{ x_{(v,a)}^p \right\}_{p,v,a}$ ，可以在多项式时间内恢复出 \mathcal{GG} 的一个纳什均衡 $\{x_a^v\}_{v,a}$ 如下：

$$x_a^v := x_{(v,a)}^{c(v)} / \sum_{j \in S_v} x_{(v,j)}^{c(v)} \quad \forall a \in S_v, v \in V.$$

证明. 构造和证明的基本思路是让博弈 \mathcal{G} 中奇数编号和偶数编号的玩家两两配对，构成一个局部的“猜硬币”博弈，然后对其做微扰。双人“猜硬币”博弈定义为当且仅当两个玩家的策略相同时，玩家 1 获得支付 u ，玩家 2 获得 $-u$ ，其余情况则支付均为 0。显然，其有且仅有唯一的纳什均衡：两个玩家都选择均匀分布。

首先，我们证明得到的确实是一个“微扰”，也就是各个玩家的均衡策略不会偏离均匀分布太远。注意到，博弈 \mathcal{G} 中玩家 p 的任意一个策略对应的是 \mathcal{GG} 中的二元组 (v, a) ，其中 v 为玩家，颜色为 p ， $a \in [t]$ 为玩家 v 的一个策略。因此，当玩家 p 以均匀分布选取策略时，对应 \mathcal{GG} 中的玩家恰好为 v 的概率为 r/n 。可以证明： \mathcal{G} 的纳什均衡中选择 v 对应的策略的概率属于 $[\frac{r}{n} - \frac{1}{M}, \frac{r}{n} + \frac{1}{M}]$ 。原理在于假如概率偏离均匀分布太多，那么可以推测出有某一对玩家根本不出现在均衡之中。

其次，我们考虑命题中定义的恢复过程，实际上就是设在 \mathcal{G} 中玩家 p 使用策略 v 的概率中 v 的各个策略 a 的占比与 \mathcal{GG} 的某个纳什均衡相同。事实上，对策略 \mathcal{G} 的任意一个纳什均衡，根据纳什均衡的定义，对任意 $a, a' \in [t]$ ，有：

$$\mathbb{E}[\text{玩家 } p \text{ 使用策略 } (v, a)] > \mathbb{E}[\text{玩家 } p \text{ 使用策略 } (v, a')] \Rightarrow x_{(v, a')}^p = 0$$

但是，根据我们之前的构造：

$$\begin{aligned} \text{左式} &= M \cdot \Pr\left(\text{玩家 } p+1 \text{ 使用策略 } v_i^{(p+1)}\right) \\ &+ \sum_{s \in S_{\mathcal{N}(v) \setminus \{v\}}} u_{as}^v \prod_{u \in \mathcal{N}(v) \setminus \{v\}} x_{(u, s_u)}^{c(u)} \end{aligned} \quad (5.3)$$

对 a' 同理。因此，这表明：

$$\sum_{s \in S_{\mathcal{N}(v) \setminus \{v\}}} u_{as}^v \prod_{u \in \mathcal{N}(v) \setminus \{v\}} x_{(u, s_u)}^{c(u)} > \sum_{s \in S_{\mathcal{N}(v) \setminus \{v\}}} u_{a's}^v \prod_{u \in \mathcal{N}(v) \setminus \{v\}} x_{(u, s_u)}^{c(u)} \Rightarrow x_{(v, a')}^p = 0.$$

注意到， $\prod_{u \in \mathcal{N}(v) \setminus \{v\}} \sum_{j \in S_u} x_{(u, j)}^{c(u)} = \prod_{u \in \mathcal{N}(v) \setminus \{v\}} \Pr(c(u) \text{ plays } u) = p(v)$ ，并且根据之前的微扰结论， $p(v) \geq (\lambda - \frac{1}{M})^d > 0$ ，因此，两边同时除以上式即有：

$$\sum_{s \in S_{\mathcal{N}(v) \setminus \{v\}}} u_{as}^v \prod_{u \in \mathcal{N}(v) \setminus \{v\}} x_{s_u}^u > \sum_{s \in S_{\mathcal{N}(v) \setminus \{v\}}} u_{a's}^v \prod_{u \in \mathcal{N}(v) \setminus \{v\}} x_{s_u}^u \Rightarrow x_{a'}^v = 0$$

而这恰恰是 $\{x_a^v\}_{v,a}$ 构成 \mathcal{GG} 的充要条件。从而，我们导出了从 \mathcal{G} 的纳什均衡到 \mathcal{GG} 的纳什均衡的一个满射。 \square

最后，我们只需要给出 $n \geq 3$ 人标准形式博弈到三人标准形式博弈的多项式归约。实际的证明中使用的是类似上面证明的技巧，只是包含了更多技术细节，因此这里我们略过。总体而言，证明分为两步，第一步是将 $n \geq 3$ 人的标准形式博弈归约为三人，每个人至多两个策略的图博弈，第二步是将该图博弈归约为三人标准形式博弈。

综上，我们最终可以证明三人博弈的精确纳什均衡的计算问题是 PPAD 完全的。

Exercise 5.4. 1. 证明例 5.1 的结论。

2. 参考 5 右图的节点构造图示，构造一个图博弈 \mathcal{GG} ，其任意 $\epsilon < 1$ 的纳什均衡中，玩家 v_1, v_2, v_3 选择策略 0 的概率满足： $\mathbf{p}[v_3] = \min(\alpha \mathbf{p}[v_1] + \beta \mathbf{p}[v_2] + \gamma \mathbf{p}[v_1] \mathbf{p}[v_2], 1) \pm \epsilon$ 。
3. 使用 Section 5.4.2 的方法，将例 5.1 中的图博弈的均衡计算问题归约为标准形式博弈的均衡计算问题。

6 总结与展望：纳什均衡计算的发展历程

本讲义基本上是按照纳什均衡计算问题的发展历史编排的。

1928 年，Von Neumann 在 [30] 中提出了双人零和博弈的均衡概念，并且列举了一些特殊的例子。参考讲义的 Section 2.1。

很快，人们提出了零和博弈的多项式算法。1951 年，Dantzig 在 [12] 中提出了双人零和博弈的线性规划算法。参考讲义的 Section 2.2.1；同年，Brown 在 [6] 中提出了双人零和博弈的另一个算法，即虚拟博弈算法；1957 年，Hannan[23] 将其改进为光滑虚拟博弈算法。本讲义简单介绍了预测论的基本思想以解释虚拟博弈算法的本质，参考 Section 2.2.2。

1951 年, 纳什在 [29] 中提出了一般标准形式博弈的纳什均衡概念, 并利用不动点定理证明了其存在性, 纳什均衡的概念随即在之后的几十年里逐渐渗透到经济学的方方面面。1964 年, Lemke 和 Howson[7] 提出了双人博弈的纳什均衡的 Lemke-Howson 算法, 参考Section 4.3, 后来被人们推广到多人博弈中。后来, 人们又提出了若干纳什均衡的精确算法, 如基于在包含纳什均衡的子单纯形分划 (类似 Sperner 分划) 上的游走的近似算法 ([36] (1967), [19] (1973), [40] (1979), [41] (1982)), 基于纳什均衡代数性质的量词消去算法 [26] (2004)。2003 年, Lipton, Markakis 和 Mehta[27] 提出了双人博弈的近似纳什均衡的 LMM 算法, 表明了如果只要求常数近似的话纳什均衡存在拟多项式复杂度的高效算法, 参考Section 4.2。但是, 与零和博弈不同, 上述的所有算法都无法证明是多项式时间的。

1971 年, Cook, Levin[11][25] 建立了复杂性的完全性理论, 为人们提供了研究计算复杂度的有力的理论工具。于是, 人们开始猜测纳什均衡的计算问题是 NP 困难的。之后的一系列研究为人们深入理解纳什均衡的计算复杂性类打下了基础。

1991 年, Megiddo 和 Papadimitriou[28] 定义了 TFNP 和 PLS 复杂性类, 并且证明了 Sperner、Brouwer、纳什均衡等计算问题属于 TFNP。但是, 他们也注意到, TFNP 问题是语义类, 并不存在完全性问题, 因此为进一步研究带来了困难。1994 年, Papadimitriou[32] 借助证明模式提出了 PPAD 复杂性类, 打破了语义类的限制, 给出了它的一个自然的完全问题 EOL, 参考Section 5.1; 他还进一步证明了 Sperner 和 Brouwer 的计算问题是 PPAD 完全的, 并且利用 Lemke-Howson 算法证明了双人博弈的精确纳什均衡计算问题属于 PPAD, 参考Section 5.2和Section 5.3。

最终, 2006 年, Dasklakis, Goldenberg, Papadimitriou; 陈汐、邓小铁、滕尚华的工作 [13],[8],[9] 填补了纳什均衡计算问题的最后一块拼图, 最终证明了即使是双人博弈, 近似比为 $1/\text{poly}(n)$ 的情况下, 纳什均衡的计算问题也是 PPAD 完全的, 从而暗示了纳什均衡是多项式时间难解的。

不过, 纳什均衡的计算理论研究还远远没有终结。我们在下面列举若干当前的研究方向:

1. 提出新的纳什均衡的精确/近似算法:

(a) 类似 LLM 的拟多项式纳什均衡算法:

聚落检测方法 [2] (2012), 自由博弈 [1] (2014), 约束满足问题的稠密子图方法 [4] (2015), Dughmi 信号方法 [10] (2015) 等

(b) 常数近似度的纳什均衡的多项式时间算法:

0.75[24] (2006), 0.5[14] (2006), 0.38[15] (2007), 0.36[5] (2007), 0.3393[39] (2007), $1/3$ [17] (2022)

2. 证明现存的纳什均衡算法的困难性:

- 虚拟博弈 (非光滑) 算法的指数时间复杂性 [16] (2014), 参考 Remark 2.3
- Lemke-Howson 算法的指数时间复杂性 [35] (2004)
- 基于同伦路径的移動的任何算法 (包括 Lemke-Howson 算法) 都是 PSPACE 完全的 [22]

3. 纳什均衡相关的其他计算理论:

- (a) 如果假设 EOL 问题具有指数时间复杂度 ($2^{\Omega(n)}$), 那么存在常数 c , 使得任意 $\epsilon < c$, ϵ -纳什均衡的计算不可能存在多项式时间算法 (利用了 PCP 定理) [34] (2016)

(b) 双人博弈的纳什均衡的通信复杂性（两个玩家之间为了计算纳什均衡至少需要交换的比特数量）为 $\Omega(n^{O(1)})$ （ n 为玩家的策略数量） [3]（2017）

(c) 纳什均衡计算的另一种刻画：

PPAD 实际上是刻画了逼近一个纳什均衡在组合上的困难程度。从不动点的角度看，其目标是求解一个点 x ， $|f(x) - x| \leq \epsilon$ ；而我们可以定义 FIXP 复杂性类，其刻画了纳什均衡在分析上的困难程度。从不动点的角度看，其目标是求解一个点 x ， $|x - x^*| \leq \epsilon$ ，其中 $f(x^*) = x^*$ 。[18]（2007）从上面不动点的概念出发定义了 FIXP 复杂性类，并证明了纳什均衡的求解是 FIXP 困难的。

(d) 纳什均衡的平滑分析：

[38]（2004）证明了纳什均衡的计算在约束的数量为 n ，向量空间为 d 的二阶段投影顶点方法在尺度为 σ 的扰动下复杂度是 $\text{poly}(n, d, 1/\sigma)$ 的。参考 Remark 4.2.

参考文献

- [1] Scott Aaronson, Russell Impagliazzo, and Dana Moshkovitz. AM with multiple merlins. In *IEEE 29th Conference on Computational Complexity, CCC 2014, Vancouver, BC, Canada, June 11-13, 2014*, pages 44–55. IEEE Computer Society, 2014. 1a
- [2] Sanjeev Arora, Rong Ge, Sushant Sachdeva, and Grant Schoenebeck. Finding overlapping communities in social networks: toward a rigorous approach. In Boi Faltings, Kevin Leyton-Brown, and Panos Ipeirotis, editors, *Proceedings of the 13th ACM Conference on Electronic Commerce, EC 2012, Valencia, Spain, June 4-8, 2012*, pages 37–54. ACM, 2012. 1a
- [3] Yakov Babichenko and Aviad Rubinfeld. Communication complexity of approximate nash equilibria. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 878–889. ACM, 2017. 3b
- [4] Siddharth Barman. Approximating nash equilibria and dense bipartite subgraphs via an approximate version of caratheodory’s theorem. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 361–369. ACM, 2015. 1a
- [5] Hartwig Bosse, Jaroslaw Byrka, and Evangelos Markakis. New algorithms for approximate nash equilibria in bimatrix games. In Xiaotie Deng and Fan Chung Graham, editors, *Internet and Network Economics, Third International Workshop, WINE 2007, December 12-14, Proceedings*, volume 4858 of *Lecture Notes in Computer Science*, pages 17–29, San Diego, CA, USA, 2007. Springer. 1b
- [6] George W. Brown. Iterative solution of games by fictitious play. *Act. Anal. Prod Allocation*, 1951. 6
- [7] Lemke C.E. and Howson J. T. Equilibrium points of bimatrix games. *Journal of the Society for industrial and Applied Mathematics*, 12(2):413–423, 1964. 6

- [8] Xi Chen and Xiaotie Deng. Settling the complexity of two-player nash equilibrium. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 261–272. IEEE Computer Society, 2006. 10, 5.4, 6
- [9] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Computing nash equilibria: Approximation and smoothed complexity. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 603–612. IEEE Computer Society, 2006. 5.4, 6
- [10] Yu Cheng, Ho Yee Cheung, Shaddin Dughmi, Ehsan Emamjomeh-Zadeh, Li Han, and Shang-Hua Teng. Mixture selection, mechanism design, and signaling. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1426–1445. IEEE Computer Society, 2015. 1a
- [11] Stephen A. Cook. The complexity of theorem-proving procedures. In Michael A. Harrison, Ranan B. Banerji, and Jeffrey D. Ullman, editors, *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, May 3-5, 1971, Shaker Heights, Ohio, USA*, pages 151–158. ACM, 1971. 6
- [12] George B Dantzig. A proof of the equivalence of the programming problem and the game problem. *Activity analysis of production and allocation*, 13, 1951. 6
- [13] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a nash equilibrium. In Jon M. Kleinberg, editor, *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*, pages 71–78. ACM, 2006. 5.3, 5.4, 6
- [14] Constantinos Daskalakis, Aranyak Mehta, and Christos H. Papadimitriou. A note on approximate nash equilibria. In Paul G. Spirakis, Marios Mavronicolas, and Spyros C. Kontogiannis, editors, *Internet and Network Economics, Second International Workshop, WINE 2006, December 15-17, Proceedings*, volume 4286 of *Lecture Notes in Computer Science*, pages 297–306, Patras, Greece, 2006. Springer. 1b
- [15] Constantinos Daskalakis, Aranyak Mehta, and Christos H. Papadimitriou. Progress in approximate nash equilibria. In Jeffrey K. MacKie-Mason, David C. Parkes, and Paul Resnick, editors, *Proceedings 8th ACM Conference on Electronic Commerce (EC-2007), June 11-15*, pages 355–358, San Diego, California, USA, 2007. ACM. 1b
- [16] Constantinos Daskalakis and Qinxuan Pan. A counter-example to karlin’s strong conjecture for fictitious play. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 11–20. IEEE Computer Society, 2014. 2.3, 2
- [17] Argyrios Deligkas, Michail Fasoulakis, and Evangelos Markakis. A polynomial-time algorithm for 1/3-approximate nash equilibria in bimatrix games. In Shiri Chechik, Gonzalo Navarro, Eva Rotenberg, and Grzegorz Herman, editors, *30th Annual European Symposium on Algorithms, ESA 2022, September 5-9*, volume 244 of *LIPICs*, pages 41:1–41:14, Berlin/Potsdam, Germany, 2022. Schloss Dagstuhl - Leibniz-Zentrum für Informatik. 1b

- [18] Kousha Etessami and Mihalis Yannakakis. On the complexity of nash equilibria and other fixed points (extended abstract). In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pages 113–123. IEEE Computer Society, 2007. [3c](#)
- [19] Crisostomo B Garcia, Carlton E Lemke, and H Luethi. Simplicial approximation of an equilibrium point for non-cooperative n-person games. In *Mathematical programming*, pages 227–260. Elsevier, 1973. [6](#)
- [20] John Geanakoplos. Nash and walras equilibrium via brouwer. *Economic Theory*, 21:585–603, 2003. [3](#)
- [21] Paul W. Goldberg and Christos H. Papadimitriou. Towards a unified complexity theory of total functions. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, volume 94 of *LIPICs*, pages 37:1–37:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. [5.1](#)
- [22] Paul W. Goldberg, Christos H. Papadimitriou, and Rahul Savani. The complexity of the homotopy method, equilibrium selection, and lemke-howson solutions. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 67–76. IEEE Computer Society, 2011. [2](#)
- [23] J. Hannan. Approximation to bayes risk in repeated play. *Contributions to the Theory of Games*, 3:97–139, 1957. [6](#)
- [24] Spyros C. Kontogiannis, Panagiota N. Panagopoulou, and Paul G. Spirakis. Polynomial algorithms for approximating nash equilibria of bimatrix games. In Paul G. Spirakis, Marios Mavronicolas, and Spyros C. Kontogiannis, editors, *Internet and Network Economics, Second International Workshop, WINE 2006, December 15-17, Proceedings*, volume 4286 of *Lecture Notes in Computer Science*, pages 286–296, Patras, Greece, 2006. Springer. [1b](#)
- [25] Leonid Anatolevich Levin. Universal sequential search problems. *Problemy peredachi informatsii*, 9(3):115–116, 1973. [6](#)
- [26] Richard J Lipton and Evangelos Markakis. Nash equilibria via polynomial equations. In *LATIN 2004: Theoretical Informatics: 6th Latin American Symposium, Buenos Aires, Argentina, April 5-8, 2004. Proceedings 6*, pages 413–422. Springer, 2004. [6](#)
- [27] Richard J. Lipton, Evangelos Markakis, and Aranyak Mehta. Playing large games using simple strategies. In Daniel A. Menascé and Noam Nisan, editors, *Proceedings 4th ACM Conference on Electronic Commerce (EC-2003), San Diego, California, USA, June 9-12, 2003*, pages 36–41. ACM, 2003. [4.2](#), [6](#)
- [28] Nimrod Megiddo and Christos H. Papadimitriou. On total functions, existence theorems and computational complexity. *Theor. Comput. Sci.*, 81(2):317–324, 1991. [6](#)
- [29] John Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951. [11](#), [6](#)
- [30] J. von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100:295–320, 1928. [6](#)

- [31] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. *Algorithmic game theory*. Cambridge university press, 2007. 5
- [32] Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *J. Comput. Syst. Sci.*, 48(3):498–532, 1994. 5.1, 9, 1, 6
- [33] Tim Roughgarden. Complexity theory, game theory, and economics: The barbados lectures. *Found. Trends Theor. Comput. Sci.*, 14(3-4):222–407, 2020. 8, 4
- [34] Aviad Rubinstein. Settling the complexity of computing approximate two-player nash equilibria. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 258–265. IEEE Computer Society, 2016. 2, 3a
- [35] Rahul Savani and Bernhard von Stengel. Exponentially many steps for finding a nash equilibrium in a bimatrix game. In *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*, pages 258–267. IEEE Computer Society, 2004. 2
- [36] Herbert Scarf. The approximation of fixed points of a continuous mapping. *SIAM Journal on Applied Mathematics*, 15(5):1328–1343, 1967. 6
- [37] Emanuel Sperner. Ein satz über untermengen einer endlichen menge. *Mathematische Zeitschrift*, 27(1):544–548, 1928. 5.2
- [38] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *J. ACM*, 51(3):385–463, 2004. 4.2, 3d
- [39] Haralampos Tsaknakis and Paul G. Spirakis. An optimization approach for approximate nash equilibria. In Xiaotie Deng and Fan Chung Graham, editors, *Internet and Network Economics, Third International Workshop, WINE 2007, December 12-14, Proceedings*, volume 4858 of *Lecture Notes in Computer Science*, pages 42–56, San Diego, CA, USA, 2007. Springer. 1b
- [40] Gerard van der Laan and Adolphus Johannes Jan Talman. A restart algorithm for computing fixed points without an extra dimension. *Mathematical Programming*, 17:74–84, 1979. 6
- [41] Gerard van der Laan and Adrian JJ Talman. On the computation of fixed points in the product space of unit simplices and an application to noncooperative n person games. *Mathematics of Operations Research*, 7(1):1–13, 1982. 6