南京大学
人工智能学院
SCHOOL OF ARTIFICIAL INTELLIGENCE, NANJING UNIVERSITY

第四部分： 多Agent决策

章宗长

2023年5月16日

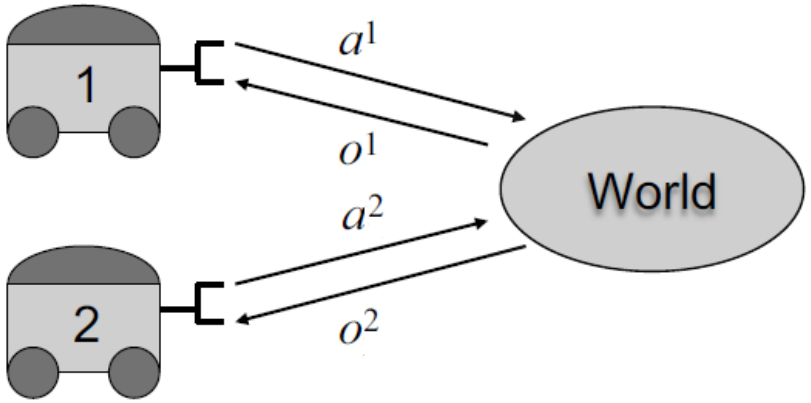
内容安排

分布式规划

- 规划的基础知识
- **分布式规划的决策模型**
- 分布式规划的离线算法
- 分布式规划的在线算法

分布式规划

- 分布式规划的过程
  - 输入问题的模型表示，通过算法输出Agent所能执行的动作序列
  - 不一定是分布式的，但要求Agent能够分布式地执行规划的结果



由两个Agent构成的分布式规划问题

- Agent之间是相互独立的，它们获得的信息也是相互独立的
  - 每个Agent只能观察到世界状态的某个局部

确定性/不确定性的分布式规划

- 确定性的分布式规划
  - 确定性环境：每个Agent动作的效果都是可以预测的
  - 这类问题本质上和经典的规划问题一致
    - 只是在状态和动作的维度上扩展为多个Agent
    - 在计算复杂度上与传统的经典规划问题相当

目前的主流研究方向，本节重点

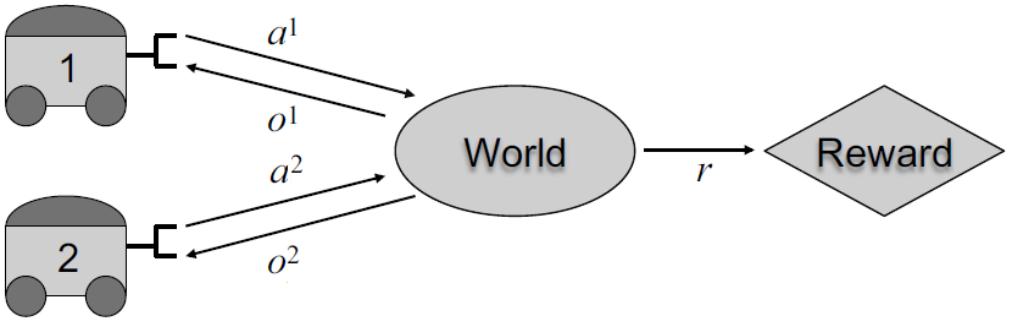
- 不确定性的分布式规划
  - 不确定性环境：动作可能有多种效果，需要有反馈信息来预测动作实际产生的效果
  - 需要在规划的过程中考虑每个动作各种可能的效果
  - 在执行阶段从环境中获取动作的反馈，指导后续动作的执行

Dec-POMDP模型

- 一个Dec-POMDP问题可以形式化地建模为
  - 有限的Agent集合I
  - 有限的状态集合$\mathcal{S}$
  - Agent i可采取的动作的集合$\mathcal{A}^i$
  - 状态转移函数$T(s'|s,\mathbf{a})$
    - 在状态s下采取联合动作$\mathbf{a}$后转移到新的状态s'的概率
  - 奖励函数$R(s,\mathbf{a})$
    - 在状态s下采取联合动作$\mathbf{a}$后所有Agent获得的立即奖励
  - Agent i可获得的观察的集合$\mathcal{O}^i$
  - 观察函数$\Omega(\mathbf{o}|s',\mathbf{a})$
    - 采取联合动作$\mathbf{a}$转移到新的状态s'后，获得联合观察$\mathbf{o}$的概率

Dec-POMDP模型的特点

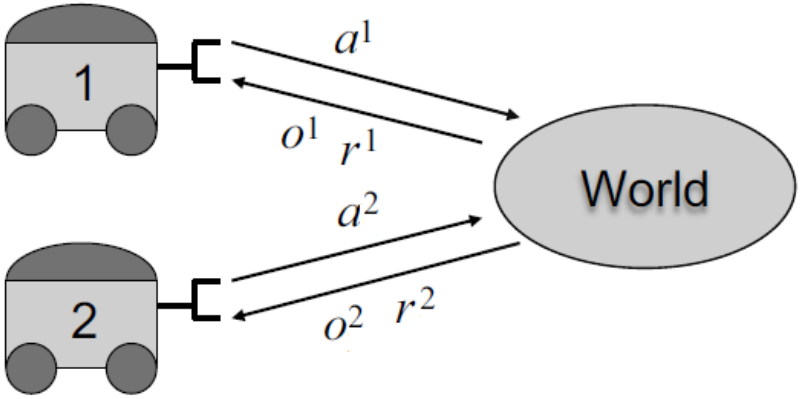
- Dec-POMDP模型是POMDP在多Agent决策问题上的自然拓展
  - 当Agent的数量为1时，等价于单Agent的POMDP模型
  - 每个Agent不仅要考虑动作效果的不确定性和环境状态感知的不确定性，还要考虑其他Agent行为上的不确定性
- 所有Agent共享一个奖励函数
  - Agent之间是完全合作的关系，具有共同的目标



由两个Agent构成的Dec-POMDP问题

Dec-POMDP的超类：POSG

- 部分可观察的随机博弈（Partially Observable Stochastic Game, POSG）
- 不同Agent可以有不同的奖励函数
  - Agent之间不一定是完全合作的关系，不一定有共同的目标



由两个Agent构成的POSG问题

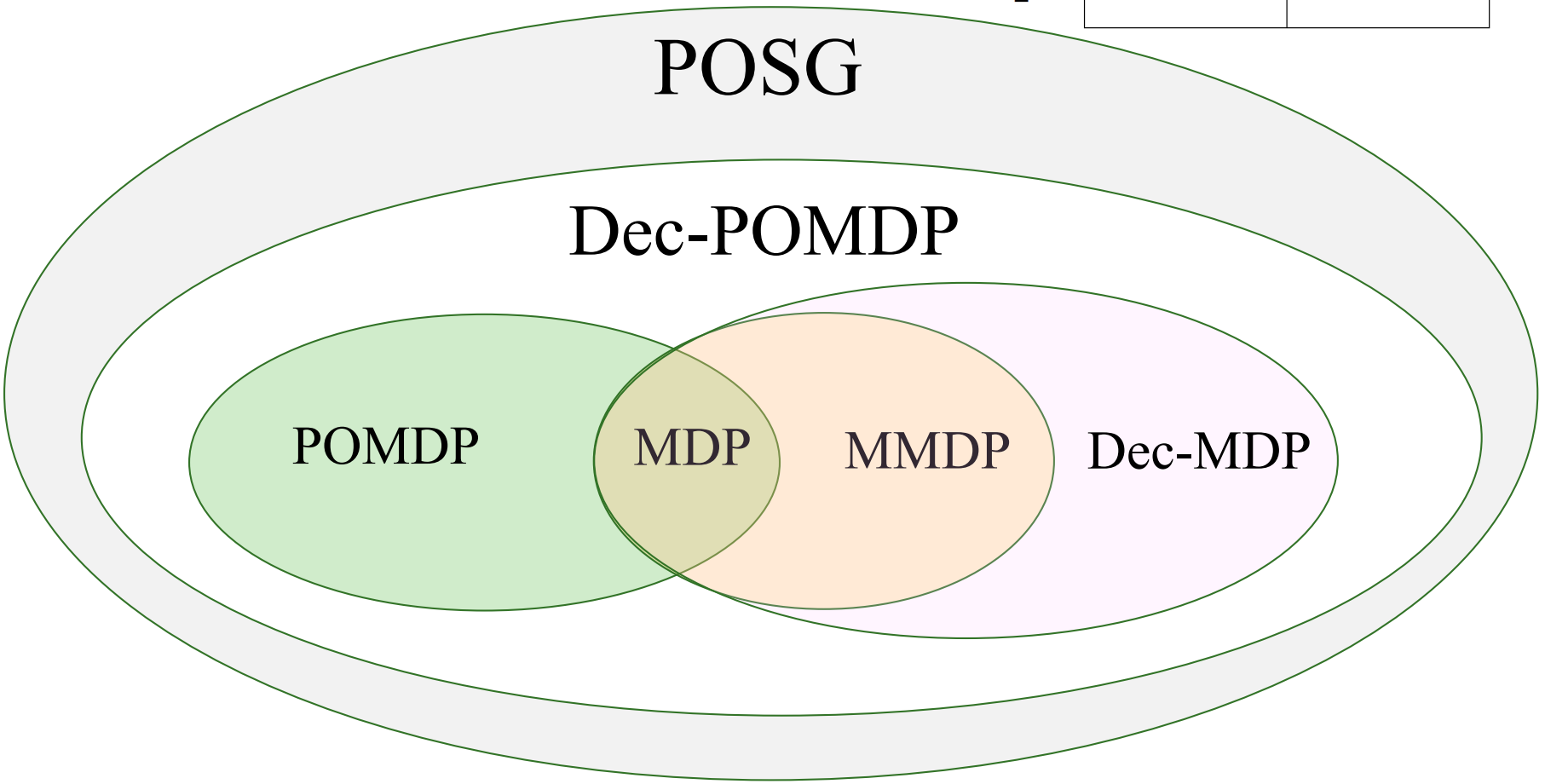
Dec-POMDP的子类：Dec-MDP、MMDP

- Dec-MDP是有联合完全观察的Dec-POMDP
  - Dec-POMDP的一个子类
  - Dec-POMDP不一定是联合完全观察的
  - 联合完全观察：由所有Agent的观察组合在一起能得到环境的状态信息

- 多Agent的MDP（Multiagent MDP, MMDP）
  - Dec-MDP的一个子类
  - 每个Agent都可以完全观察到环境的真实状态

Dec-POMDP与其他模型之间的关系

Number of Agents	Observability: Perfect	Observability: Imperfect
Multiple	MMDP	Dec-POMDP
Single	MDP	POMDP



Dec-POMDP模型详解

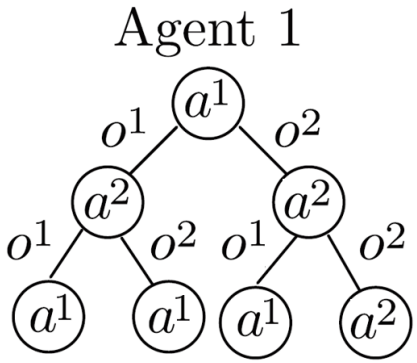
策略

- Agent i的策略$\mathbf{q}^i$为其观察的历史序列到动作的一个映射
- 联合策略$\mathbf{q}$为所有Agent策略构成的一个向量：

$$\mathbf{q} = [\mathbf{q}^1 \quad \mathbf{q}^2 \quad \dots \quad \mathbf{q}^n]^{\mathrm{T}}$$

- 策略树是Dec-POMDP最常见的策略表示形式
  - 每个Agent的策略是一棵独立的策略树
  - 联合策略是所有Agent的策略树的集合

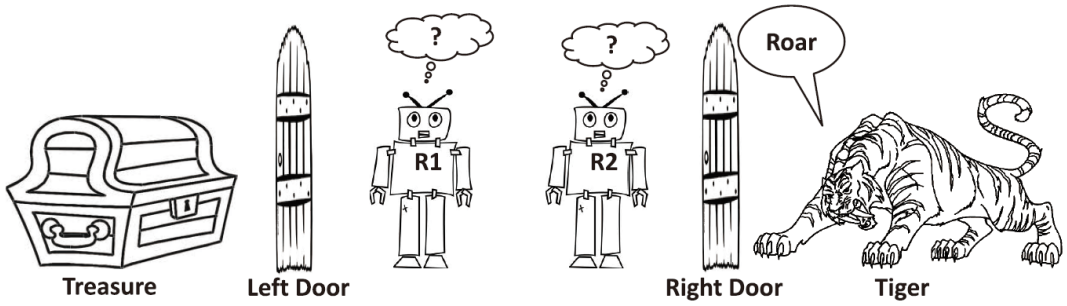
树的节点代表一个将被执行的动作
树的边代表Agent执行动作后可能获得的观察



- 有限状态机是另一类更加一般的策略表示

例子1：分布式老虎问题

- 分布式老虎问题：有两个Agent，通过听老虎的叫声判断它的位置，尽可能打开没有关老虎的那扇门
  - 状态集合$S = \{\text{Tiger}_{\text{Left}}, \text{Tiger}_{\text{Right}}\}$
  - Agent i的动作集合$\mathcal{A}^i = \{\text{Open}_{\text{Left}}, \text{Open}_{\text{Right}}, \text{Listen}\}$
  - Agent i的观察集合$\mathcal{O}^i = \{\text{Roar}_{\text{Left}}, \text{Roar}_{\text{Right}}\}$

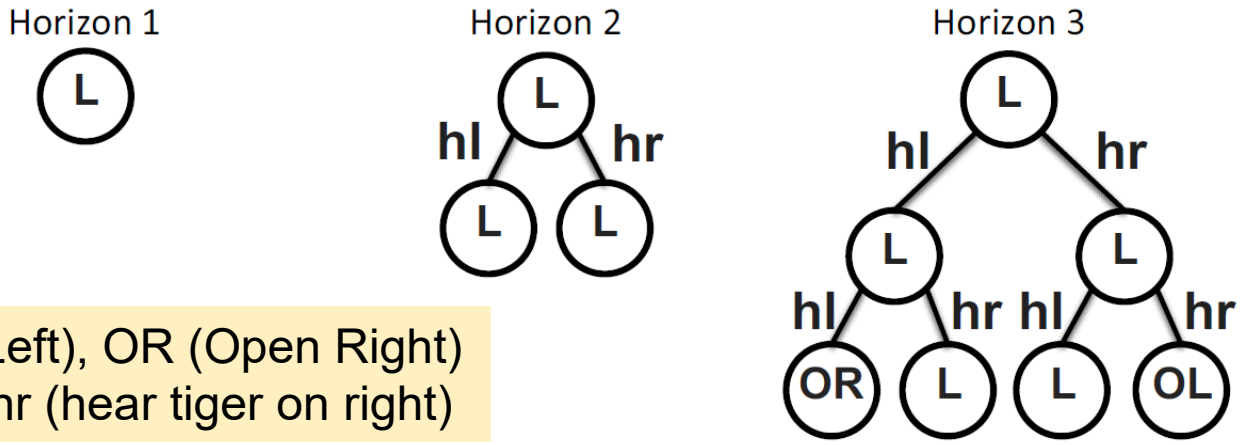


- 状态转移函数
  - 所有Agent均执行动作Listen后，老虎的位置不变
  - 只要有一个Agent执行了开门的动作，老虎会等概率地放置在两扇门后

- 观察函数
  - 建模Agent监听老虎叫声的准确度
- 奖励函数
  - 共同打开无老虎的门奖励最高
  - 即使打开的是有老虎的门，共同行动受到的伤害更小

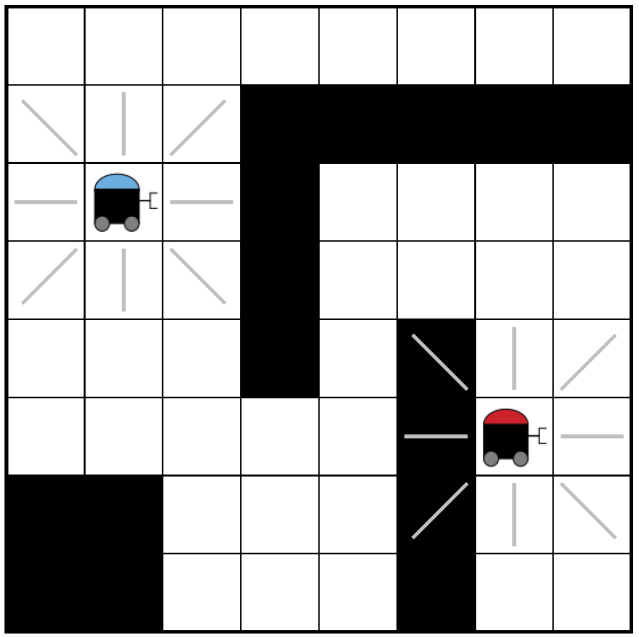
Agent之间必须合作

- 初始状态分布$b_0 = [0.5 \quad 0.5]^{\mathrm{T}}$
- 策略树
  - 1~3步



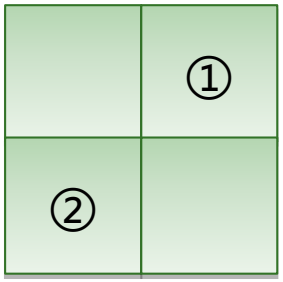
L (Listen), OL (Open Left), OR (Open Right)
hl (hear tiger on left), hr (hear tiger on right)

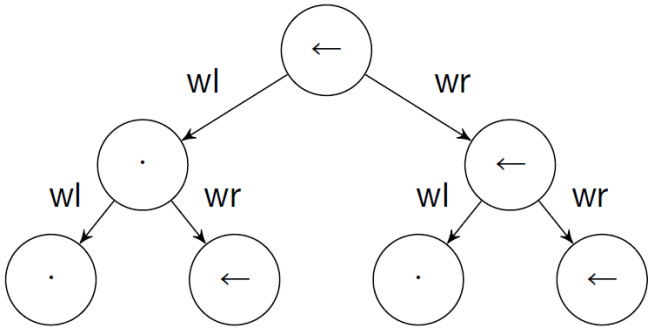
例子2：机器人导航问题



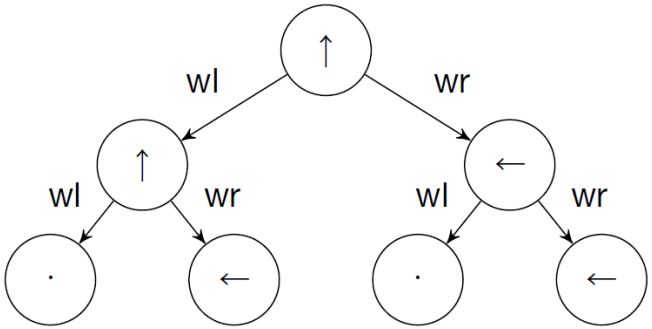
- 状态
  - 两个机器人的位置
- 动作
  - 上下左右、原地不动
- 状态转移函数
  - 以0.6的概率朝动作方向移动一格
  - 分别以0.1的概率朝其他3个方向移动一格或原地不动
- 观察函数
  - 能准确感知到相邻方格内的物体
- 奖励函数
  - 当两个机器人处在同一个方格时，收到奖励1，否则收到奖励0
- 初始状态
  - 如图所示
- 目标
  - 让两个机器人尽快碰面

- 策略树
  - 3步



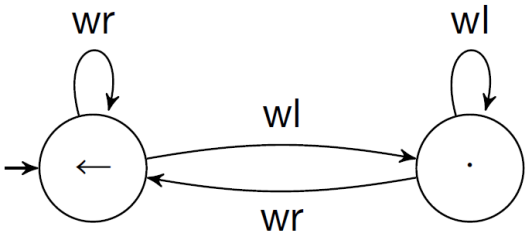


(a) Agent 1.

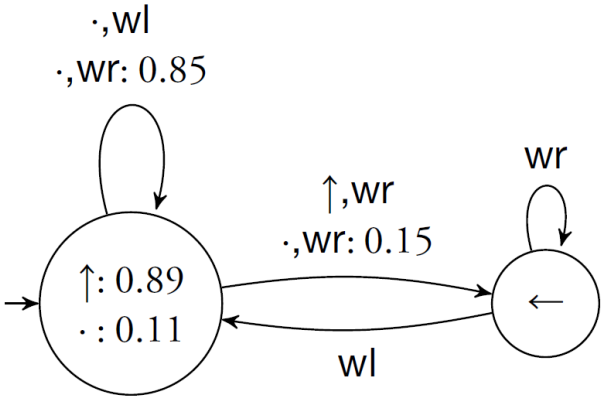


(b) Agent 2.

- 有限状态机
  - 2个节点



(a) Agent 1.



(b) Agent 2.

值函数

- 在给定一个状态s的情况下，联合策略$\mathbf{q}$的值函数：

$$V(s,\mathbf{q}) = \mathbb{E}_{\mathbf{a}\sim\mathbf{q}}\left[\sum_{t=0}^{T-1}\gamma^t R(s_t,\mathbf{a}_t) \mid s_0 = s\right]$$

- 在给定一个状态分布b的情况下，联合策略$\mathbf{q}$的值函数：

$$V(b,\mathbf{q}) = \mathbb{E}_{s\sim b,\,\mathbf{a}\sim\mathbf{q}}\left[\sum_{t=0}^{T-1}\gamma^t R(s_t,\mathbf{a}_t) \mid s_0 \sim b\right]$$

- 两者的关系式：

$$V(b,\mathbf{q}) = \sum_{s\in S} b(s)V(s,\mathbf{q})$$

值函数（续）

- 联合策略$\mathbf{q}$的值函数满足贝尔曼等式：

$$V(s,\mathbf{q}) = R\left(s,\mathbf{a}_{\mathbf{q}}\right) + \gamma \sum_{s',\mathbf{o}} T(s'|s,\mathbf{a})\Omega(\mathbf{o}|s',\mathbf{a})V(s',\mathbf{q}_{\mathbf{o}})$$

根据$\mathbf{q}$的根节点给出的联合动作

执行动作后根据$\mathbf{q}$和所获得的观察信息$\mathbf{o}$给出的联合子策略

- 求解Dec-POMDP问题的目标是找到最优联合策略$\mathbf{q}^*$：

$$\mathbf{q}^* = \arg\max_{\mathbf{q}} V(b_0,\mathbf{q})$$

b_0：初始状态的概率分布

分布式规划

- 规划的基础知识
- 分布式规划的决策模型
- **分布式规划的离线算法**
- 分布式规划的在线算法

离线规划

- 基本原理：把Dec-POMDP问题的求解过程分为离线规划和在线执行两个阶段
- 离线规划
  - 求解器：输入一个Dec-POMDP问题，输出一个联合策略
  - 联合策略的要求：方便分配到每个Agent中进行执行
- 在线执行
  - 将每个Agent的策略分配到Agent的执行系统中
  - Agent根据自己所获得的局部信息来进行动作选择
- 求解范式：集中式规划、分布式执行

精确算法1：暴力枚举法

- 基本思想：枚举生成所有可能的策略树，并对每个联合策略计算V值，然后从中选择出最优的联合策略

- 所有可能的联合策略的个数：
  - 极其大：随着决策步数T呈双指数式增长

$$O\left(m \times |\mathcal{A}^{\max}|^{\frac{|\mathcal{O}^{\max}|^{T}-1}{|\mathcal{O}^{\max}|-1}}\right)$$

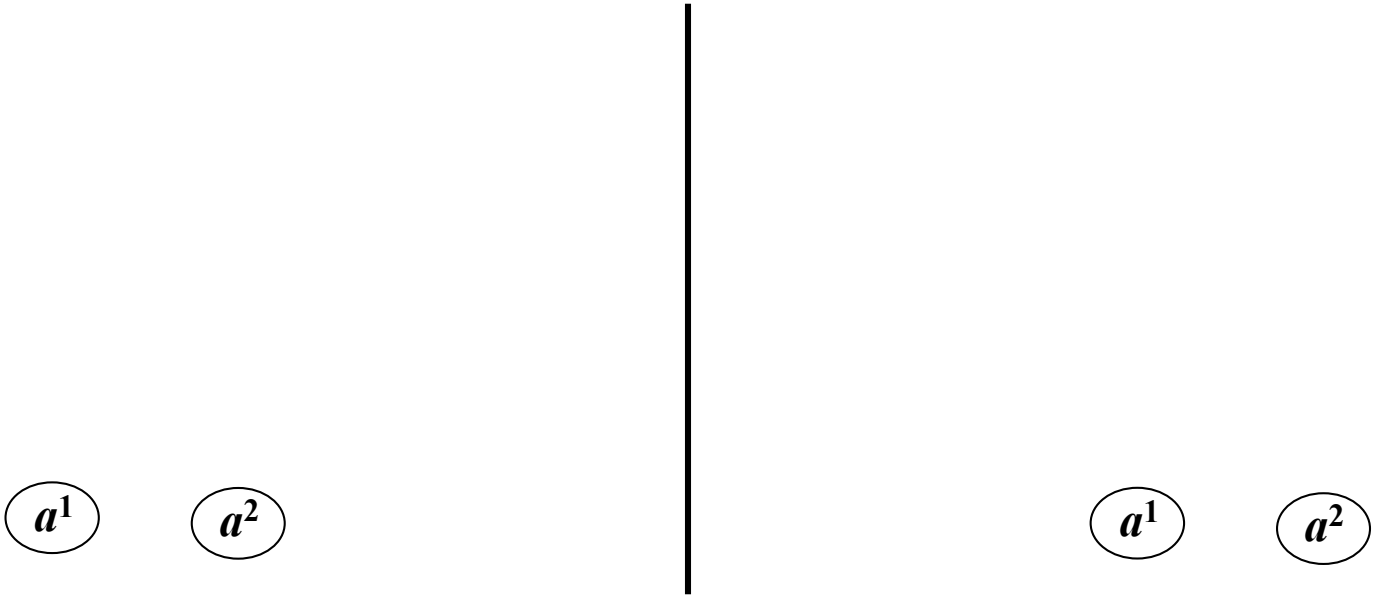
- 结论：暴力枚举法完全不具有实用性
  - 哪怕最小规模的Dec-POMDP问题都求解不了
  - 巨大的联合策略空间是求解Dec-POMDP问题的关键障碍

暴力枚举法：联合策略的个数（T=1）

$$O\left(m \times |\mathcal{A}^{\max}|^{\frac{|\mathcal{O}^{\max}|^T-1}{|\mathcal{O}^{\max}|-1}}\right)$$

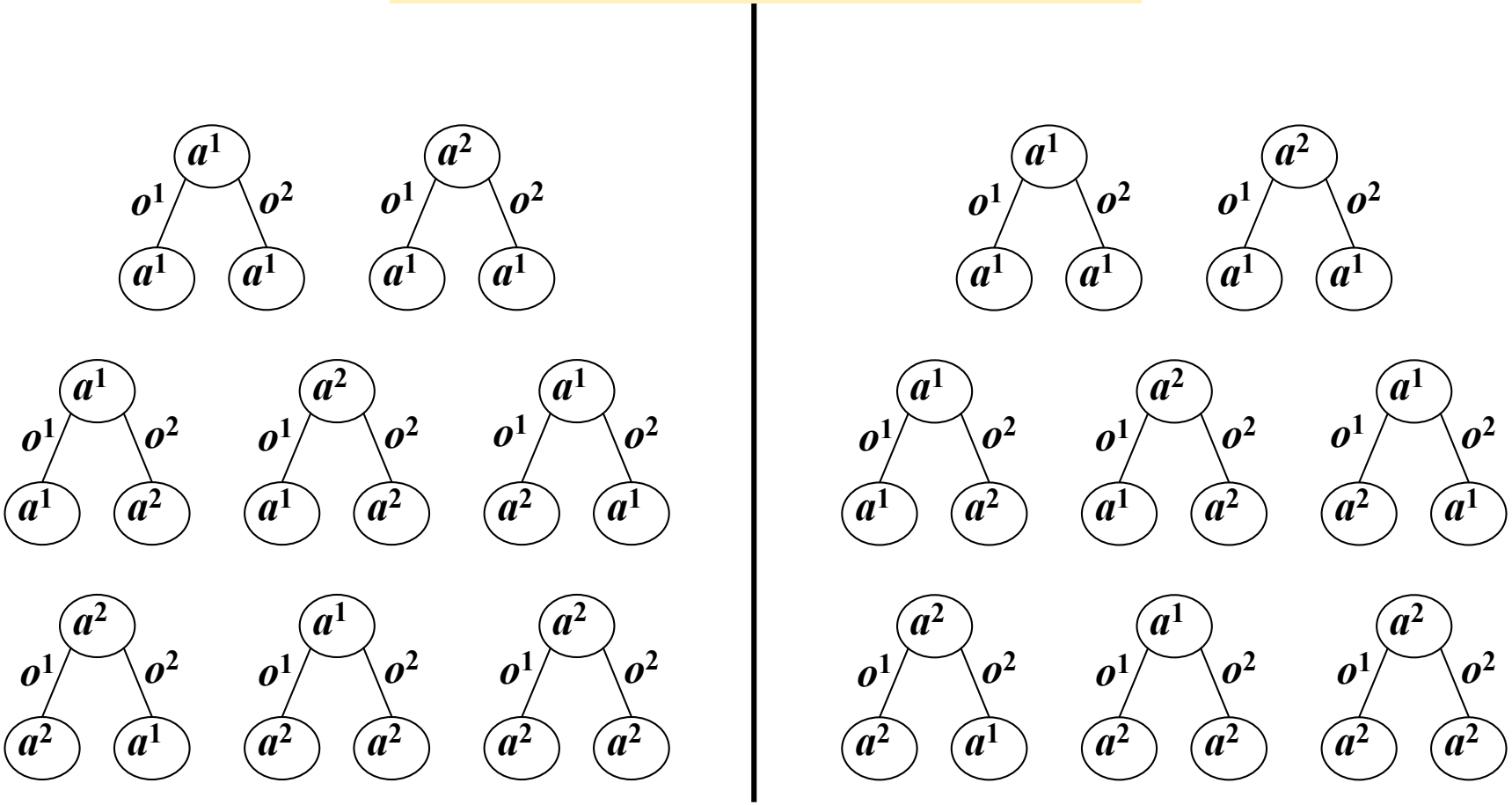
例子：两个Agent，每个Agent的动作数、观察数均为2

$$m \times |\mathcal{A}^{\max}|^{\frac{|\mathcal{O}^{\max}|^T-1}{|\mathcal{O}^{\max}|-1}} = 2 \times 2^{\frac{2^T-1}{2-1}}$$



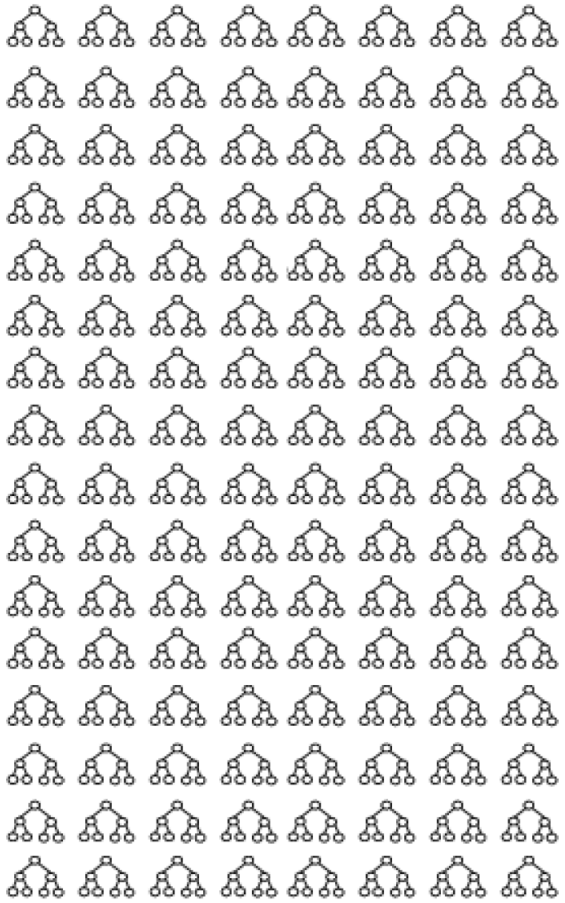
暴力枚举法：联合策略的个数（T=2）

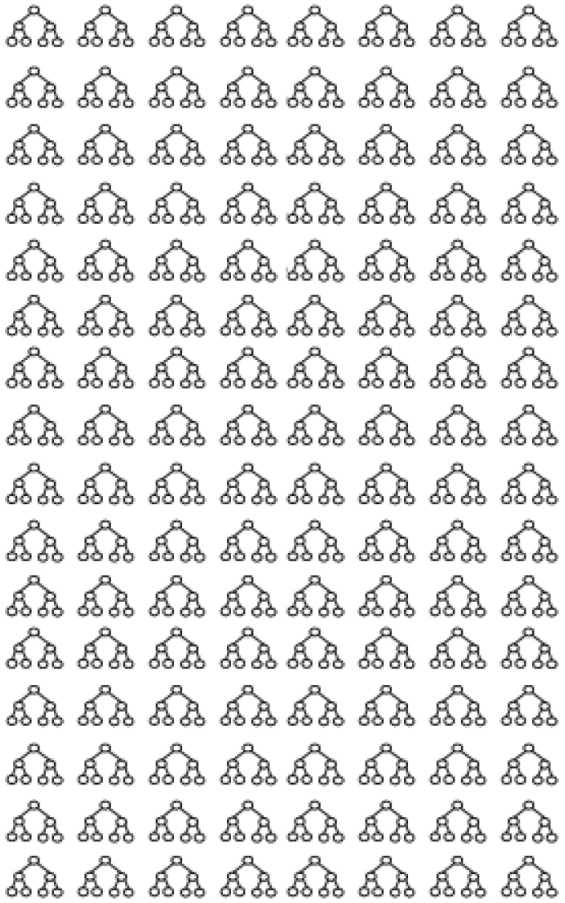
$$m \times |\mathcal{A}^{\max}|^{\frac{|\mathcal{O}^{\max}|^T-1}{|\mathcal{O}^{\max}|-1}} = 2 \times 2^{\frac{2^T-1}{2-1}}$$



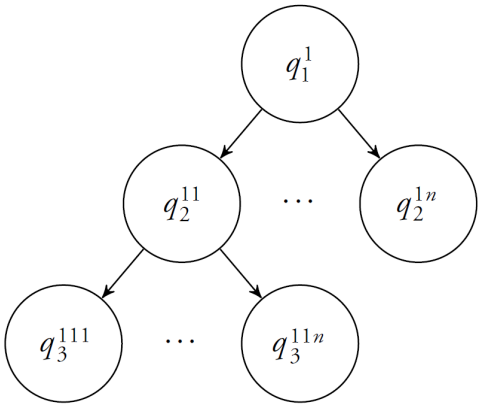
暴力枚举法：联合策略的个数（T=3）

$$m \times |\mathcal{A}^{\max}|^{\frac{|\mathcal{O}^{\max}|^T-1}{|\mathcal{O}^{\max}|-1}} = 2 \times 2^{\frac{2^T-1}{2-1}}$$





精确算法2：多Agent的A*算法



- 搜索树
  - 每一个节点表示一个给定步数的联合策略

以节点$\mathbf{q}_3^{11n}$为例，$11n$是其索引号，其中11是其父节点的索引号，3表示该策略的步数

$$n = \mathcal{O}\left(|\mathcal{A}^{\max}|^{m|\mathcal{O}^{\max}|}\right)$$

- 特点：OPEN表中的每个节点$\mathbf{q}_t^k$都有一个优先值$V(\mathbf{q}_t^k)$

$$V(\mathbf{q}_t^k) = g(\mathbf{q}_t^k) + h(\mathbf{q}_t^k)$$

越大优先级越高

执行节点$\mathbf{q}_t^k$中的t步策略能获得的累积奖励

启发式函数：把节点$\mathbf{q}_t^k$展开为T步策略后，能从余下的$T-t$步决策中获得的额外奖励

- 常用的启发式函数：忽略Dec-POMDP的部分可观察性后退化成的MMDP问题的最优值函数（可容纳最优）

精确算法3：动态规划法

- 基本思想
  - 从最后一步开始逐步向前，计算出一个完整的联合策略
  - 生成策略树的时候，从叶子节点开始，自底向上一步步构建，直至构建出高度为T的策略树

- 由t步策略树构建$t+1$步策略树的过程
  - 备份（Backup）：给定t步策略树的情况下，生成$t+1$步策略树
  - 评价（Evaluate）：对$t+1$步的所有联合策略进行值评价
  - 裁剪（Prune）：将所有可以删减的$t+1$步策略树剔除

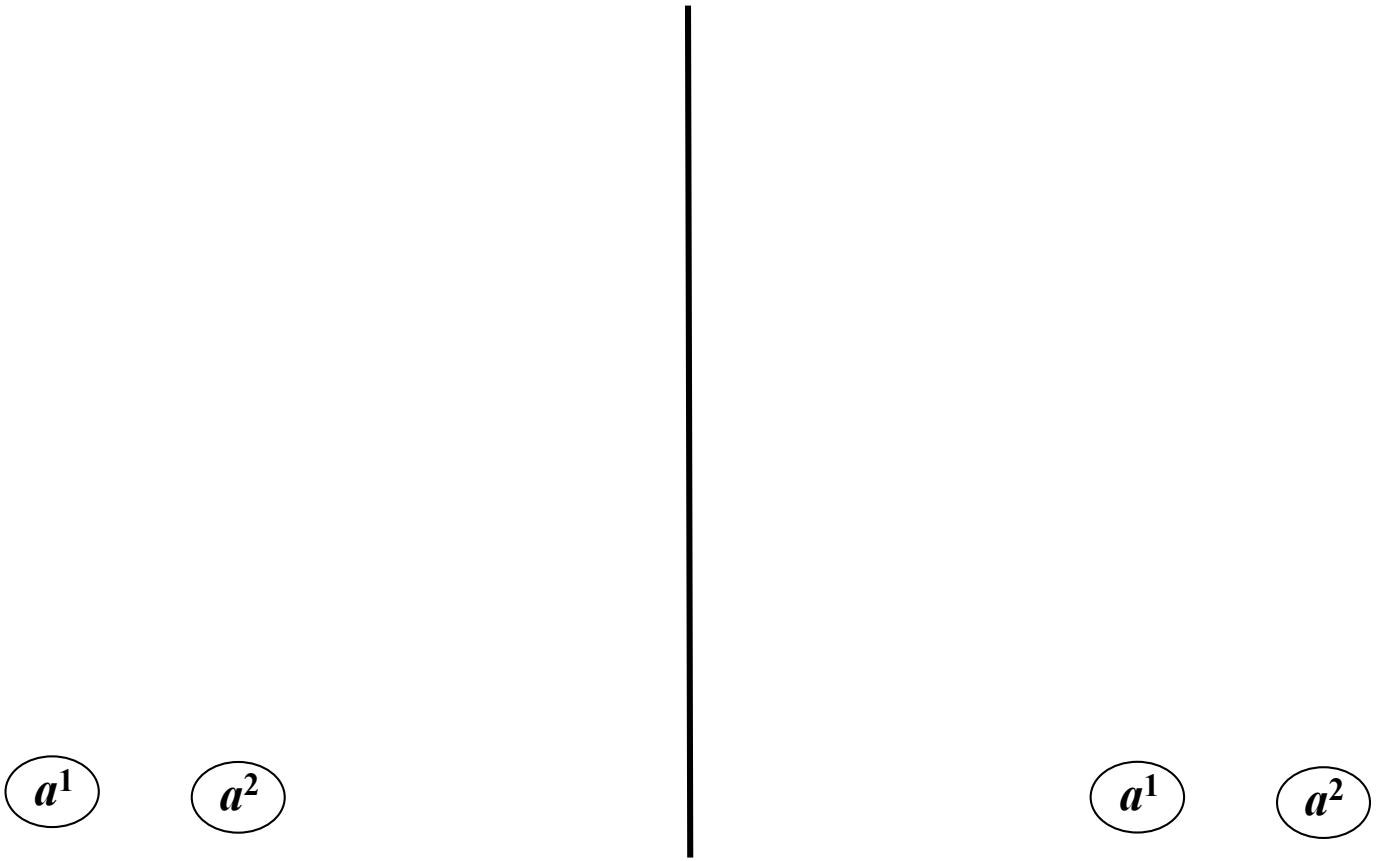
- 每一步生成的策略树数量：

$$|Q_{t+1}^i| = O\left(|\mathcal{A}^i||Q_t^i|^{|\mathcal{O}^i|}\right)$$

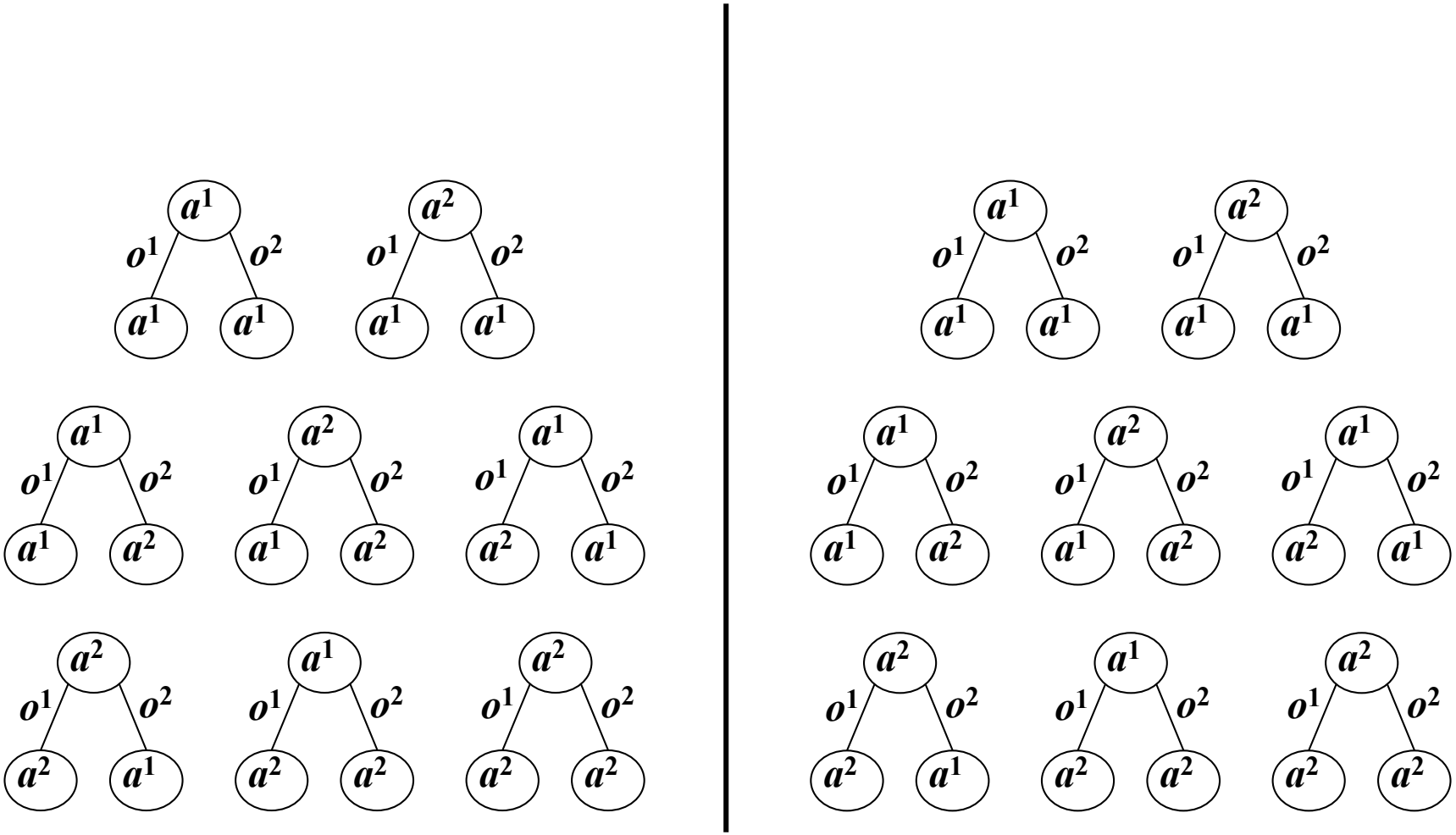
Q_t^i和Q_{t+1}^i分别为Agent i的t步和$t+1$步策略树集合

动态规划法示例：联合策略的个数（T=1）

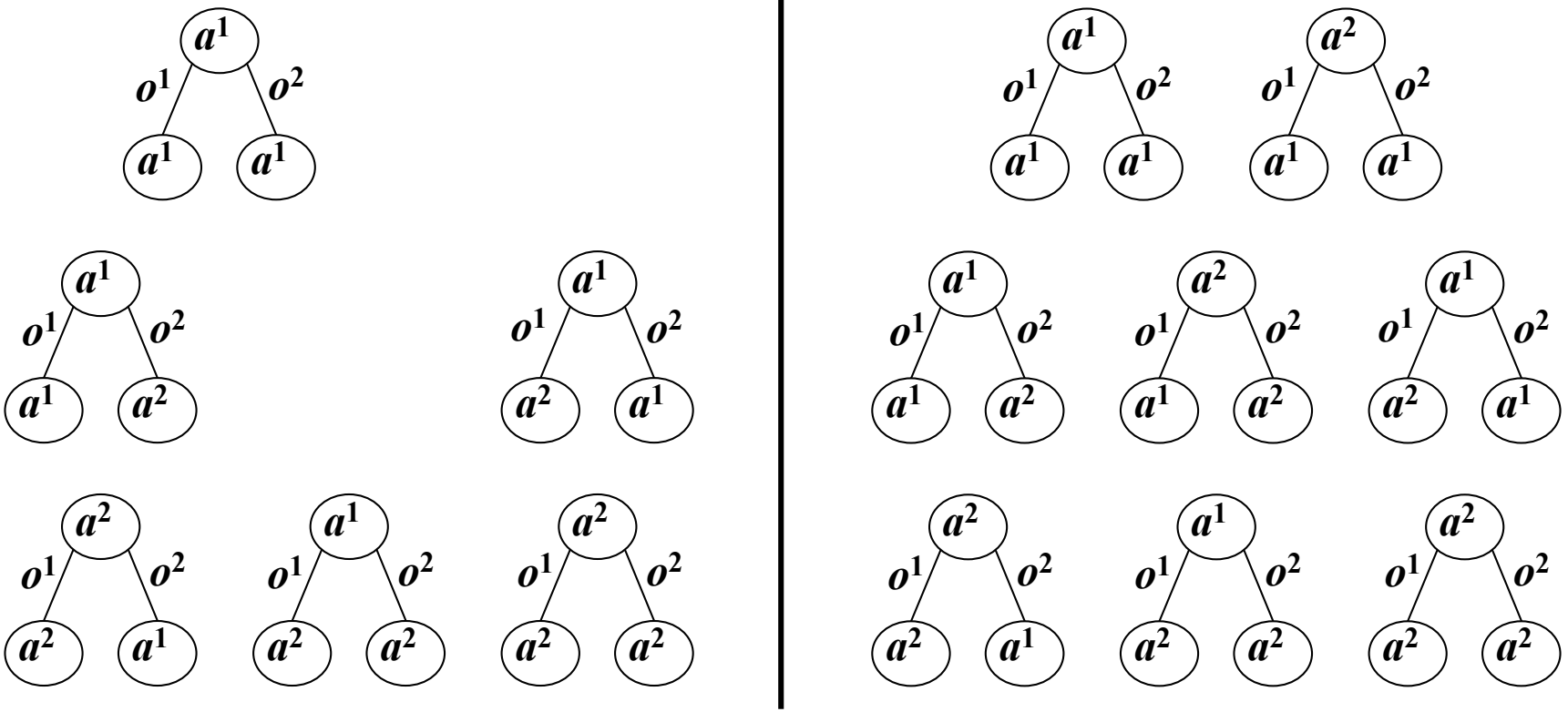
例子：两个Agent，每个Agent的动作数、观察数均为2



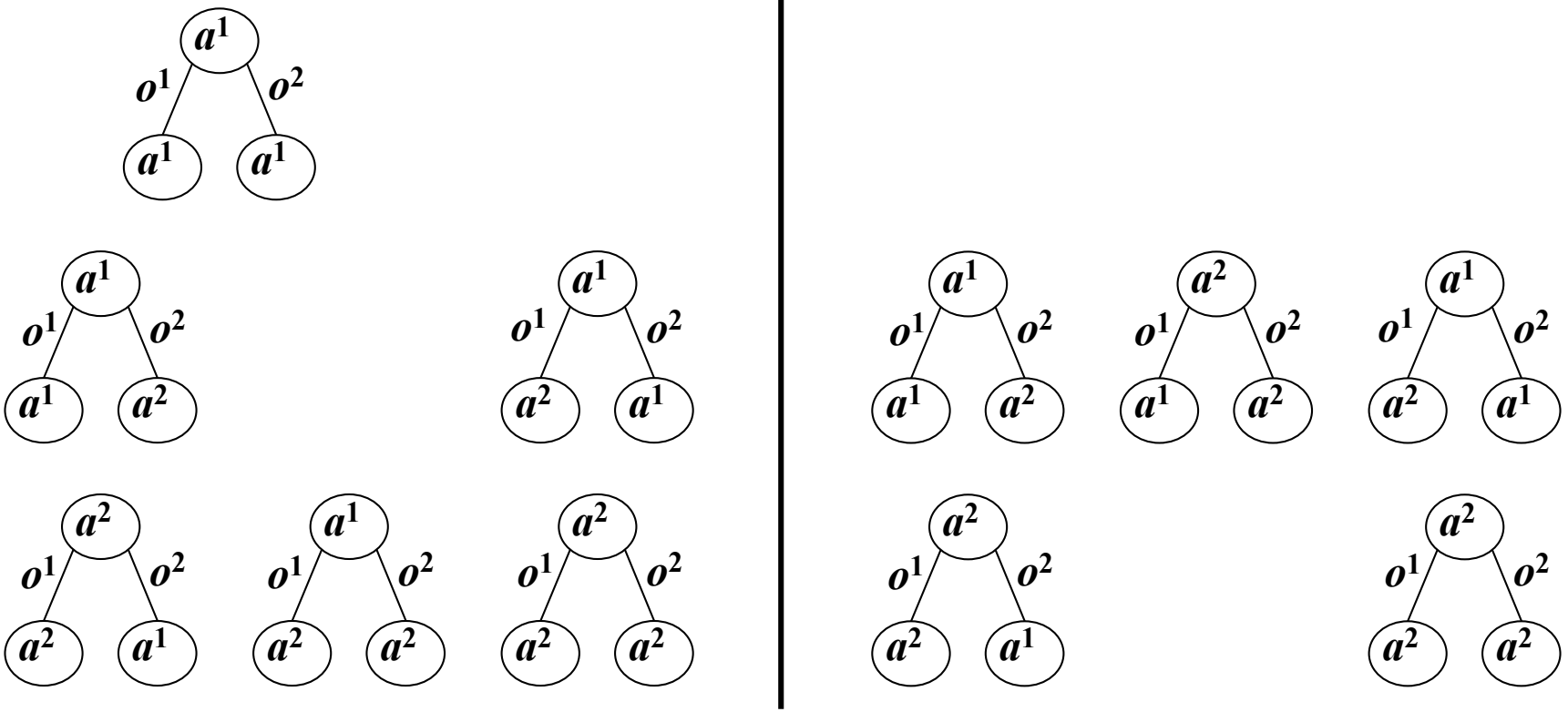
动态规划法示例：联合策略的个数（T=2）



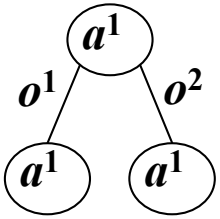
动态规划法示例：联合策略的个数（T=2）

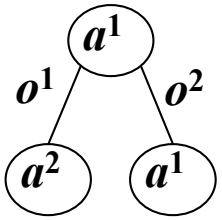


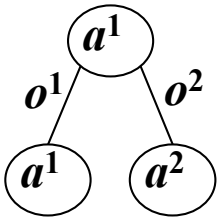
动态规划法示例：联合策略的个数（T=2）

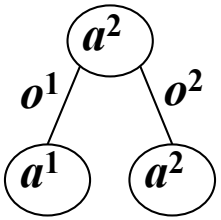


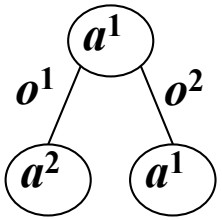
动态规划法示例：联合策略的个数（T=2）

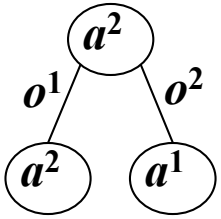


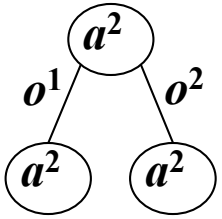


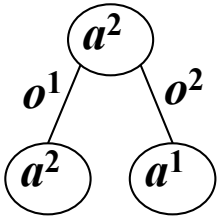


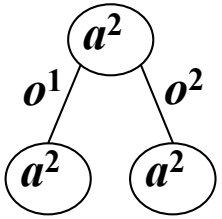




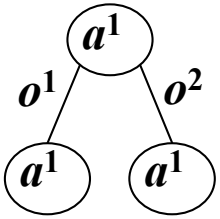


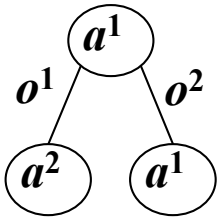


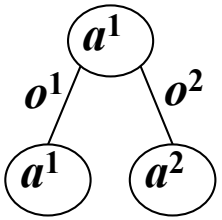


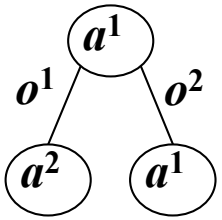


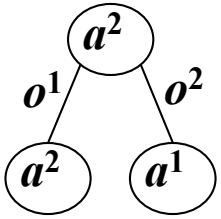
动态规划法示例：联合策略的个数（T=2）

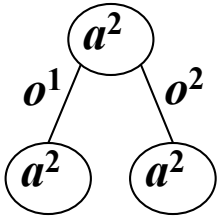


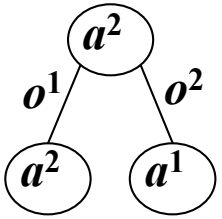


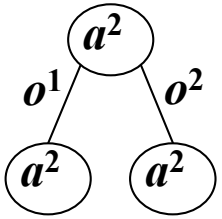




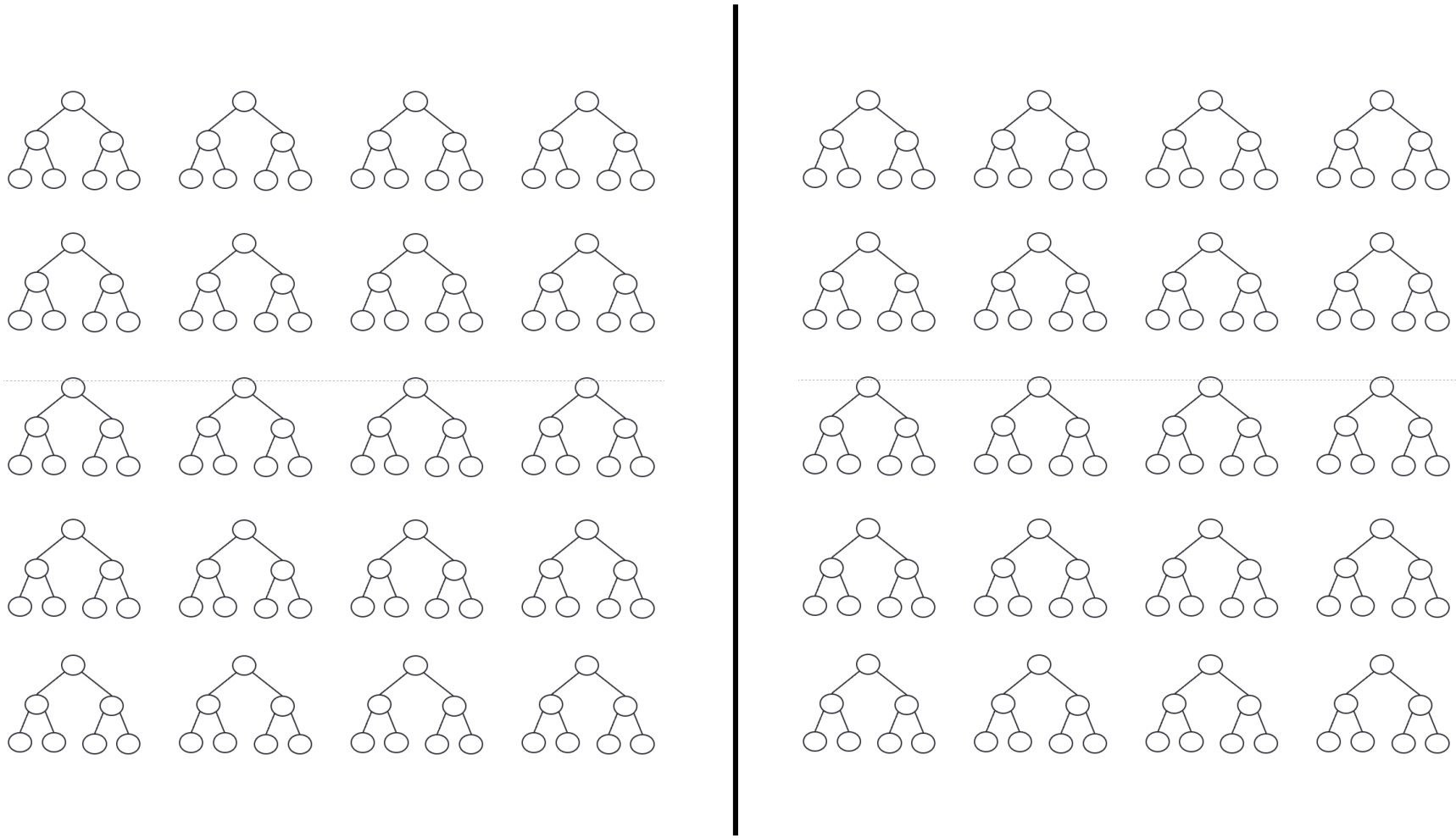






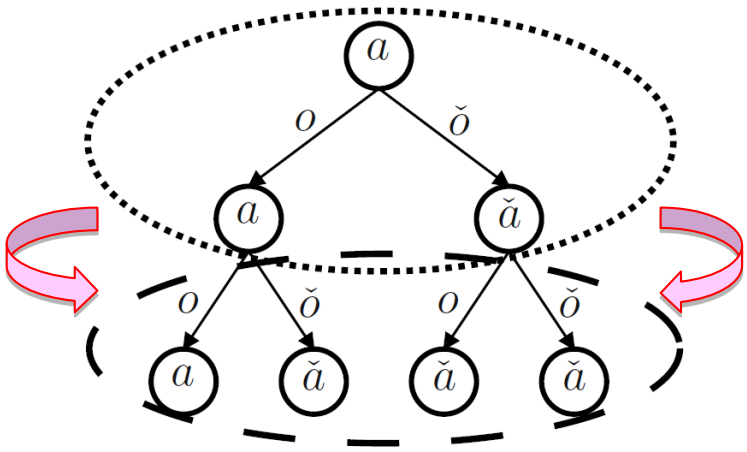


动态规划法示例：联合策略的个数（T=3）

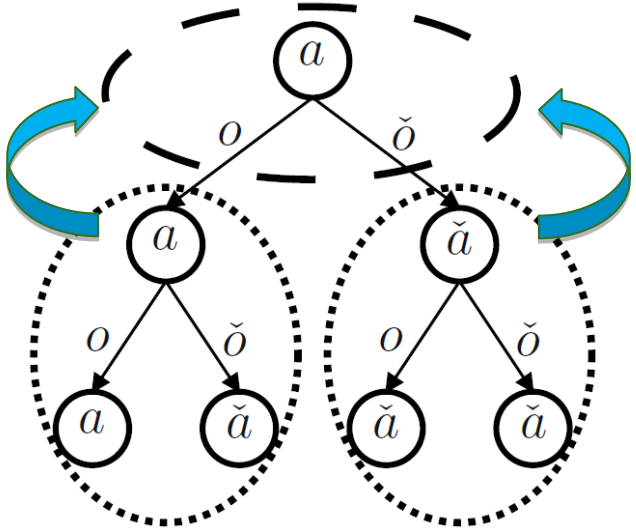


多Agent的A*算法 vs. 动态规划法

- 多Agent的A*算法
  - 构建策略树的方式：自顶向下



- 动态规划法
  - 构建策略树的方式：自底向上

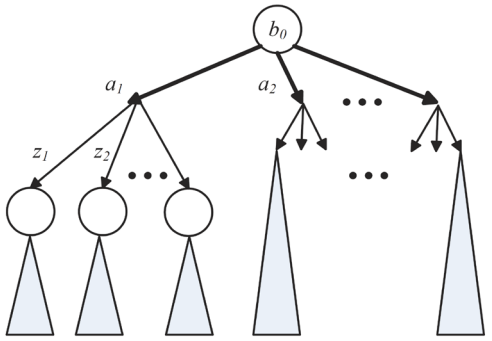


近似算法1：基于联合均衡的策略搜索算法

- 基本思想：寻找纳什均衡点的联合策略
- 假设：对每个Agent i而言，其他Agent的策略集π^{-i}给定
  - Agent i的信念状态可以定义为系统的状态和其他Agent策略集的联合概率分布$b^i \in \Delta(s \times \pi^{-i})$
- 具体做法
  - 每次选定一个Agent i，在固定其他Agent策略的情况下，对Agent i的策略进行改进
  - 改进的方法：生成一系列的候选策略，从中选出Agent i的最好策略
  - 保证所有Agent都有等概率的机会被选择并进行策略的改进，直到所有Agent的策略都无法改进为止

近似算法2：基于信念点的动态规划算法

- 基本思想
  - 使用动态规划自底向上构建策略树
  - 只考虑巨大信念状态空间的可达信念点



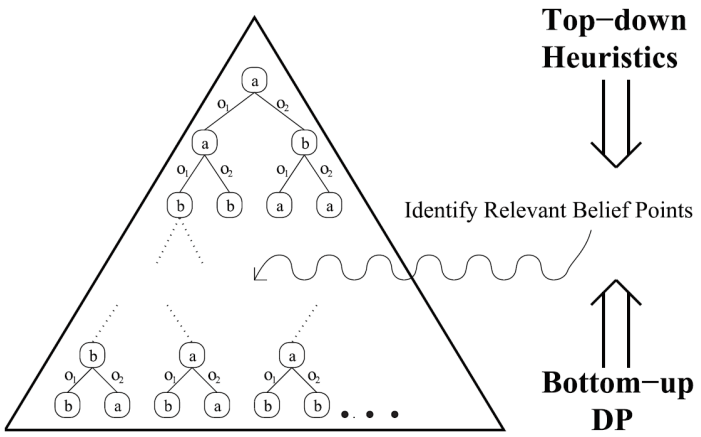
POMDP中的可达信念点示例

- 由t步策略树构建$t+1$步策略树的过程
  - 备份（Backup）：给定t步策略树的情况下，生成$t+1$步策略树
  - 评价（Evaluate）：生成一系列的可达信念点，评价$t+1$步的所有联合策略在这些点的值
  - 裁剪（Prune）：只保留在至少一个可达信念点上有最优值的策略树

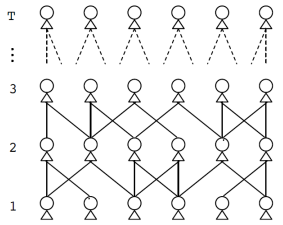
- 主要问题：可达的信念点可能非常的多，在最坏情况下，需要考虑的策略树依然呈双指数式的增长

近似算法3：内存受限的动态规划算法

- 基本思想
  - 将自顶向下的搜索过程和自底向上的动态规划相结合来生成策略树
  - 在自顶向下的搜索中只考虑状态的分布，不考虑其他Agent的策略



- 由t步策略树构建$t+1$步策略树的过程
  - 备份（Backup）：给定t步策略树的情况下，生成$t+1$步策略树
  - 评价（Evaluate）：只生成固定个数的联合信念状态，评价$t+1$步的所有联合策略在这些点的值
  - 裁剪（Prune）：只保留在至少一个联合信念状态上有最优值的策略树
- 优点：策略树随求解的步数呈线性增长
  - 实际效果好，带动了实用性Dec-POMDP近似算法的研究



分布式规划

- 规划的基础知识
- 分布式规划的决策模型
- 分布式规划的离线算法
- **分布式规划的在线算法**

在线规划

- 基本原理：Agent在决策周期内一边进行规划一边执行动作
- 主要优势
  - Agent无需考虑所有的可能，而仅仅需要对当前遇到的情况进行规划
  - 可以随时改变模型或者规划算法来处理一些突发事件
- 主要劣势
  - 用于规划的时间往往非常的有限

仿真机器人足球

Dota2: OpenAI Five

误协调

- 在Dec-POMDP中的误协调现象
  - 每个Agent获得的信息是各自不同的局部信息
  - 根据这些不同的局部信息进行规划有可能出现误协调

甲、乙机器人根据自己获得的信息都觉得对方离球更近，对方更应该去抢球，造成的结果是这两个机器人都不去抢球

实际情况：甲、乙机器人离球都差不多近，由甲或者乙去抢球都是可以接受的

在线协调机制

- Dec-POMDP的在线规划算法需要考虑多Agent之间的协调与合作
  - 产生好的团队决策
- 避免误协调的解决办法
  - 在进行规划的时候考虑其他Agent可能采取的策略
- 情况1：Agent之间不能通信
  - 在线规划的时候只能对其他Agent的行为进行大致的预测
- 情况2：Agent之间可以通信
  - 通过通信进行信息共享和策略协调

情况1：预测其他Agent的行为

- 预测方式1：考虑其他Agent决策的最坏可能情况
  - 无论其他Agent的决策有多么糟糕，都能尽自己最大的可能来弥补
  - 按照最坏情况假设，甲、乙机器人都会同时去抢球
  - 典型的最坏假设在线规划算法：序列贝叶斯博弈近似算法

- 预测方式2：考虑其他Agent决策的平均可能情况
  - Agent会忽略一些局部的不一致信息，用等概率的方式取代该信息，从而达到信息的一致性
  - 通过一定的方式保证甲、乙机器人有等同概率去抢球
  - 典型的平均假设在线规划算法：可能联合信念推理算法

情况2：Agent之间通信

- Dec-POMDP中的隐式通信
  - 通过自己的动作改变其他Agent的观察

这里讨论显式通信

- 通信没有限制
  - 所有的Agent可以共享各自的观察信息
  - 分布式控制的Dec-POMDP变为了集中式控制的POMDP
- 通信受限
  - 物理条件的限制
  - 通信媒介的限制
  - 通信带宽的限制
  - ……

通信受限的在线规划

- 需要解决的几个问题
  - 如何在不通信的情况下也能进行决策？
  - 如何在能通信时只在必要时进行通信？
  - 在通信时如何最小化通信量？
- 在线规划算法往往利用通信来进行信息共享
- 通信方式
  - 广播式的通信：较简单
    - 主动通信：广播者自动发起通信
    - 被动通信：其他Agent请求广播者进行通信
  - 点对点的通信：更复杂

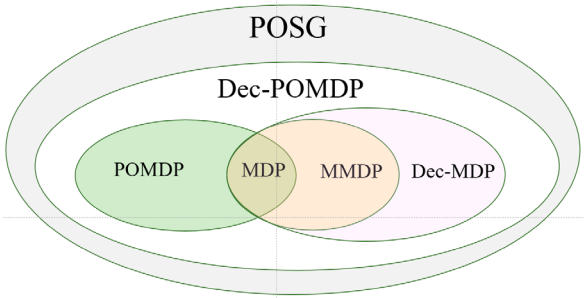
通信策略

- 定时广播
  - 实现简单，可能会浪费掉一些宝贵的通信资源
- 考虑决策过程的通信策略
  - 原理：预测通信前和通信后求解出的策略之间的差别
  - 基于策略的结构：策略有差别说明通信能导致策略的改变，但策略的不同不代表策略效果的不同
  - 基于策略的值评价：使用**通信的信息值**衡量一次通信对决策的价值

通信的信息值=通信后的策略值-通信前的策略值-通信代价

小结

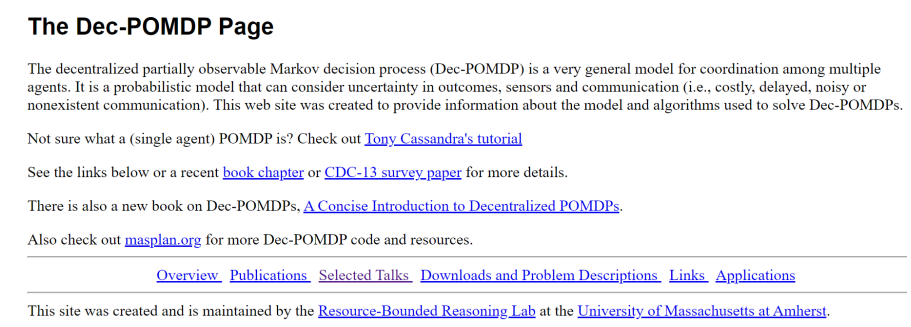
- 规划的基础知识
  - 经典规划（A*算法）、概率规划（动态规划算法）
- 分布式规划的决策模型
  - Dec-POMDP模型、策略、值函数



- 分布式规划的离线算法
  - 精确算法：暴力枚举法、多Agent的A*算法、动态规划法
  - 近似算法：基于联合均衡的策略搜索算法、基于信念点的动态规划算法、内存受限的动态规划算法
- 分布式规划的在线算法
  - 误协调
  - 在线协调机制
    - 预测其他Agent的动作
    - Agent之间通信：通信受限、通信方式、通信策略

资源

- http://rbr.cs.umass.edu/camato/decpomdp/



The Dec-POMDP Page

The decentralized partially observable Markov decision process (Dec-POMDP) is a very general model for coordination among multiple agents. It is a probabilistic model that can consider uncertainty in outcomes, sensors and communication (i.e., costly, delayed, noisy or nonexistent communication). This web site was created to provide information about the model and algorithms used to solve Dec-POMDPs.

Not sure what a (single agent) POMDP is? Check out Tony Cassandra's tutorial

See the links below or a recent book chapter or CDC-13 survey paper for more details.

There is also a new book on Dec-POMDPs, A Concise Introduction to Decentralized POMDPs.

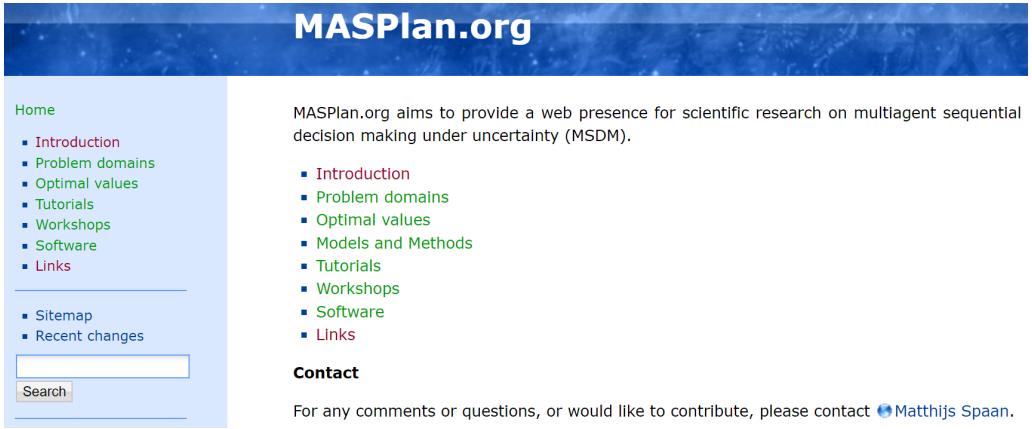
Also check out masplan.org for more Dec-POMDP code and resources.

Overview Publications Selected Talks Downloads and Problem Descriptions Links Applications

This site was created and is maintained by the Resource-Bounded Reasoning Lab at the University of Massachusetts at Amherst.

Chris Amato, MIT

- http://masplan.org/



MASPlan.org

Home
- Introduction
- Problem domains
- Optimal values
- Tutorials
- Workshops
- Software
- Links

- Sitemap
- Recent changes

Search

MASPlan.org aims to provide a web presence for scientific research on multiagent sequential decision making under uncertainty (MSDM).

- Introduction
- Problem domains
- Optimal values
- Models and Methods
- Tutorials
- Workshops
- Software
- Links

Contact

For any comments or questions, or would like to contribute, please contact Matthijs Spaan.

Matthijs Spaan, UT Delft

课后作业4-17~4-20

- 完成下列文档中的1~4题。

  http://users.isr.ist.utl.pt/~jmessias/easss14/Practical-part1.pdf

交第二次课后作业（第4部分）的截止时间为：

2023年5月30日