中国科学院大学 · 人工智能学院 · 本科生专业课

博弈论

上课时间：每周一晚上18:10-19:50

上课地点：玉泉路校区 教学楼阶一1

授课团队：兴军亮（教师）、徐航（助教）

上一讲内容回顾：算法博弈论（1）

知识点

- **算法博弈论研究内容**
- **算法化机制设计概念**
- **拍卖及完美拍卖**
- **完美拍卖应用案例**

能力线

- **算法博弈论**：博弈论+算法，学科交叉能力
- **机制化设计**：如何设计一个好的制度？

价值面

- **算法博弈论启示**：

① 要知其然、知其所以然，还要知其何以做

② 学会正向和逆向博弈

第七讲：算法博弈论（2）

兴军亮

授课时间：2024年4月8日

联系方式：jlxing@nlpr.ia.ac.cn

第七讲：算法博弈论（2）

内容提纲

1. 自私路由的低效率性
2. 自私路由的改进措施
3. 不同类型的均衡求解
4. 均衡计算复杂度分析

第七讲：算法博弈论（2）

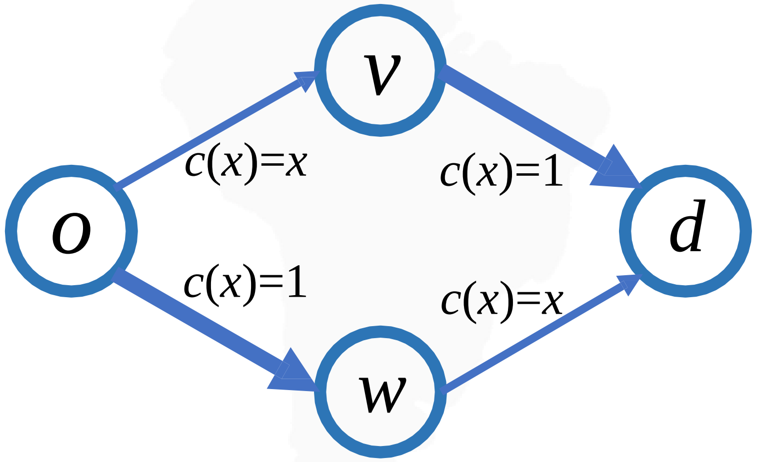
内容提纲

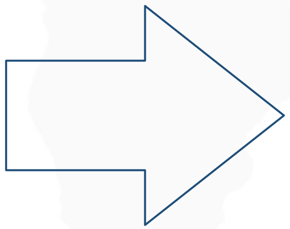
自私路由（Selfish Routing）

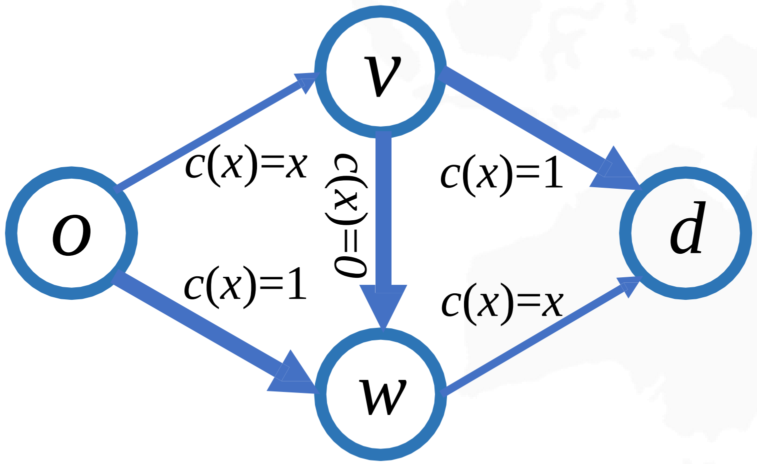
- **自私路由**：最初由计算机科学家蒂姆·拉夫加登（Tim Roughgarden）提出。每个人在道路网络中的移动方式在自己看来是最佳的（“用户优先”），不过大家的整体行为对道路网络产生的影响却可能是最差的（未达到“系统优先”）。
- **约翰·纳什**（John Nash）：“在一个国家中，没有任何一个实验性游戏的参与者可以仅凭自己的努力来改善状况。具有讽刺意味的是，如果大家所做的事情对自己最有利，他们的做法就不会对每个人都有益。”
- 每个人都是“自私的出行人”，纳什均衡会陷入局部最优陷阱，没有达到系统最优。这一现象称为**均衡的低效率性**。

布雷斯悖论详细讲解

- **布雷斯悖论**（名字来自德国数学家迪特里希·布雷斯）指在一个交通网络上增加一条路段反而使网络上的旅行时间增加；这一附加路段不但没有减少交通延滞，反而降低了整个交通网络的服务水准，这种出力不讨好且与人们直观感受相背的交通网络现象主要源于“纳什均衡点并不一定是社会最优化”这一现象。

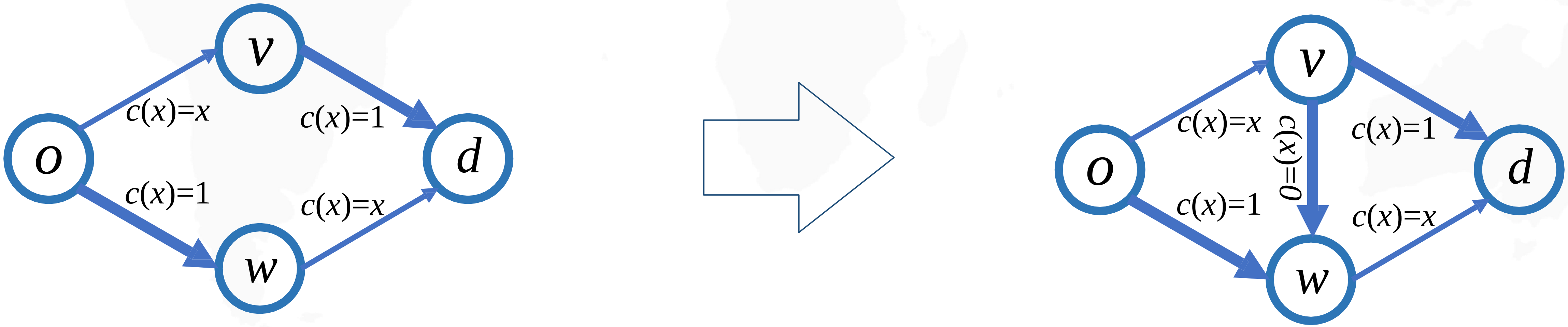






布雷斯悖论详细讲解

- 符号含义：o为起点，d为终点。x代表路径上的流量比例($x \in [0,1]$)。$c(\cdot)$为路径的代价函数，即通过时间。
- 在原始网络（左图）中，o-v-d 和 o-w-d路径代价函数完全相同，50%司机选择o-v-d，剩下50%选择o-w-d，最终期望代价为1.5。
- 在扩展网络（右图）中，加入一条通过时间近于0的v-w，此时所有司机都会选择o-v-w-d，此时最终期望代价为2.0。



无秩序代价（Price of Anarchy，PoA）

- **无秩序代价**（Price of Anarchy，PoA）：博弈论中用于评测在一个系统中，因为其参与单位的利己行为（或自私行为）而导致的效率下降程度。PoA值越大，说明均衡的效率越低。

$$\text{PoA} = \frac{\text{均衡策略的代价}}{\text{最优策略的代价}}$$

可以得知：$\text{PoA} \geq 1$。

- 考虑一个博弈系统$G = (N, S, u)$，N为参与人员，S为所有策略组合，u为效用函数。某一个策略 s 对整个系统的收益函数为：$\text{Welf}(s) = \sum_{i \in N} u_i\,(s)$，在博弈过程中，我们期望收益函数越大越好。同理可以定义代价函数 $\text{Cost}(s)$，期望代价函数越小越好。

无秩序代价（Price of Anarchy，PoA）

- 从效用函数角度定义无秩序代价:

$$\mathrm{PoA} = \frac{\max_{s \in S} \mathrm{Welf}(s)}{\min_{s \in E} \mathrm{Welf}(s)}$$

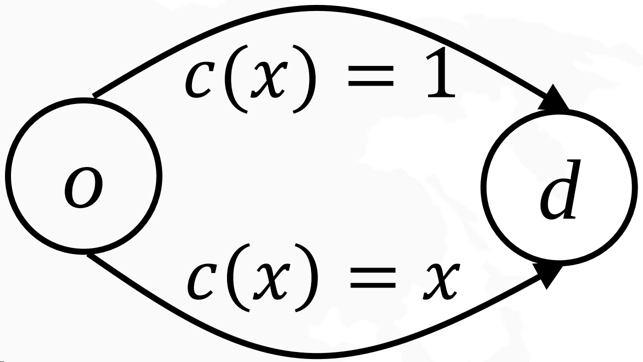
其中 E 为纳什均衡集中的策略组合， S 为所有的策略组合。根据定义可以得到：$\mathrm{PoA} \geq 1$

- 从代价函数角度定义无秩序代价:

$$\mathrm{PoA} = \frac{\max_{s \in E} \mathrm{Cost}(s)}{\min_{s \in S} \mathrm{Cost}(s)}$$

- 以布雷斯悖论中的扩展网络为例，$\mathrm{PoA} \geq 4/3$。

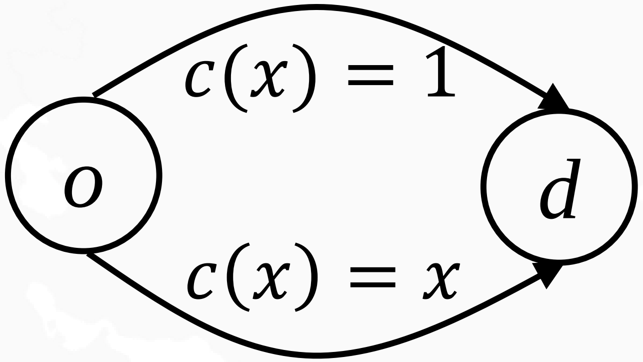
布雷斯悖论分析：Pigou网络



- 布雷斯悖论中的网络可简化为更一般的情况：
  - 网络包含两个顶点，出发点 o 和目标点 d；
  - 从 o 到 d 有两条路径，“上边一条”和“下边一条”；
  - 网络中有非负的交通流量 x，表示使用某路径的交通量占所有交通量的比例；
  - 在“上边一条”中，代价函数为常数 $c(x) = 1$，与当前路径上的交通流量比例无关；
  - 在“下边一条”中，代价函数为 $c(x) = x$，函数变量为当前路径上的交通流量比例（小于等于 x），其中 $c(\cdot)$ 非负、连续、非减；
  - “下边一条”是每个参与者的纳什均衡策略。

布雷斯悖论分析：Pigou网络

- Pigou 网络举例：线性代价函数



  - 对每一个单位来说，“下边一条”为纳什均衡策略。因此在纳什均衡策略集合中，系统的整体代价期望为1；
  - 对整个系统来说，代价期望为$x^2 - x + 1$，当 $x = 1/2$，代价期望达到最小值，为 3/4；
  - 因此在该Pigou网络中，PoA为 4/3。

布雷斯悖论分析：Pigou网络

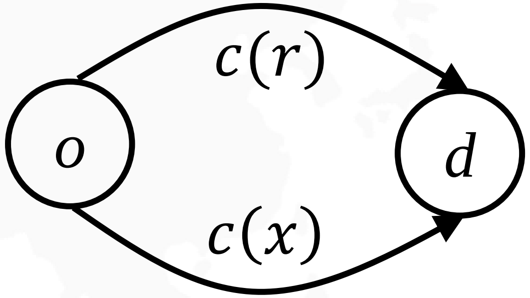
- Pigou 网络举例：非线性代价函数

函数类型	代表公式	PoA值
线性函数	x	$4/3 \approx 1.3$
二次函数	$x^2 + x$	$\frac{3\sqrt{3}}{3\sqrt{3}-2} \approx 1.6$
三次函数	$x^3 + x^2 + x$	$\frac{4\sqrt[3]{4}}{4\sqrt[3]{4}-3} \approx 1.9$
p次函数	$x^p + x^{p-1} + \cdots + x$	$\frac{(p+1)\sqrt[p]{p+1}}{(p+1)\sqrt[p]{p+1}-p} \approx \frac{p}{\ln p}$

- 随着代价函数阶 p 的增大，PoA增大。$p \to +\infty$时，PoA$\to +\infty$。

布雷斯悖论分析：Pigou网络

- 考虑通用的 Pigou 网络，其系统的最小代价为 $\inf_{0\le x\le r}\{x\cdot c(x)+(r-x)\cdot c(r)\}$，则对应的PoA为：



$$\sup_{0\le x\le r}\left\{\frac{r\cdot c(r)}{x\cdot c(x)+(r-x)\cdot c(x)}\right\}$$

- Pigou界：对于任意一个包含非负、连续、非减的函数集合 C，定义 Pigou 界 $\alpha(C)$ 为 Pigou 网络 PoA 的最大值，其中这些 Pigou 网络“下边一条”的代价函数来源于 C。形式上可以表示为：

$$\sup_{c\in C}\ \sup_{r\ge 0}\ \sup_{0\le x\le r}\left\{\frac{r\cdot c(r)}{x\cdot c(x)+(r-x)\cdot c(x)}\right\}$$

布雷斯悖论分析：Pigou网络

- 自私路由网络中的Pigou界定理

定理1：最大PoA值可达性

对于任意形式的自私路由网络，假设它每一条边上的代价函数都属于函数集合C，那么该网络的最大PoA值（最坏情况下的PoA值）可以在对应的Pigou网络中达到，其中对应Pigou网络的“下一条边”的代价函数属于函数集合C.

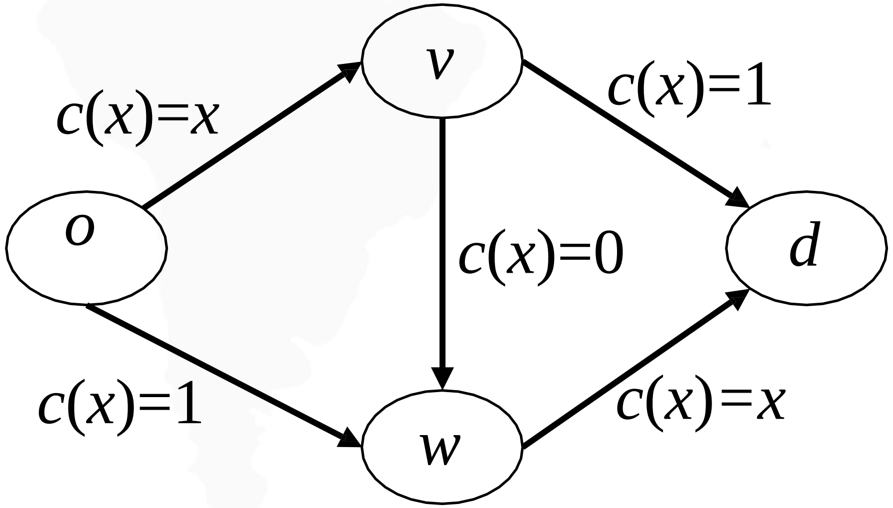
定理2：PoA值的Pigou界定理

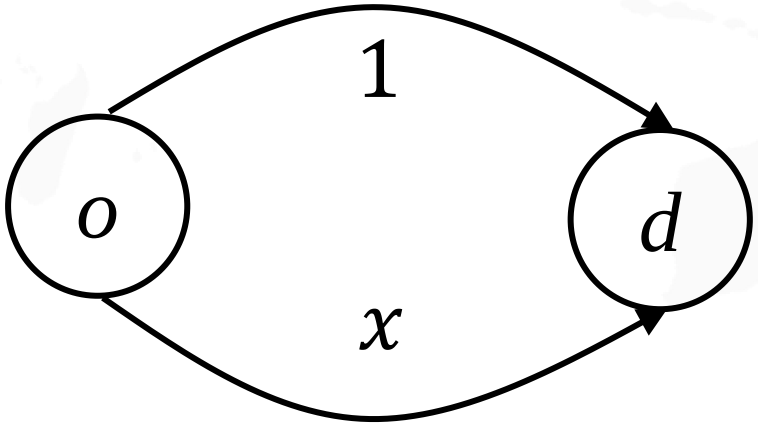
对于任意的自私路由网络，其每一条边上的代价函数都属于集合C，则该网络的PoA $\le a(C)$

- 关于定理的一些解释
  - 自私路由网络的PoA值和网络结构复杂度无关, 而和代价函数有关。
  - 计算复杂网络的PoA值问题可以转化为计算相对应的Pigou网络的PoA值。

布雷斯悖论分析：Pigou网络

- 应用举例：求解布雷斯悖论中扩展网络的PoA值
  - 50%司机选择o-v-d，剩下50%司机选择o-w-d，最终期望代价为1.5。在均衡策略o-v-w-d中，最终期望代价为2.0。根据PoA的定义，PoA≥4/3。
  - 将扩展网络转化为右边的 Pigou 网络，该Pigou 网络的PoA值为4/3。根据Pigou界定理，扩展网络的PoA ≤ 4/3。
  - 综上所述，布雷斯悖论中扩展网络的PoA值等于4/3。





第七讲：算法博弈论（2）

内容提纲

1 自私路由的低效率性

2 **自私路由的改进措施**

3 不同类型的均衡求解

4 均衡计算复杂度分析

对自私路由网络的改进策略

- 如何设计一些策略避免自私路由现象的发生或者减少其影响？

对自私路由网络的改进策略

- 改进策略：网络预留
  - 自私路由网络适用于实际应用中许多不同类型的网络，包括交通、通信和电子网络等；
  - 在通信网络中，一个很大的优势是，向网络添加额外的容量通常相对便宜和方便。正因为如此，一个通用的通信网络管理策略是安装比需要更多的容量，这意味着网络的容量通常不会被充分利用；
  - 这种网络预留的动机一方面是为了预测未来需求的增长。另一方面，过度供应也和网络性能有关，因为根据经验，在有额外容量的情况下，网络往往会遭受更少的数据包丢失和延迟；
  - 类似的还有排队等待限流。在上下班高峰期的地铁站中，通常用排队等待来限制地铁站中的人流。

对自私路由网络的改进策略

- 改进策略：网络预留
  - 引入预留参数$\beta \in (0,1)$，则β预留网络定义为：当网络达到均衡时，每一条边e上的流量都满足$f_e \leq (1-\beta)u_e$，也就是说在均衡时刻，每条边上的最大利用率不超过$1-\beta$。
  - 利用Pigou界定理可得β预留网络中最坏情况下的PoA值：

$$\frac{1}{2}(1+\sqrt{\frac{1}{\beta}})$$

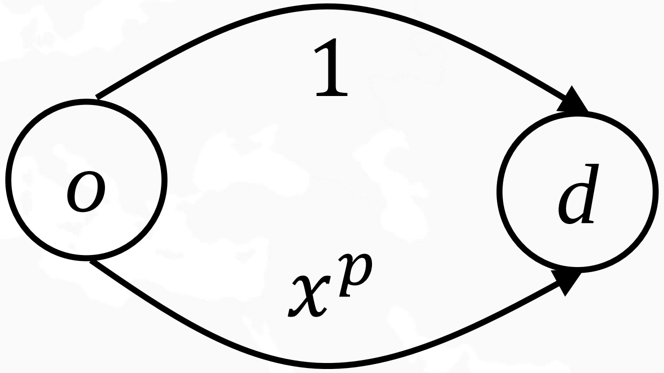
➢当$\beta \rightarrow 0$时，PoA$\rightarrow +\infty$，队列发生完全阻塞情况。

➢当$\beta \rightarrow 1$时，PoA$\rightarrow 1$，不存在自私路由带来的低效率，但流量利用率为0%，此时网络没有通信功能。

➢因此实际使用中采用一个折中的β，比如$\beta = 0.1$，此时网络流量利用率为90%，PoA值最多为2.1。

对自私路由网络的改进策略

- 改进策略：技术升级
  - 考虑Pigou网络



- 当流量为1时，该网络均衡策略的代价为$1^p = 1$
- 当流量为2时，该网络最优策略的代价为$(1+\varepsilon)+(1-\varepsilon)^{p+1}$
- 流量为1时网络均衡策略的代价小于流量为2时的网络最优策略代价

这一现象可以扩展为更一般的定理，即**资源扩大边界**（Resource Augmentation Bound）定理：对于任意的自私路由网络和流量r，其均衡策略的代价小于等于当流量为$2r$时网络最优策略的代价。

对自私路由网络的改进策略

- 对资源扩大边界定理的解释
  - 同一代价函数不同流量的网络进行比较，等价于对不同代价函数相同流量的网络进行比较。前者更容易解释和推理。
  - 直观地，对网络的通信技术进行升级，信息通过速度变得更快相当于网络流量变小。
  - 假定为网络流量变为原来的1/2，根据资源扩大边界定理，流量变为原来的1/2时，网络均衡策略代价小于等于技术改进之前网络的最优策略代价。
  - 因此，技术升级可以改进自私路由网络。

对自私路由网络的改进策略

- 还有其他改进策略吗？

第七讲：算法博弈论（2）

内容提纲

1 自私路由的低效率性
2 自私路由的改进措施
3 不同类型的均衡求解
4 均衡计算复杂度分析

布雷斯悖论中的均衡：纯策略纳什均衡

- 问题：对于更多的博弈，如果纯策略纳什均衡不存在怎么办？

纯策略纳什均衡 → 混合策略纳什均衡 → X均衡

？

均衡的重新形式化：纯策略纳什均衡

- 代价最小化博弈（Cost-Minimization Games）
  - 包含 k 个智能体（k 为有限个数）
  - 每个智能体 i 有一个有限的纯策略集合S_i，$s_i \in S_i$
  - 每个智能体都有一个非负代价函数$C_i(\boldsymbol{s})$，$\boldsymbol{s} = (s_1, \dots, s_k)$
- 使用代价最小化博弈定义的纯策略纳什均衡：

纯策略纳什均衡（Pure Nash Equilibrium，PNE）

一个策略组合$\boldsymbol{s} = (s_1, \dots, s_k)$是PNE，如果对于任意智能体$i$来说，$C_i(\boldsymbol{s}) \le C_i(s_i', \boldsymbol{s}_{-i})$。

纯策略纳什均衡不一定存在！

纯策略纳什均衡的PoA分析

- 在PNE条件下分析代价最小化博弈的PoA：

$$\text{PoA} = \frac{\text{纯策略纳什均衡的代价}}{\text{最优策略的代价}}$$

- 计算PNE的PoA前提条件是：确定纯策略纳什均衡的存在性。

势博弈（Potential Games）

存在势函数Φ，对于任意玩家i的单方面改变s_i'引起的代价改变等于势函数的变化：$\Phi(s_i', \boldsymbol{s}_{-i}) - \Phi(\boldsymbol{s}) = C_i(s_i', \boldsymbol{s}_{-i}) - C_i(\boldsymbol{s})$。

势博弈中PNE的存在性定理

势博弈必定存在至少一个纯策略纳什均衡。

均衡的重新形式化：混合策略纳什均衡

- 使用代价最小化博弈定义的混合策略纳什均衡：

混合策略纳什均衡（Mixed Nash Equilibrium，MNE）

一个混合策略组合$\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_k)$是MNE，如果对于任意智能体$i$和$s_i'$来说，$\mathbf{E}_{\boldsymbol{s}\sim\boldsymbol{\sigma}}[C_i(\boldsymbol{s})] \le \mathbf{E}_{\boldsymbol{s}\sim\boldsymbol{\sigma}}[C_i(s_i', \boldsymbol{s}_{-i})]$。

PNE是MNE的特例

代价最小化博弈中MNE一定存在

混合策略纳什均衡的PoA分析

- 在MNE条件下分析代价最小化博弈的PoA：

$$\mathrm{PoA}=\frac{\text{最差混合纳什均衡的期望代价}}{\text{最优策略的代价}}$$

- 但是计算MNE是困难的，普遍认为不存在多项式复杂度的解法！
- 如果算都算不出来，智能体怎么会达到MNE呢？
- 如果达不到MNE，那么分析它的PoA又有什么意义呢？
- 找更加容易计算的均衡！

比MNE容易计算的均衡概念

- 相关均衡（Correlated Equilibrium，CE）定义：

相关均衡（Correlated Equilibrium，CE）

一个定义在策略集$S_1 \times \cdots \times S_k$上的概率分布$\sigma$是CE，对于任意智能体$i$，若纯策略$s_i$及单方面策略改变$s_i'$满足：$\mathbf{E}_{\boldsymbol{s}\sim\sigma}[C_i(\boldsymbol{s})|s_i] \leq \mathbf{E}_{\boldsymbol{s}\sim\sigma}[C_i(s_i', \boldsymbol{s}_{-i})|s_i]$。

- 期望比较计算：$\sum_{\boldsymbol{s}_{-i}} C_i(s_i, \boldsymbol{s}_{-i}) p(s_i, \boldsymbol{s}_{-i}) \leq \sum_{\boldsymbol{s}_{-i}} C_i(s_i', \boldsymbol{s}_{-i}) p(s_i, \boldsymbol{s}_{-i})$
- $p(s_i, \boldsymbol{s}_{-i})$是纯策略组合$(s_i, \boldsymbol{s}_{-i})$在$\sigma$中对应的概率。
- 注意：σ定义在策略集$S_1 \times \cdots \times S_k$上的任意概率分布，而不一定是联合概率分布（MNE是），因此MNE是CE的一个子集。
- MNE是CE的子集，MNE一定存在，因此CE一定存在。

相关均衡的PoA分析

- 相关均衡的PoA计算公式同MNE，但比MNE更容易计算！
- 相关均衡的直观理解：
  - 一个受信任的第三方U，σ各方都知道
  - U首先从σ中采样一个策略组合$(s_1, \ldots, s_k)$
  - 然后U私底下告诉每一个参与者i：$s_i, i \in [1, \ldots, k]$
  - 参与者i决定是否按照U的建议s_i行动，或者是采取其他策略
  - 由于i知道s_i也知道σ，因此i可以计算出当所有玩家都采取第三方建议时自身的期望代价$\mathbf{E}_{\boldsymbol{s}\sim\sigma}[C_i(\boldsymbol{s})|s_i]$；$i$也可以计算出自己采取其他策略$s_i'$，而其他玩家采取第三方建议时自身的期望代价$\mathbf{E}_{\boldsymbol{s}\sim\sigma}[C_i(s_i', \boldsymbol{s}_{-i})|s_i]$。
- 相关均衡的含义是：假设其他人都采取第三方建议，每个人采取第三方建议时的期望代价最小，没人愿意偏离第三方的建议！

相关均衡的PoA分析

P1 \ P2	Stop	Go
Stop	1,1	1,0
Go	0,1	5,5

- 相关均衡的例子
  - 注意我们讨论的是代价，值越小越好
  - 两个PNE：（Go，Stop）和（Stop，Go）
  - 不公平！我们想要两方各有一半的时间Go，一半的时间Stop
  - 也就是说$\sigma(\text{Stop}, \text{Go}) = {}^{1}/_{2}$, $\sigma(\text{Go}, \text{Stop}) = {}^{1}/_{2}$
  - σ明显不是一个Product Distribution，不是MNE，是CE
  - 第三方U可以看作是一个信号灯
  - 如果U告诉我们Go，那么通过σ我们知道对方的建议是Stop，如果对方采取建议动作Stop，我们最好也选择建议的动作Go
  - 同理，如果U告诉我们Stop，那么通过σ我们知道对方的建议是Go，如果对方采取建议动作Go，我们最好也选择建议的动作Stop，因此，没人愿意偏离第三方的建议

比MNE容易计算的均衡概念

- 粗相关均衡（Coarse Correlated Equilibrium，CCE）定义：

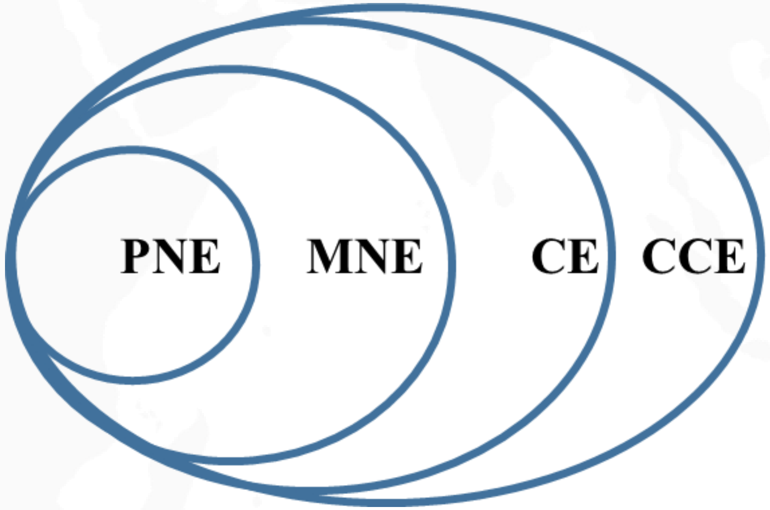
粗相关均衡（Coarse Correlated Equilibrium，CCE）

一个定义在策略集$S_1 \times \cdots \times S_k$上的分布$\sigma$是CE，对于任意智能体$i$，若单方面策略改变$s_i'$满足：$\mathbf{E}_{\boldsymbol{s}\sim\sigma}[C_i(\boldsymbol{s})|s_i] \leq \mathbf{E}_{\boldsymbol{s}\sim\sigma}[C_i(s_i', \boldsymbol{s}_{-i})|s_i]$。

- 期望比较计算：$\sum_{\boldsymbol{s}_{-i}} C_i(s_i, \boldsymbol{s}_{-i}) p(s_i, \boldsymbol{s}_{-i}) \leq \sum_{\boldsymbol{s}_{-i}} C_i(s_i', \boldsymbol{s}_{-i}) p(s_i, \boldsymbol{s}_{-i})$
- $p(s_i, \boldsymbol{s}_{-i})$是纯策略组合$(s_i, \boldsymbol{s}_{-i})$在$\sigma$中对应的概率。
- 注意： σ定义在策略集$S_1 \times \cdots \times S_k$上的任意分布，而不一定是概率分布（要求变得更低），因此CCE只防止无条件的单方面改变，而不以s_i 为条件。CCE比CE更容易计算。

粗相关均衡的PoA分析

- 粗相关均衡的PoA计算公式同CE，但比CE更容易计算！
- PNE、MNE、CE、CCE范围越来越大，越来越容易计算，存在性有保证，使得PoA分析越来越有意义且越容易，但也会使得PoA的值理论上越来越大！



- 是否存在一些特殊类型的博弈，使得CCE之类的均衡解的PoA值也比较小？
- 有！ Smooth Games！

平滑博弈（Smooth Games）

- 平滑博弈（Smooth Games）定义：

平滑博弈（Smooth Games）

一个代价极小化博弈是(λ, μ)-smooth的，如果对于任何策略组合$\boldsymbol{s}$和$\boldsymbol{s}^*$来说：

$$\sum_{i=1}^{k} C_i(s_i^*, \boldsymbol{s}_{-i}) \leq \lambda \cdot \mathrm{cost}(\boldsymbol{s}^*) + \mu \cdot \mathrm{cost}(\boldsymbol{s}),\ \mathrm{cost}(\boldsymbol{s}) = \sum_{i=1}^{k} C_i(\boldsymbol{s})$$

其中，当(λ, μ)取值较小时，玩家i偏离自己策略带来的整体代价是有界的，PoA值一般也比较小。

平滑博弈（Smooth Games）

- 对于一个(λ,μ)-smooth博弈，它的PNE解$\boldsymbol{s}$的代价至多是最优解$\boldsymbol{s}^*$代价的$\frac{\lambda}{1-\mu}$倍，也就是$\mathrm{PoA} \le \frac{\lambda}{1-\mu}$：
  - $\mathrm{cost}(\boldsymbol{s}) = \sum_{i=1}^{k} C_i(\boldsymbol{s})$ 定义
  - $\le \sum_{i=1}^{k} C_i(s_i^*, \boldsymbol{s}_{-i})$ 均衡的定义，偏离均衡代价只会变大
  - $\le \lambda \cdot \mathrm{cost}(\boldsymbol{s}^*) + \mu \cdot \mathrm{cost}(\boldsymbol{s})$ Smooth Game的定义
  - 推出$\frac{\mathrm{cost}(\boldsymbol{s})}{\mathrm{cost}(\boldsymbol{s}^*)} \le \frac{\lambda}{1-\mu}$，也就是$\mathrm{PoA} \le \frac{\lambda}{1-\mu}$
- PNE不一定存在，但是这个结论可以推广到一定存在且更容易计算的粗相关均衡CCE上，
- 因此，Smooth Games的PoA是鲁棒的（Robust）

平滑博弈（Smooth Games）

- 对于一个(λ,μ)-smooth博弈，它的CCE解σ的代价至多是最优解$\boldsymbol{s}^*$代价的$\frac{\lambda}{1-\mu}$倍，也就是PoA $\leq \frac{\lambda}{1-\mu}$:
  - $\mathbf{E}_{\boldsymbol{s}\sim\sigma}[\text{cost}(\boldsymbol{s})] = \mathbf{E}_{\boldsymbol{s}\sim\sigma}[\sum_{i=1}^{k} C_i(\boldsymbol{s})]$ cost的定义
  - $= \sum_{i=1}^{k} \mathbf{E}_{\boldsymbol{s}\sim\sigma}[C_i(\boldsymbol{s})]$
  - $\leq \sum_{i=1}^{k} \mathbf{E}_{\boldsymbol{s}\sim\sigma}[C_i(s_i^*, \boldsymbol{s}_{-i})]$均衡的定义，偏离均衡代价只会变大
  - $= \mathbf{E}_{\boldsymbol{s}\sim\sigma}[\sum_{i=1}^{k} C_i(s_i^*, \boldsymbol{s}_{-i})]$
  - $\leq \mathbf{E}_{\boldsymbol{s}\sim\sigma}[\lambda \cdot \text{cost}(\boldsymbol{s}^*) + \mu \cdot \text{cost}(\boldsymbol{s})]$ Smooth Game的定义
  - $= \lambda \cdot \text{cost}(\boldsymbol{s}^*) + \mu \cdot \mathbf{E}_{\boldsymbol{s}\sim\sigma}[\text{cost}(\boldsymbol{s})]$
- 推出$\frac{\mathbf{E}_{\boldsymbol{s}\sim\sigma}[\text{cost}(\boldsymbol{s})]}{\text{cost}(\boldsymbol{s}^*)} \leq \frac{\lambda}{1-\mu}$，也就是PoA $\leq \frac{\lambda}{1-\mu}$，证毕

平滑博弈（Smooth Games）

- 这些良好的性质同样可以扩展到近似纳什均衡上，这里以近似PNE为例，其他均衡同理：

ϵ–纯策略纳什均衡（ϵ-PNE）

给定$\epsilon > 0$，策略组合$\boldsymbol{s}$是ϵ-PNE，如果对于任意智能体i，s_i'来说：

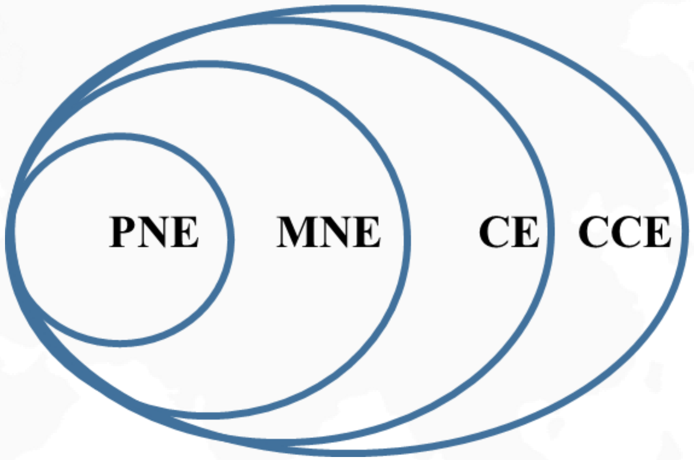
$$C_i(\boldsymbol{s}) \leq (1+\epsilon)C_i(s_i', \boldsymbol{s}_{-i})$$

Smooth Game的ϵ-PNE的PoA界

对于一个(λ, μ)-smooth博弈，它的ϵ-PNE的PoA：

$$\text{PoA} \leq \frac{(1+\epsilon)\lambda}{1-\mu(1+\epsilon)}$$

不同类型的均衡解概念小结



- PNE：不一定存在
- MNE：一定存在，难以计算
- CE：一定存在，容易计算
- CCE：一定存在，更容易计算
- 势函数、势博弈：对于任意玩家的单方面改变引起的代价改变等于势函数的变化
- 势博弈必定存在至少一个纯策略纳什均衡
- 均衡解范围越来越大，存在性有保证，但是PoA的值理论上说也会越来越大，而Smooth Games可以保证PoA值不会太大

第七讲：算法博弈论（2）

内容提纲

1 自私路由的低效率性

2 自私路由的改进措施

3 不同类型的均衡求解

4 均衡计算复杂度分析

研究均衡计算复杂度的原因

- 前面所讲内容尚未回答的问题

对于一个存在纳什均衡的博弈，即使所有参与人都足够“聪明”，他们都能找到均衡吗?

即使所有参与人都能够找到纳什均衡，他们用来找到纳什均衡的算法是什么呢?

即使他们都有一个能够找到纳什均衡的算法，算法会在多长时间找到均衡呢?

回顾：纳什均衡的计算问题

- 证明了纳什均衡的存在性，如何得到这个纳什均衡？
- 理论计算机科学家更关心如何利用计算机算法搜索得到博弈的具体均衡解：

求解纳什均衡是极具挑战性的开放难题
- **纳什均衡**这个博弈博弈论解来源于经济学中分析经济现象，经济学家只关心存在性，根本不关心均衡解如何计算
- 求解一般的**两人博弈纳什均衡**是PPAD-Hard问题（不可解，除非P=NP）
- 求解四人/三人纳什均衡都是PPAD-Hard问题，但**求解多人（五人及以上）纳什均衡**是否仍是PPAD-Hard问题还不确定
- **近似纳什均衡**求解相关研究进展缓慢
- **纳什均衡选择**问题仍然存在很多问题
- 纳什均衡的**理性假说**实际中可能不成立

一些**典型问题**的纳什均衡求解复杂度
- **两人变和策略式（矩阵式）博弈**
  - 计算纳什均衡：PPAD-Hard
  - 计算均衡个数：#P-Hard
  - 检查均衡唯一性：NP-Hard
  - 计算个体最少收益：NP-Hard
  - 计算群体最小收益和：NP-Hard
- **随机博弈**（带随机状态转移的动态博弈）
  - 检查纯策略纳什均衡存在性：**PSPACE-Hard**
  - 计算任意策略的最优响应：**Not Turing-computable**
  - 上述结论适用于两人对称博弈

这一部分，我们就讲解关于博弈计算的复杂性问题

计算复杂性理论简介

- Computational Complexity Theory
  - 理论计算机科学和数学的一个分支
  - 对于计算机而言，任何一个问题的求解都需要资源，计算复杂性理论定量分析计算解决问题所需的资源：时间和空间
  - 将计算问题按照所需时间资源的不同予以分类，就得到了常见的P、NP、NP完全、NP难这样的概念
- 时间复杂度：问题规模扩大后，程序需要的时间增长得有多快
  - $O(1)$：常数级复杂度，不管数据有多大，花的时间总是那么多
  - $O(n)$：数据规模变得有多大，花的时间也跟着变得有多长
  - $O(n^2)$：数据扩大2倍，时间变慢4倍
  - $O(a^n)$：指数级复杂度

计算复杂性理论简介

- 时间复杂度：问题规模扩大后，程序需要的时间增长得有多快
  - $O(n!)$：阶乘级复杂度
  - $O(2n^2) = O(n^2)$
  - $O(n^3 + n^2) = O(n^3)$
  - $O(0.01n^3)$效率低于$O(100n^2)$
  - $O(1.01^n)$效率低于$O(n^{100})$
  - $O(1), O(\log(n)), O(n^a)$：多项式复杂度
  - $O(a^n), O(n!)$：非多项式复杂度
- 后者的复杂度无论如何都远远大于前者
- 我们选择的算法通常都需要是多项式级的复杂度，非多项式级的复杂度需要的时间太多，其复杂度计算机往往不能承受

计算复杂性理论简介

- 是不是所有的问题都可以找到多项式复杂度的算法呢?
  - Hamilton回路问题：给一个图，能否找到一条经过每个顶点一次且恰好一次（不遗漏也不重复）最后又走回来的路（满足这个条件的路径叫做Hamilton回路）。这个问题现在还没有找到多项式的算法
- P类问题
  - 如果一个问题可以找到一个能在多项式的时间里解决它的算法，那么这个问题就属于P问题
  - 大学《算法与数据结构》课程中的大部分问题都是P类问题
- NP问题
  - NP问题不是非P类问题，NP问题是指可以在多项式的时间里验证一个解的问题；有可能找一个解很困难，但验证一个解很容易

计算复杂性理论简介

- Hamilton回路是一个NP问题
  - 因为验证一条路是否恰好经过了每一个顶点非常容易
  - 但是找到一条这样的路径却很难
  - 通常只有NP问题才可能找到多项式的算法，我们不会指望一个连多项式时间验证一个解都不行的问题存在一个解决它的多项式时间复杂度的算法
- P类问题都是NP问题
  - 能多项式时间地解决一个问题，必然能多项式时间验证一个问题的解
- 是否所有的NP问题都是P类问题?
  - 即是通常所谓的NP问题：证明或推翻P=NP
  - 世界七大数学难题之一

纯策略纳什均衡求解算法

- 纯纳什均衡（PNE）定义回顾：假定G是最小化代价的博弈，$s = \{s_1, \ldots, s_k\}$是每一个参与者的策略组合。s_i代表参与者 i可选的策略集合s_{-i}代表除了i之外其他人的策略组合。$C_i(\cdot)$ 为参与者i的代价函数。如果对于每一个参与者 i 和任意一个单方面的偏差$s_i{}'$都满足：$c_i(s) \leq c_i(s_i', s_{-i})$, 则称 s 为G的纯策略纳什均衡。
- 势博弈（potential game，也称为潜在博弈）定义：
  - 存在有界实值势函数 Φ，使得对于任意单方面偏差s_i'引起的代价改变都等于势函数的变化：$\Phi(s_i', s_{-i}) - \Phi(s) = c_i(s_i', s_{-i}) - c_i(s)$
- 纯纳什均衡在很多博弈问题中不存在
- 势博弈中存在至少一个纯纳什均衡

纯策略纳什均衡求解算法

- **最优反应**（Best Response）：在博弈中，假如其他人所采取的行动（策略选择）是已知或者能被预测的，根据这个已知的或可预测的行动而采取的能使自己的代价最小化（收益最大化）的策略。
- 动态最优反应算法 （Best-Response Dynamics）

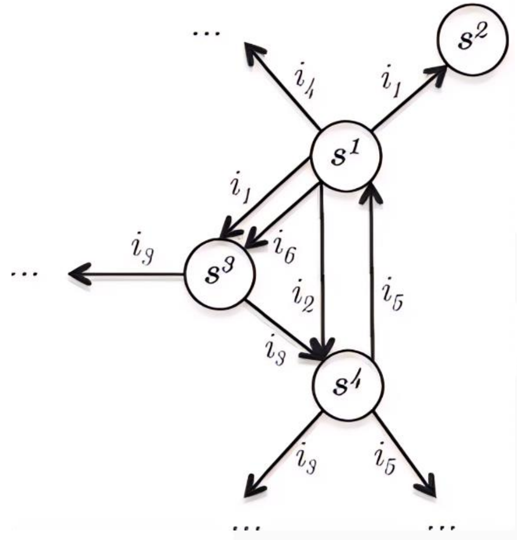
```
While 当前策略 s 不是纳什均衡
   选择任意一个参与者i
   针对 i 选择一个单方面偏差s_i'，使得
      c_i(s_i', s_{-i}) < c_i(s)
   更新s = {s_i', s_{-i}}
End While
```

- 第二行中i 可以任意选择，也可以改进为按照预定次序选择，比如$i = 1, \ldots, k$
- 为了提升算法效率，选择s_i'的时候也可以使之满足$s_t' = \text{argmin}_{s_t'} c_i\,(s_i', s_{-i})$
- 该算法可以用任意随机策略作为初始化

纯策略纳什均衡求解算法

- 动态最优反应算法可以看作是有向图上的行走过程，每个顶点代表一个策略组合，每条有向边的末端策略对应的代价函数要小于前端策略对应的代价函数。当一个顶点只有进入的边，没有出去的边，则该顶点为当前博弈问题的纯纳什均衡策略。比如：

S^2为该博弈问题的纯纳什均衡策略



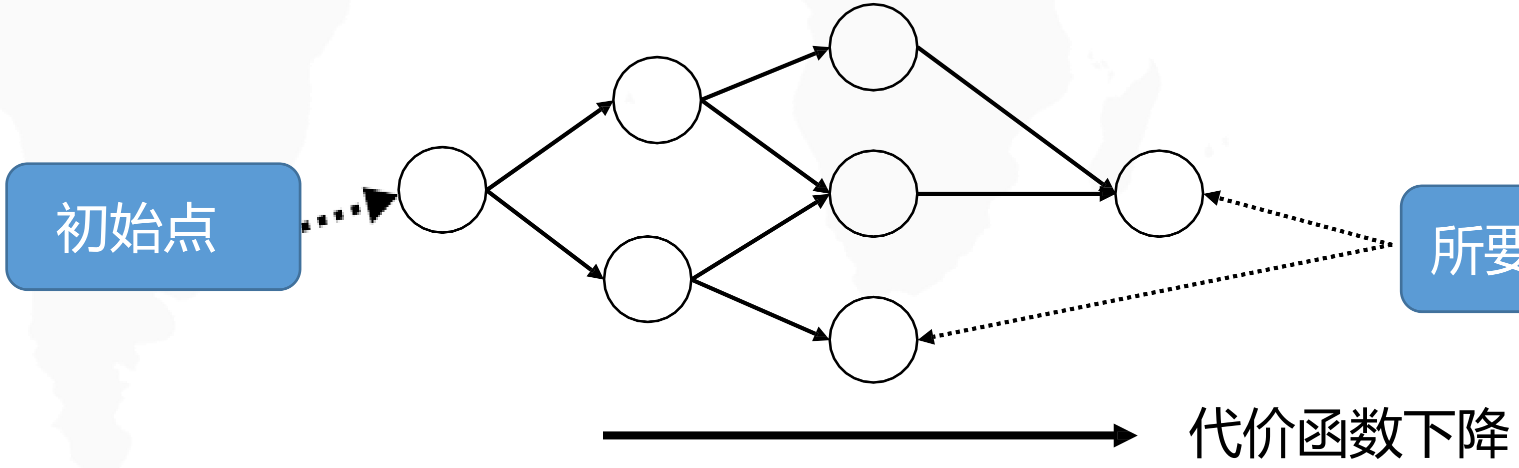
- 在势博弈问题中，从任意初始策略开始，动态最优反应算法都会收敛到一个纯纳什均衡上。

纯策略纳什均衡求解算法

- 局部搜索问题包含三个子算法:
  - 第一个多项式算法：输入某个样例，输出可行解
  - 第二个多项式算法：输入该样例和可行解，输出目标函数值
  - 第三个多项式算法：输入该样例和可行解，输出是否局部最优，或者给出更好的解
- 在势博弈中，动态最优反应算法是一个局部搜索问题:
  - 第一个算法：任意选择一个策略组合
  - 第二个算法：计算该策略组合下的代价函数值
  - 第三个算法：判断是否为纯纳什均衡，并更新策略组合
  - 第三个算法需遍历所有可能，通常遍历次数会根据输入空间呈指数增长

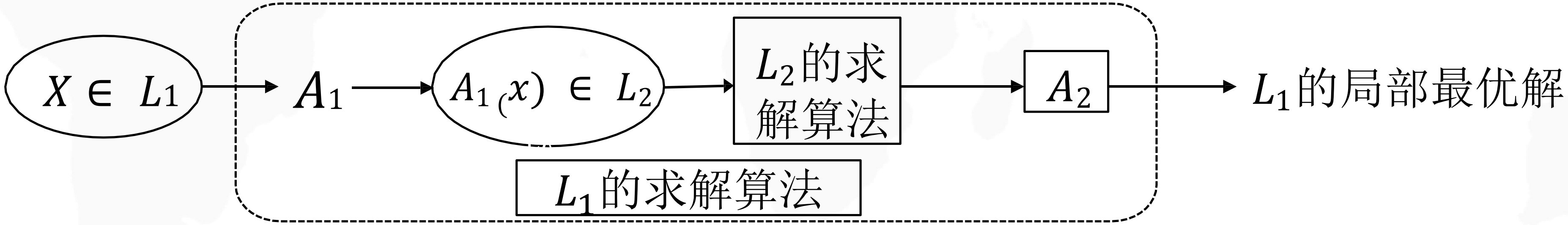
纯策略纳什均衡求解算法

- 关于局部搜索算法的一些说明
  - 局部搜索的第三个算法通常需要进行多次迭代才能确认。
  - 如果求解全局最优解，则局部搜索问题为NP难问题。
  - 如果求解局部最优解，则局部搜索问题很容易被解决。
  - 局部搜索算法可以看作是在有向无圈图上的行走过程。每一个顶点代表可行解，终点代表一个局部最优解。局部搜索问题可以描述为从任意一个点出发寻找终点的问题。



纯策略纳什均衡的计算复杂度

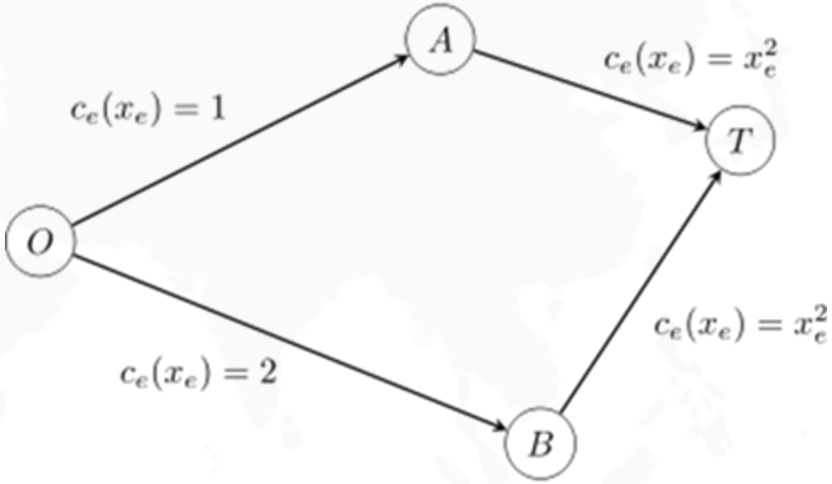
- PLS：局部搜索问题集合（Problems of Local Search）
- PLS 归纳（PLS Reduction）：
  - 定义：设$L1 \in PLS, L2 \in PLS$,如果存在两个多项式复杂度算法 $A1, A2$ 满足如下条件，则称为$L1$问题归纳到 $L2$问题：
    - $A1$把L_1的每一个实例x映射到$L2$的一个实例，即$A1(x) \in L2$
    - A2把L_2的每一个局部最优解$A1(x)$映射到L_1的局部最优解x
  - 解释：可以在多项式时间内将求解L_1问题转化为求解L_2问题



- PLS 完全问题：如果所有PLS问题都可以归纳到 L 问题，则称 L 为 PLS 完全问题。多数专家认为不存在多项式复杂度的算法求解PLS完全问题。

纯策略纳什均衡的计算复杂度

- 堵塞博弈（Congestion game）
  - 通信网络中的博弈包含所有堵塞元素的基本集合E，n个参与人，每个参与人i的策略输入策略集合$S_i \subseteq 2^E$。E代表每一条边，每个人的策略为是否选择其中的每一条通道。每条边上都有代价函数，代价函数非负且单调递增。如右图：



- 在堵塞博弈中，计算纯纳什均衡是一个PLS完全问题
- 假设所有的参与人都有同一个出发点和目的点，计算堵塞博弈的纯纳什均衡依然是一个PLS完全问题。

混合策略纳什均衡求解算法

- 混合纳什均衡（MNE）定义回顾：$\sigma_1, \dots, \sigma_k$是策略集合$S_1, \dots, S_k$上的概率分布，$\sigma = \sigma_1 \times \cdots \times \sigma_k$，对于每一个参与人$i \in \{1, \dots k\}$和任何的单方面偏差$s_i'$,都满足：$E_{s\sim\sigma}[C_i(s)] \le E_{s\sim\sigma}[C_i(s_i', s_{-i})]$。
- 在所有有限博弈中混合纳什均衡一定存在，但是很难计算。
- 后悔值：设已经博弈了T个回合，对于参与人i，执行策略$s_i^1, \dots s_i^T$的后悔值为：$\frac{1}{T}\left[\sum_{t=1}^{T} c_i^t(s_i^t) - \min_{s\in S_i} \sum_{t=1}^{T} c_i^t(s)\right]$
  - 定义中的代价函数(c)与MNE定义中的代价函数(C)不同，在本定义中，代价函数本身包含了对手的决策，因此是回合相关的。MNE中，代价函数只与博弈规则有关。

混合策略纳什均衡求解算法

- 不后悔算法（Not-Regret Algorithm）
  - 对于任意小的数$\varepsilon > 0$，存在一个回合范围T_0，当$T > T_0$时，根据算法A选择的决策的后悔值都小于ε，则称A为不后悔算法。换言之，在不后悔算法中，$T \to +\infty$时，后悔值趋于0。
- 不后悔算法设计原则
  - 过去的博弈经验应该应用于指导未来的选择策略的概率分布
  - 如果某个策略带来了高代价，则需要降低该策略的选择概率

不后悔算法存在性定理：

对于n个动作的集合A，$T \geq 4\ln n$，存在一个在线决策算法，对任意对手，它的后悔值至多为$2\sqrt{(\ln n)/T}$；如果$\epsilon \in (0,1]$，$T \geq 4\ln n/\epsilon^2$，存在一个在线决策算法，对任意对手，它的后悔值至多为ϵ。

混合策略纳什均衡求解算法

- 不后悔算法举例-权重更新算法（Multiplicative Weights）

初始化每个策略的权重$w^1(a) = 1, a \in A$
for $t = 1,2,\ldots,T$ do
  利用权重计算每个策略的选择概率$p^t(a) = w^t(a)/\sum_{\tilde{a}\in A} w^t(\tilde{a})$
  根据代价重新计算每个策略的权重$w^{t+1}(a) = w^t(a)(1-\beta c^t(a))$

代价越大，权重越低

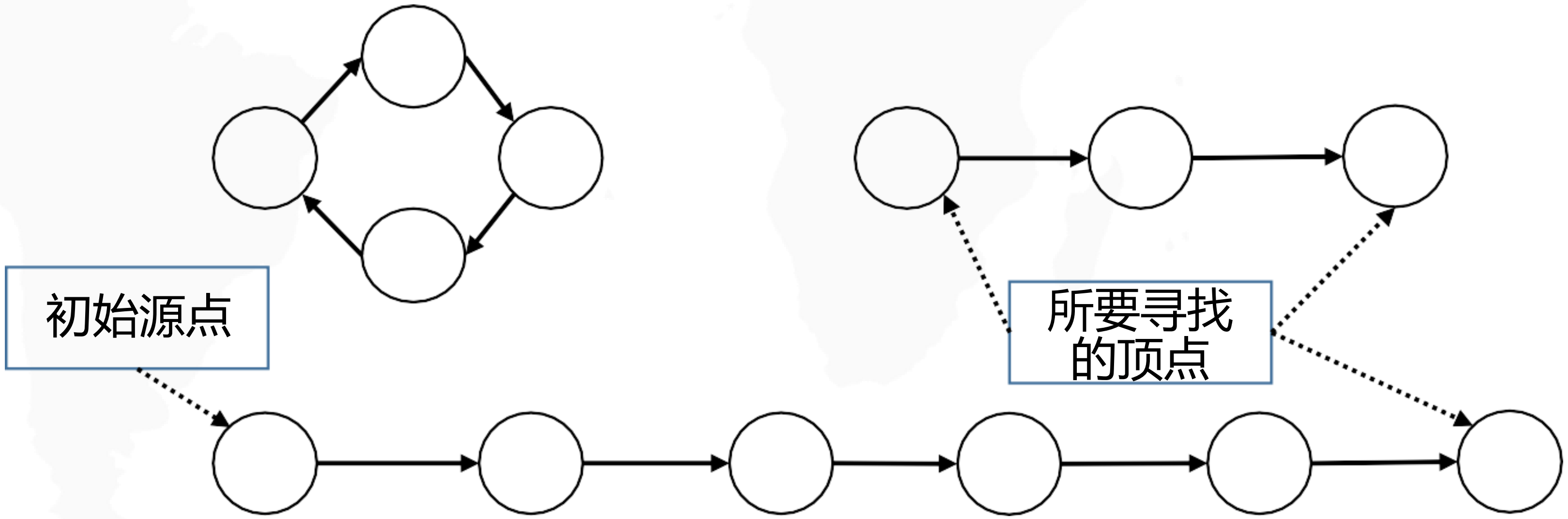
- 动态不后悔算法（No-Regret Dynamics，从一个玩家到多个玩家）

初始化每个策略的选择概率
for $t = 1,2,\ldots,T$ do
  每一个参与人i根据当前各自的概率分布选择决策，看作是纯决策
  对每个参与人i的决策计算代价值$c_i^t(a_i) = E_{e_{-i}^t \sim \sigma_{-i}^t}\left[C_i\left(a_i, a_{-i}^t\right)\right], \sigma_{-i}^t = \prod_{j\neq i} \sigma_j^t$
  选择不后悔算法更新每个参与人i的策略选择分布函数

- 在二人零和博弈中，动态不后悔算法收敛于混合策略纳什均衡

混合策略纳什均衡算法复杂度

- PPAD（有向图的多项式校验参数）问题定义
  - 假设G（尺度很可能是指数增长）为一个不包含孤立结点的有向图，每一个顶点最多只有一个前驱顶点和后向顶点，即出度和入度最多为1。G 中存在一个多项式时间的函数$f(v)$可以从 顶点v中得到它的前驱顶点和后向顶点。给定某一个源点 s（入度为0），从 G 中找出除了 s 之外的所有出度为 0 或者入 度为 0 的顶点。 如下图所示：



混合策略纳什均衡算法复杂度

- 双矩阵博弈问题
  - 两个参与人分别有不同的 $n \times m$ 维的收益矩阵A 和 B，其中 n 和 m 分别为两个人可采取的策略数目。
  - 二人零和博弈问题（如猜拳游戏）是双矩阵博弈问题的一种特殊情况，即：$A = -B$
- 目前没有结论能证明，双矩阵博弈的混合纳什均衡问题可以用多项式复杂度的算法解决。
- 在双矩阵博弈问题中，混合纳什均衡是一个PPAD 完全问题。 虽然 PPAD 和 NP 很接近，但PPAD完全是 NP 完全的不充分条件。

粗相关均衡求解算法

- 粗相关均衡定义回顾：一个定义在策略集$S_1 \times \cdots \times S_k$上的分布$\sigma$是CE，当且仅当对于任意智能体$i$，若单方面策略改变$s_i'$满足$\mathbf{E}_{\boldsymbol{s}\sim\sigma}[C_i(\boldsymbol{s})|s_i] \le \mathbf{E}_{\boldsymbol{s}\sim\sigma}[C_i(s_i', \boldsymbol{s}_{-\boldsymbol{i}})|s_i]$。
- 均衡求解：同样使用动态不后悔策略（No-Regret Dynamics, NRD）

每一个时刻$t = 1,2, \dots, T$:
- ➢ 每一个智能体i都独立按照一个不后悔算法选择当前时刻的混合策略p_i^t
- ➢ 每个智能体i都得到一个代价向量c_i^t， $c_i^t(s_i) = \mathbf{E}_{\boldsymbol{s}^t_{-i}\sim\sigma^t_{-i}}[C_i(s_i, \boldsymbol{s}^t_{-\boldsymbol{i}})]$，其中，$\sigma^t_{-i}$是product distribution：$\prod_{j\neq i} p_j^t$，注意$c$和$C$的区别，$C$是博弈规则决定的

- 非常重要的结论：历史平均策略收敛到CCE
- Fundamental Connection Between a **Static** Equilibrium Concept and the Outcomes of Natural Learning **Dynamics**!

粗相关均衡求解算法

- 算法的收敛性问题

No-Regret Dynamics收敛性定理：

假设NRD迭代了T轮，每一个智能体$i = 1, \ldots, k$的后悔值都至多为ϵ。记$\sigma^t = \prod_{i=1}^{k} p_i^t$，$\sigma = \frac{1}{T}\sum_{t=1}^{T} \sigma^t$为历史平均策略，那么$\sigma$是一个近似粗相关均衡CCE，也就是说：

$$\mathbf{E}_{\boldsymbol{s}\sim\sigma}[C_i(\boldsymbol{s})] \le \mathbf{E}_{\boldsymbol{s}\sim\sigma}[C_i(s_i', \boldsymbol{s}_{-i})] + \epsilon$$

- 证明：根据σ的定义，$\mathbf{E}_{\boldsymbol{s}\sim\sigma}[C_i(\boldsymbol{s})] = \frac{1}{T}\sum_{t=1}^{T} \mathbf{E}_{\boldsymbol{s}\sim\sigma^t}[C_i(\boldsymbol{s})]$，$\mathbf{E}_{\boldsymbol{s}\sim\sigma}[C_i(s_i', \boldsymbol{s}_{-i})] = \frac{1}{T}\sum_{t=1}^{T} \mathbf{E}_{\boldsymbol{s}\sim\sigma^t}[C_i(s_i', \boldsymbol{s}_{-i})]$，由于每一个智能体的后悔值至多为$\epsilon$：$\frac{1}{T}\sum_{t=1}^{T} \mathbf{E}_{\boldsymbol{s}\sim\sigma^t}[C_i(\boldsymbol{s})] \le \frac{1}{T}\sum_{t=1}^{T} \mathbf{E}_{\boldsymbol{s}\sim\sigma^t}[C_i(s_i', \boldsymbol{s}_{-i})] + \epsilon$，
- 证毕。

相关均衡求解算法

- 相关均衡的定义回顾：一个定义在策略集$S_1 \times \cdots \times S_k$上的概率分布$\sigma$是CE，对于任意智能体$i$，若纯策略$s_i$及单方面策略改变$s_i'$满足$\mathbf{E}_{\boldsymbol{s}\sim\sigma}[C_i(\boldsymbol{s})|s_i] \le \mathbf{E}_{\boldsymbol{s}\sim\sigma}[C_i(s_i', \boldsymbol{s}_{-\boldsymbol{i}})|s_i]$。

交换后悔值（Swap Regret，SR）：

给定每一轮的代价向量$c^1, \dots, c^T$，动作序列$a^1, \dots, a^T$的SR：

$$\frac{1}{T}\left[\sum_{t=1}^{T} c^t(a^t) - \min_{\delta: A\to A} \sum_{t=1}^{T} c^t(\delta(a^t))\right]。$$

后悔值（Regret），又称外部（External）后悔值：

给定每一轮的代价向量$c^1, \dots, c^T$，动作序列$a^1, \dots, a^T$的后悔值：

$$\frac{1}{T}\left[\sum_{t=1}^{T} c^t(a^t) - \min_{a\in A} \sum_{t=1}^{T} c^t(a)\right]。$$

- 后悔值是SR的特例，可以认为采用一个constant swapping function！

相关均衡求解算法

- No-Swap-Regret Dynamics（NSRD）

每一个时刻$t = 1,2,\ldots,T$:
- 每一个智能体i都独立按照一个NSRA选择当前时刻的混合策略p_i^t
- 每个智能体i都得到一个代价向量c_i^t， $c_i^t(s_i) = \mathbf{E}_{\boldsymbol{s}_{-i}^t \sim \sigma_{-i}^t}[C_i(s_i, \boldsymbol{s}_{-i}^t)]$，其中，$\sigma_{-i}^t$是product distribution： $\prod_{j \neq i} p_j^t$

- No-Swap-Regret Algorithm（NSRA）

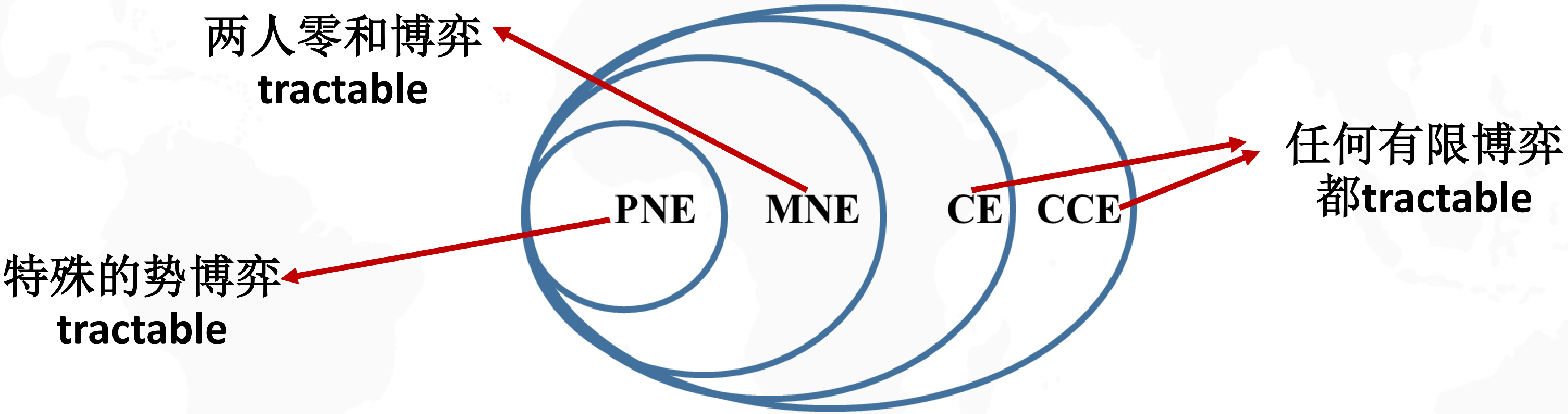
No-Swap-Regret Algorithm（NSRA）：

一个在线决策算法$\mathcal{M}$，如果对于任意$\epsilon > 0$，存在一个足够大的$T = T(\epsilon)$，满足对于任意的对手，期望Swap Regret至多为ϵ。

- 收敛性定理：只要有一个NDRA，NSRD就能保证收敛。

上述算法的Tractability问题

- NRD和NSRD的收敛性保证了任何有限博弈的CE和CCE都是可以处理的。



均衡计算复杂性小结

- 四种Tractability的结果： No-Regret Dynamics收敛到CCE、 No-Swap-Regret Dynamics收敛到CE、ϵ-BRD-MG在α-bounded原子自私路由中收敛到PNE
- P、NP、NP完全、NP难
- 抽象局部搜索问题
  - PLS问题：所有抽象局部搜索问题的集合
  - PLS约化、PLS完全问题
- 计算阻塞博弈中的PNE是一个PLS完全问题
- 求解一般的两人博弈的MNE是PPAD完全问题
- 求解三人/四人博弈的MNE是PPAD完全问题

本讲内容小结

知识点

- **自私路由：建模分析**
- **自私路由的改进策略**
- **不同类型的均衡求解**
- **均衡计算复杂度分析**

能力线

- **均衡目标分析能力**：问题定义+建模分析
- **均衡计算求解能力**：计算复杂性理论结合能力+算法设计实现能力

价值面

- **算法博弈论给予启示**：均衡可实现性和计算性

① 目标低效，设法避免

② 难以计算，近似求解

感谢大家认真听讲

兴军亮

jlxing@nlpr.ia.ac.cn

2024年4月8日