人工智能创新实践04

基于决策树和搜索的智能系统：井字棋

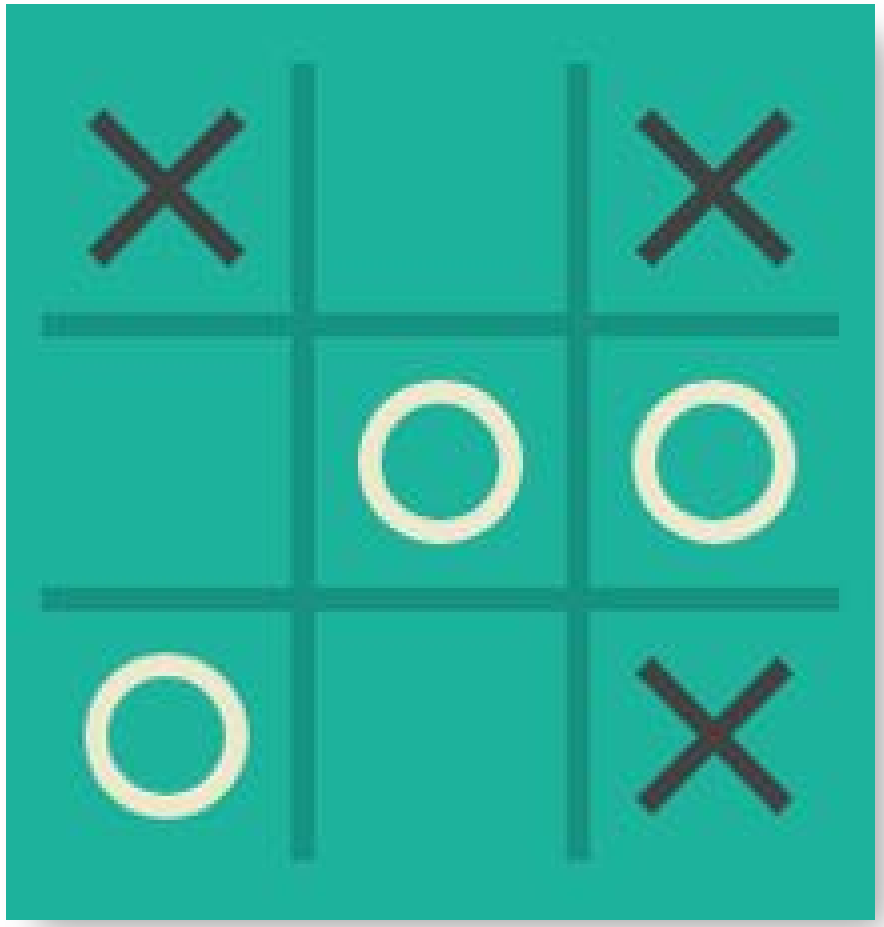
2019.7.16

北京大学 陈斌 gischen@pku.edu.cn

北京大学 PEKING UNIVERSITY

暑期学校 Summer School

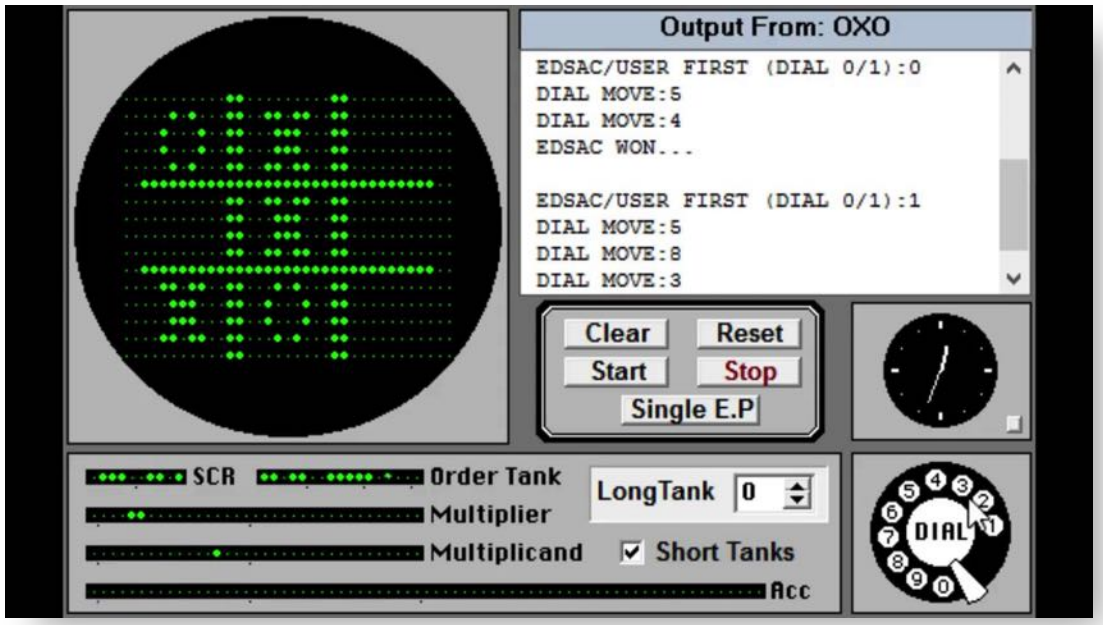
实例2：井字棋



- 井字棋(Tic-Tac-Toe)是由两个玩家轮流在3乘3的格上打自己的符号（圈或者叉）
- 最先以横、直、斜连成一线则为胜。

北京大学 陈斌 gischen@pku.edu.cn

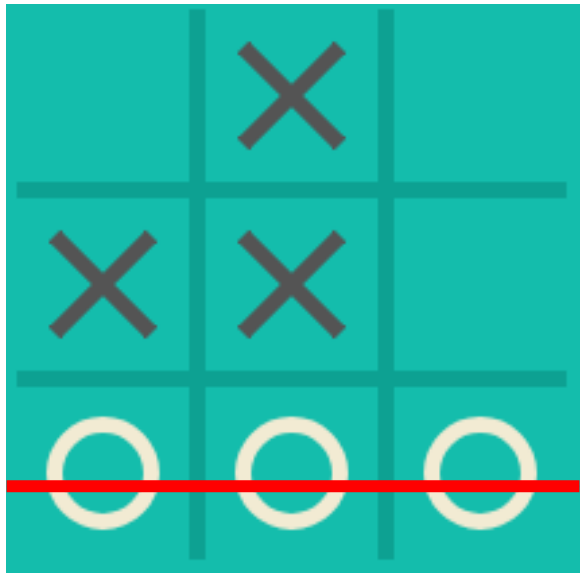
第一款电子游戏

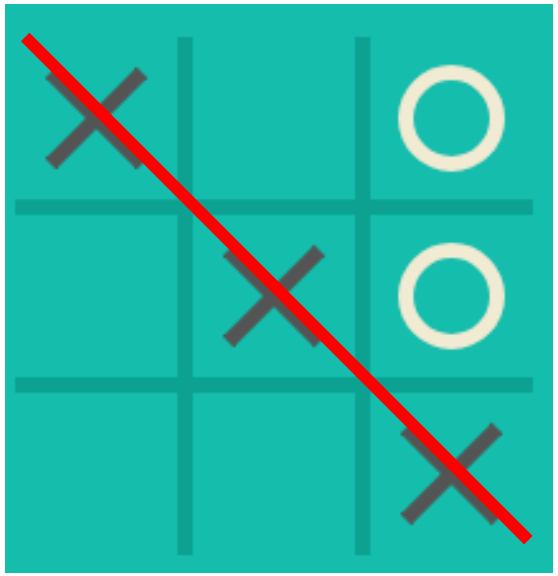


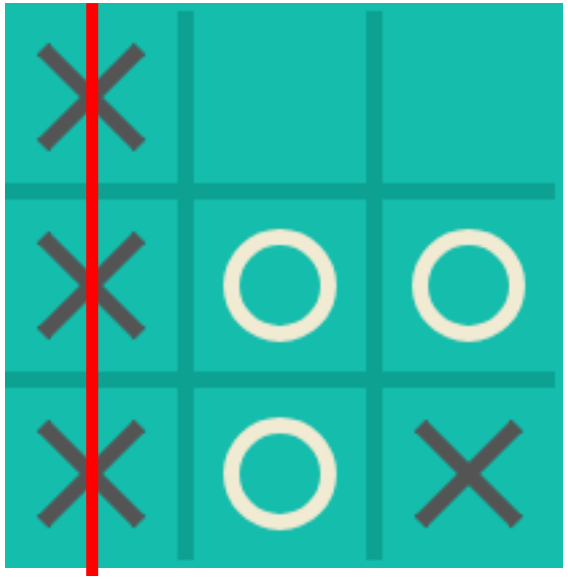
- 1952年，英国的计算机科学家Alexander S. Douglas开发出了井字棋游戏《Noughts and Crosses》

游戏目标

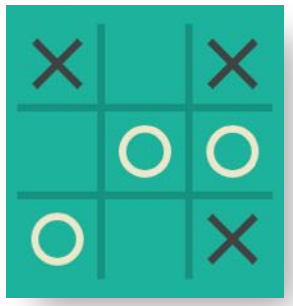
- 让自己的三个棋子连在一起
- 阻止对方的三个棋子连在一起

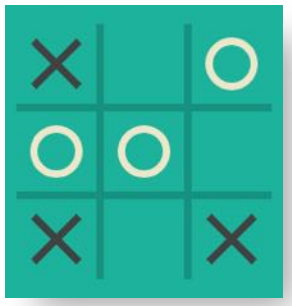


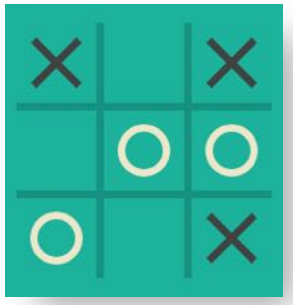


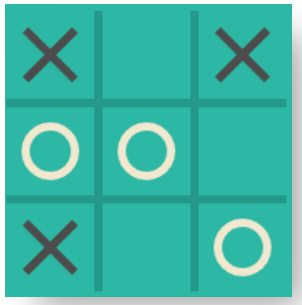


局面特征





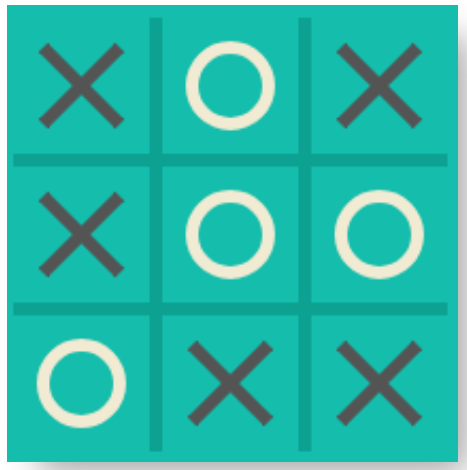
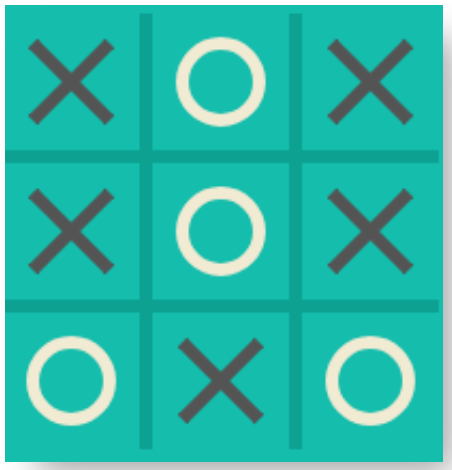
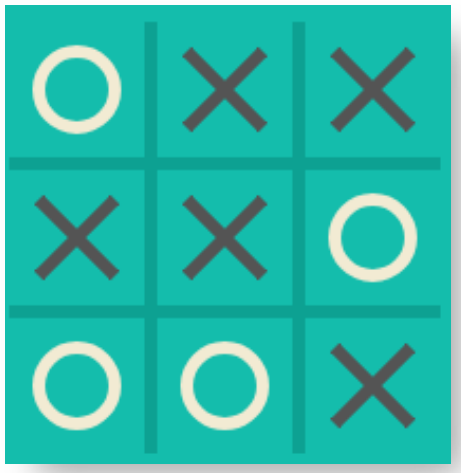




- 旋转对称性
- 轴对称性

局面简单

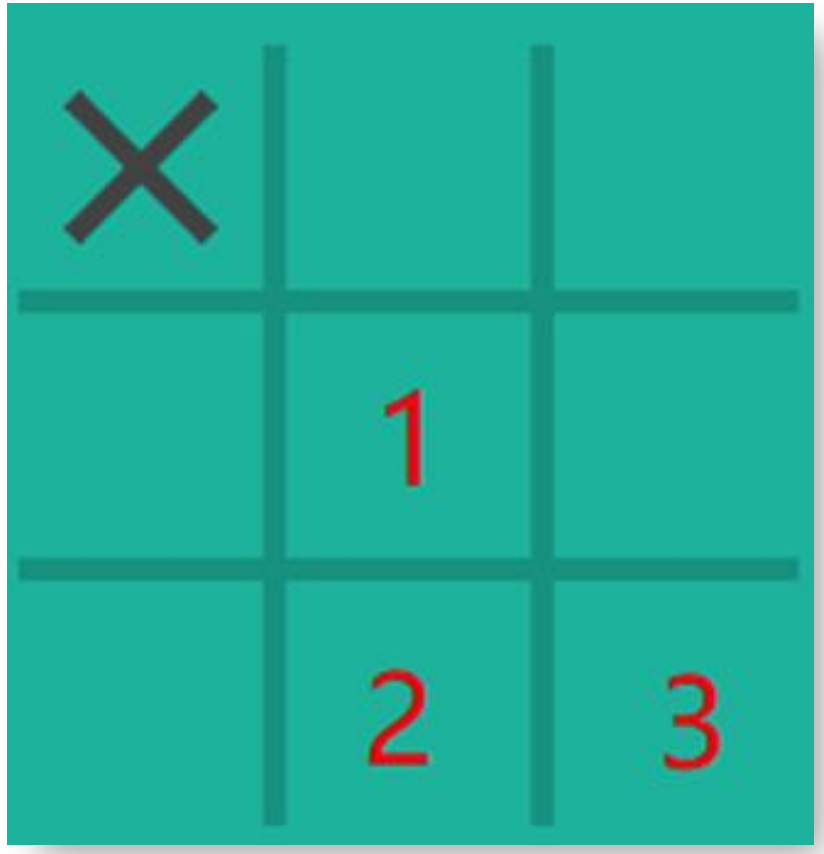
- 考虑对称性，最终局面只有138种
- 其中先手方（X）获胜的局面有91种，后手方获胜的局面有44种，
- 平局的局面有3种

游戏策略

- 让自己获得胜利
  - 当你有两粒连子的时候，把他们连成3个
- 阻止对方获得胜利
  - 如果对方有两粒连子，阻止它们构成3连
- 尽量创造出能够获胜的机会
  - 争取同时有两个路径能够完成3个棋子连在一起，使得对方无法一次性封堵。
- 阻止对方创造这样的机会

我们如何决定下在哪里



- 当先手玩家选择下在角上时，
- 后手方在上图中的1/2/3哪个位置应对最好呢？

我们如何决定下在哪里

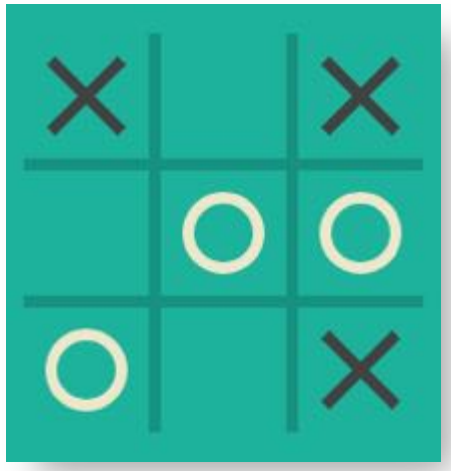
- 如果双方玩家都采取最优策略的话
  - 那么井字棋一定是一个平局
- 如果双方玩家的水平都较高的话
  - 那么先手方下在角落更有机会获胜
- 如果双方玩家水平都不高的话
  - 那么下在中心会更容易获胜。

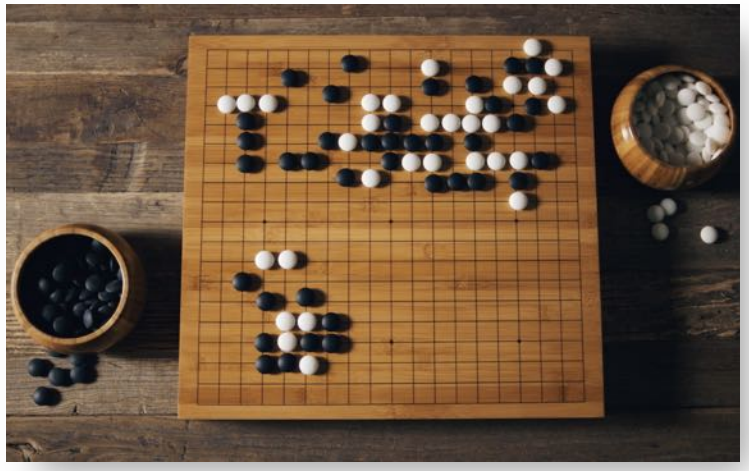
让AI决定下在哪里

- AI学会下棋，可没有直觉的说法
- 在特定的局面下，AI会按照一定的规则给出固定的决定

完全信息

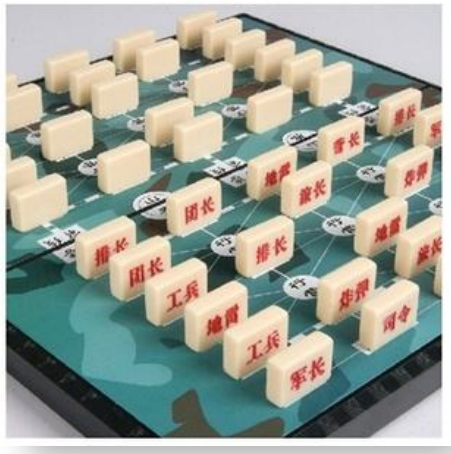
- 游戏的状态信息对所有玩家都是完全可见的。
  - 井字棋、黑白棋、象棋、围棋

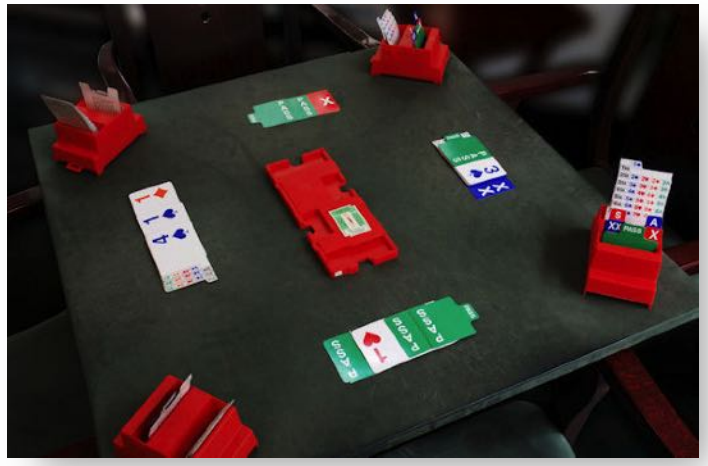




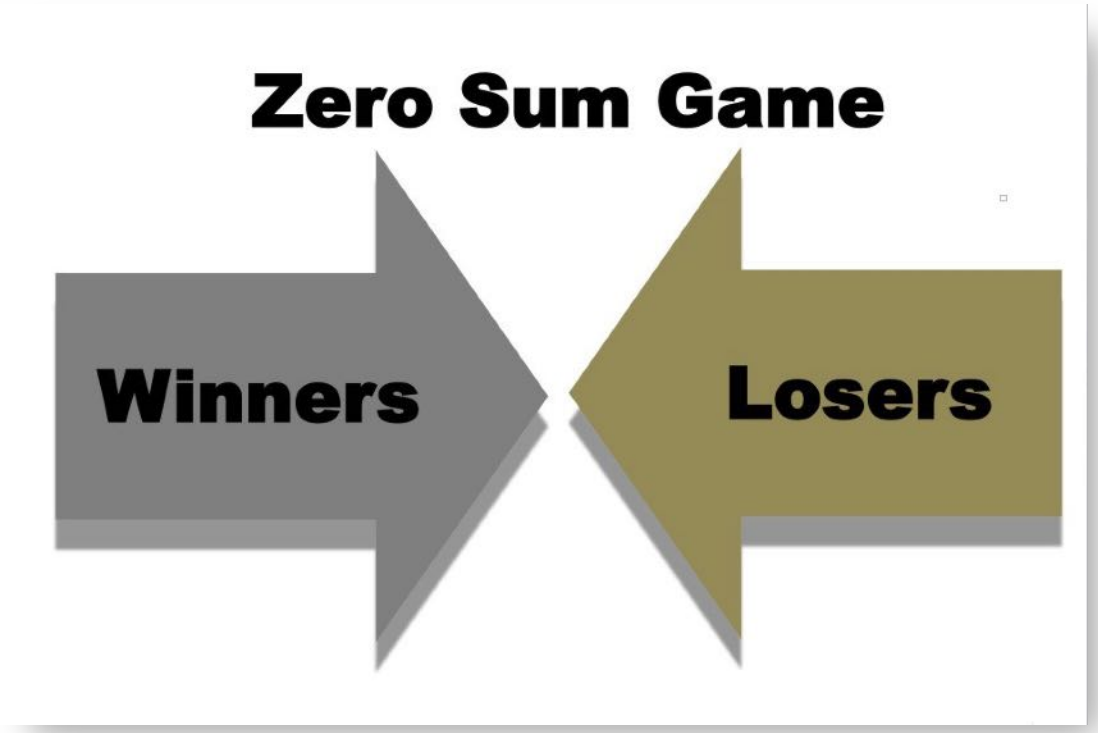
不完全信息

- 每个玩家有自己的私有信息，游戏的策略需要建立在对真实状态的猜测之上
  - 军棋、牌类游戏



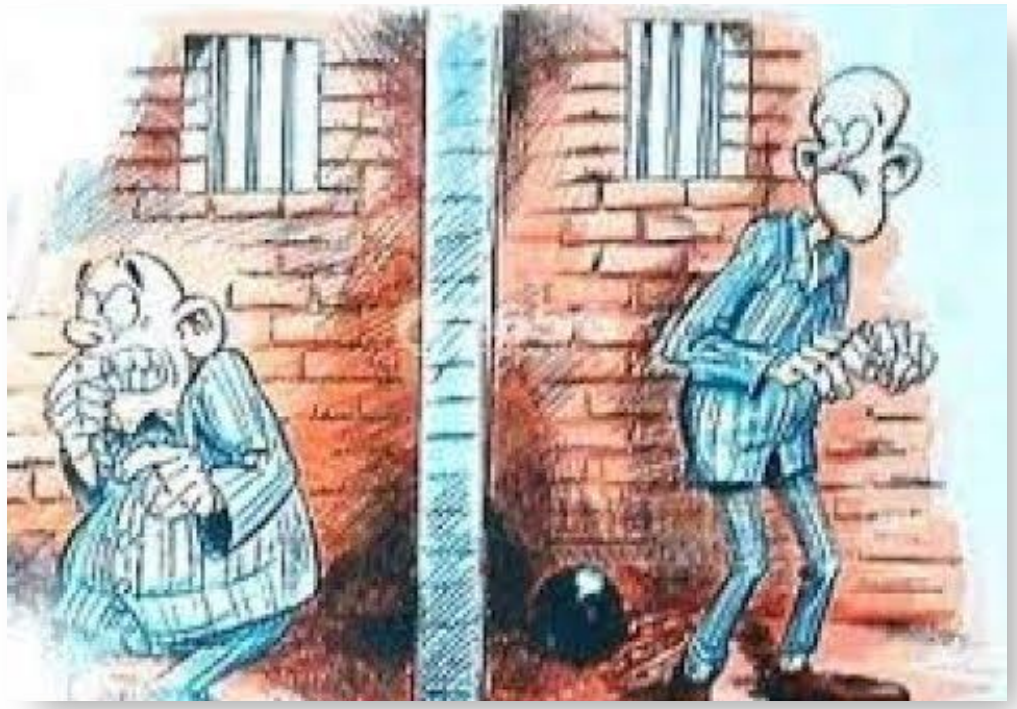


零和博弈



- 零和博弈中双方（或多方）的收益相加为0
- 只要让其他人的收益最小化，即可使自己的收益最大化

非零和博弈



- 非零和博弈中，所有人的收益之和不为0，存在“合作”或者“双赢”的可能。
- 自己的所得并不与他人的所失的大小相等，使他人收益最小化也可能“损人不利己”
  - 囚徒困境，麻将

非零和博弈：囚徒困境

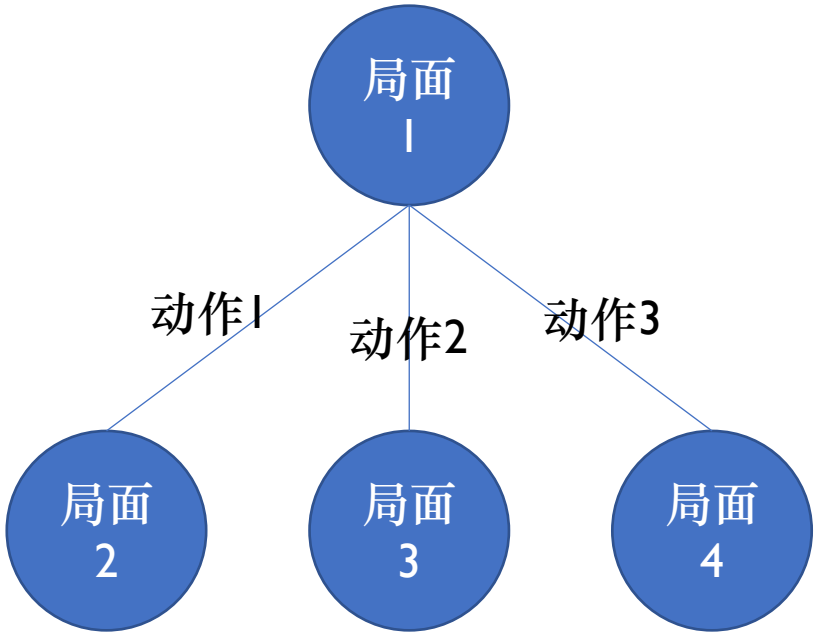
	乙沉默（合作）	乙认罪（背叛）
甲沉默（合作）	二人同服刑半年	甲服刑10年；乙即时获释
甲认罪（背叛）	甲即时获释；乙服刑10年	二人同服刑5年

非零和博弈

- 只考虑一个人最佳选择并非考虑团体的最佳选择。
- 选择使对方收益最小化的策略并不能使自己获得最大收益
- 在麻将中常有为了不让一个对手胡大牌，故意让另一个对手胡小牌的策略

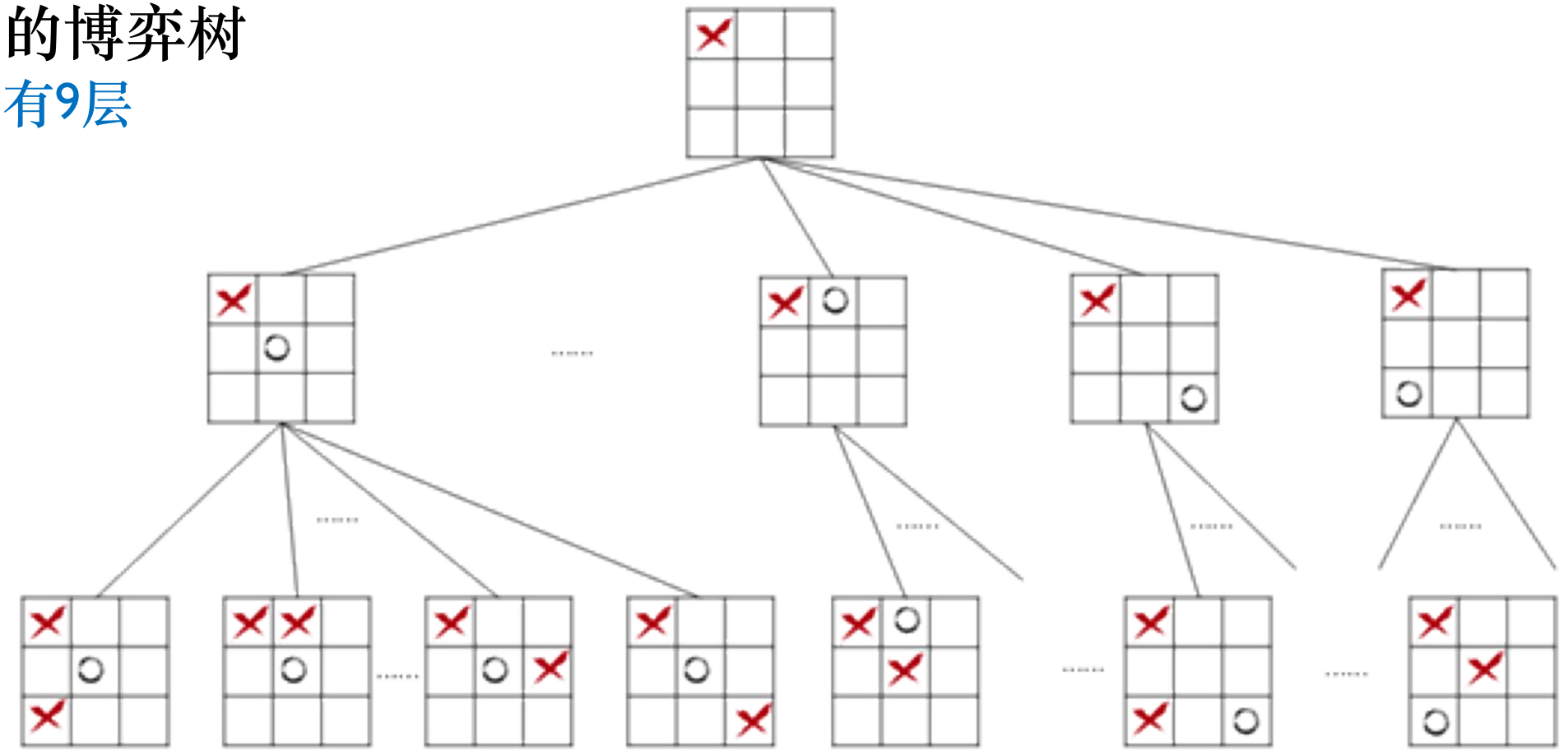
博弈树构建

- 博弈树的每个节点对应于每一个局面，每条边对应于一个动作
- 在完全信息零和博弈的条件下，能够构建简单的博弈树
- 如果在不完全信息、非零和博弈的情况下，博弈树较为复杂



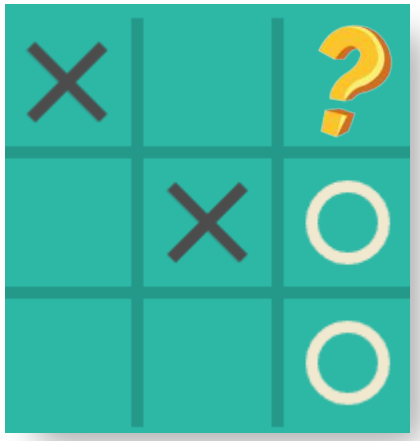
博弈树构建

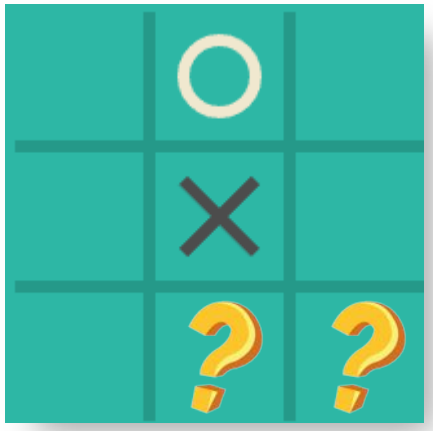
- 井字棋的博弈树
  - 最高有9层



如何判断是否有利

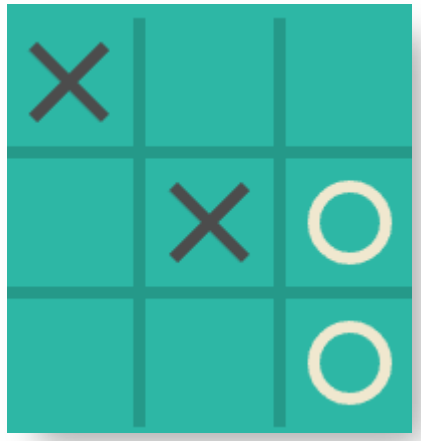
- 当前局面如何
- 接下来走哪一步棋是最有利的





估值函数

- 估值函数
  - 对每一种局面给出一个估值

$f($  $) = ?$

北京大学 陈斌 gischen@pku.edu.cn

估值函数

- 静态子力、数量

		中国象棋		国际象棋	
		价值	数量	价值	数量
	兵(卒)	1	5	2	8
守子	仕(士)	2	2	-	-
	相(象)	2	2	-	-
轻子	马	5	2	6	2
	炮	5	2	-	-
	象	-	-	6	2
重子	车	10	2	10	2
	后	-	-	18	1
总价值(双方)		106	32	156	32
棋盘大小		90		64	

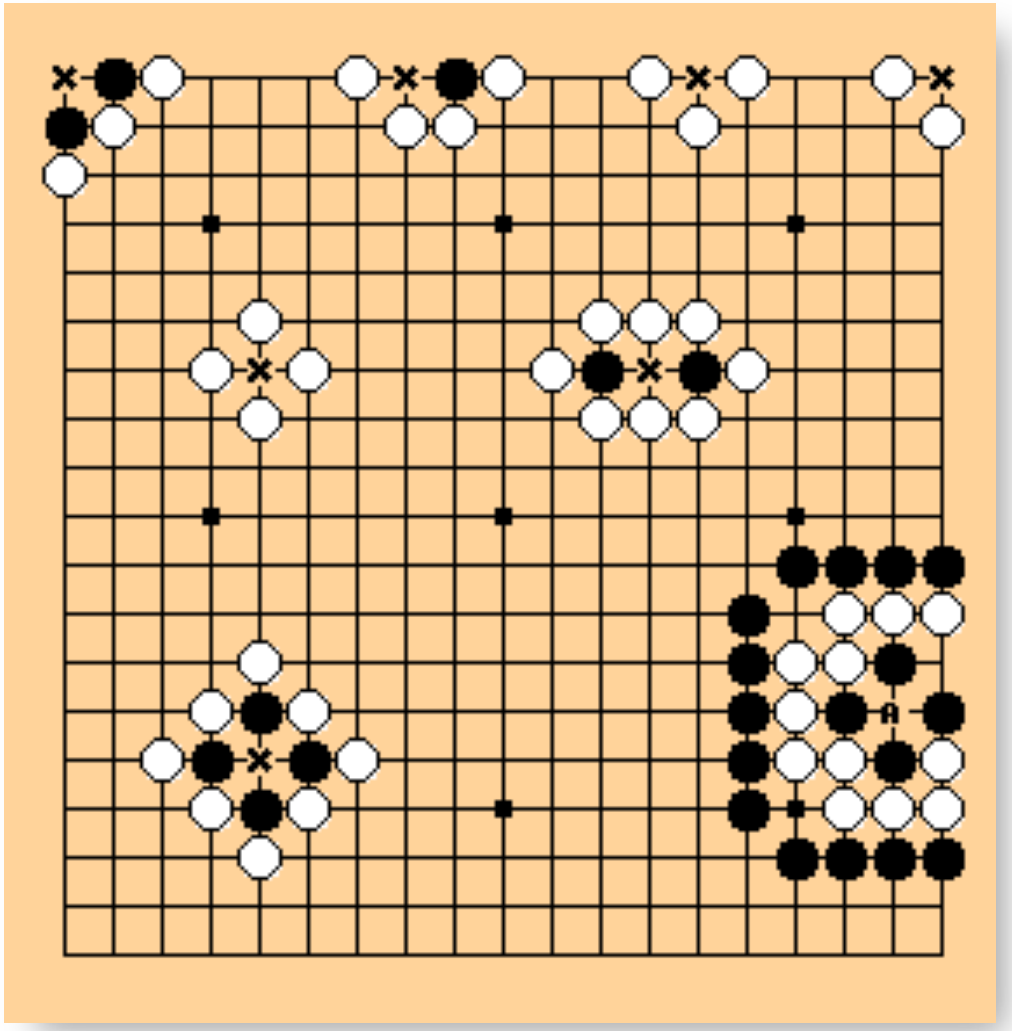
估值函数

- 棋盘特征
  - 不同位置、不同时间
  - 价值不同

棋子名称		活动范围			子力价值		
红	黑	中央	边线	角落	开局	中局	残局
帅	将	4	3	2	-	-	-
仕	士	4	-	1	1	2	2
相	象	4	2	-	2	2	3
马	马	8/6/4	4/3	2	4	5	5
车	车	17	17	17	10	10	10
炮	炮	17/13	17/14	17/15	5	5	6
兵	卒	1/3	1/2	1	2	1/3/5	3/2/1

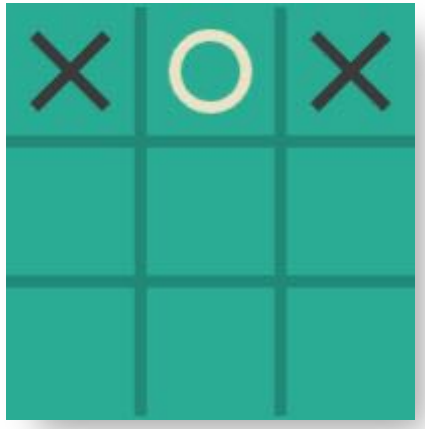
围棋的估值函数

- 难以通过子力数量和棋盘特征构建
- 不同局势下同一个子价值不同

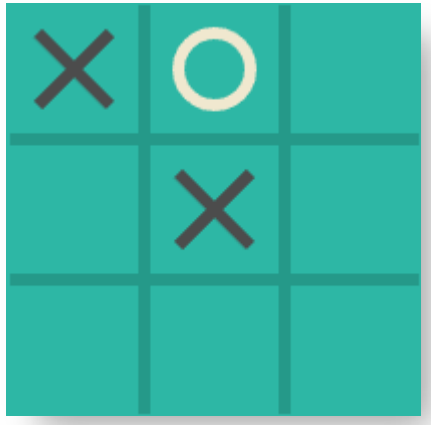


井字棋的估值函数

- 玩家X还存在可能性的行、列、斜线数减去
- 玩家O还存在可能性的行、列、斜线数



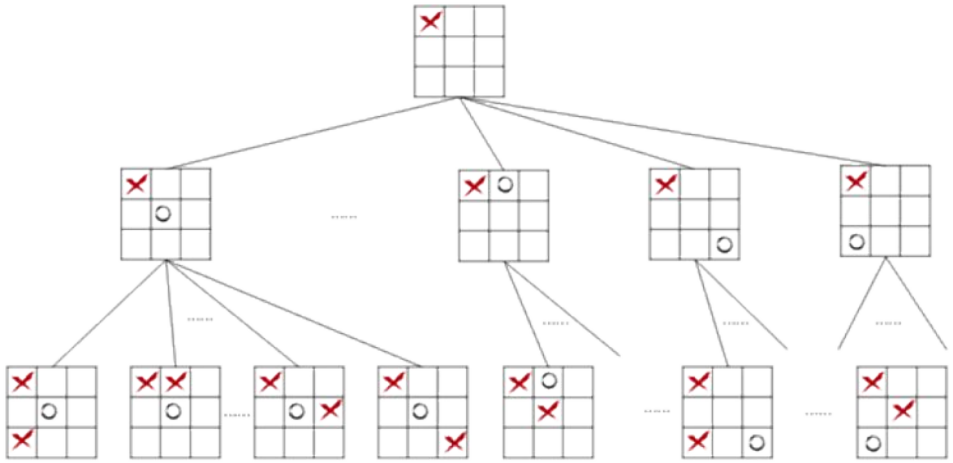
6-3=3

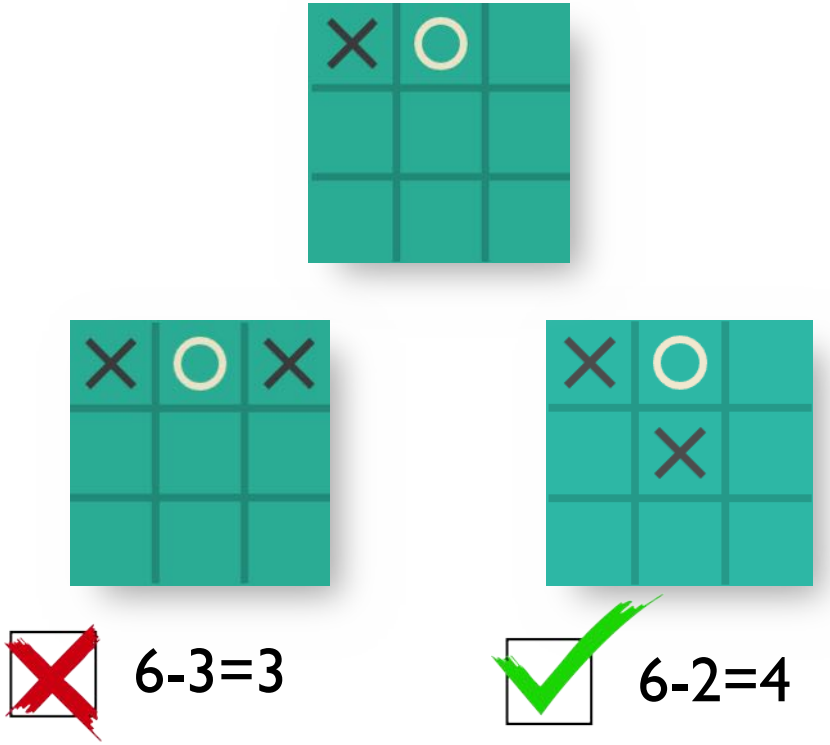


6-2=4

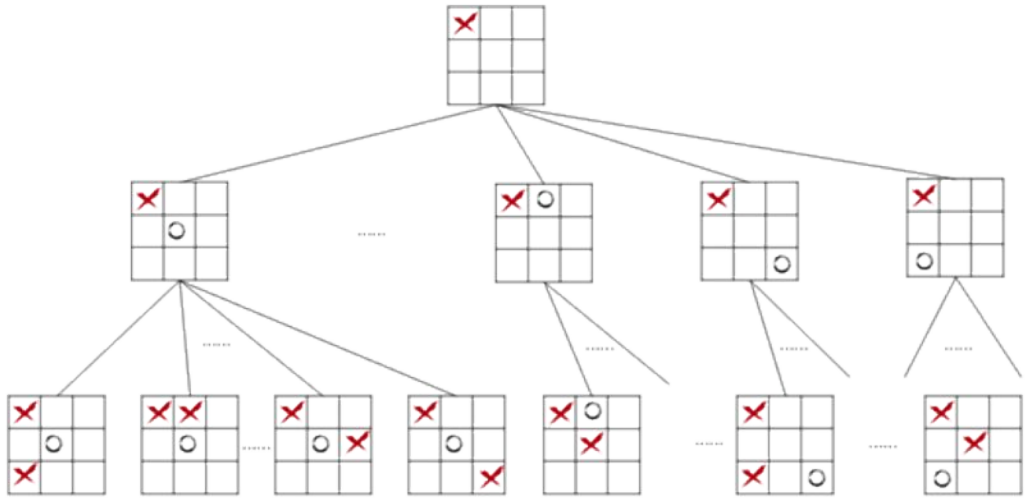
选取策略

- 根据估值函数选择最优策略
- 最佳行动就是能够使得下一个状态的评估值最大的行动



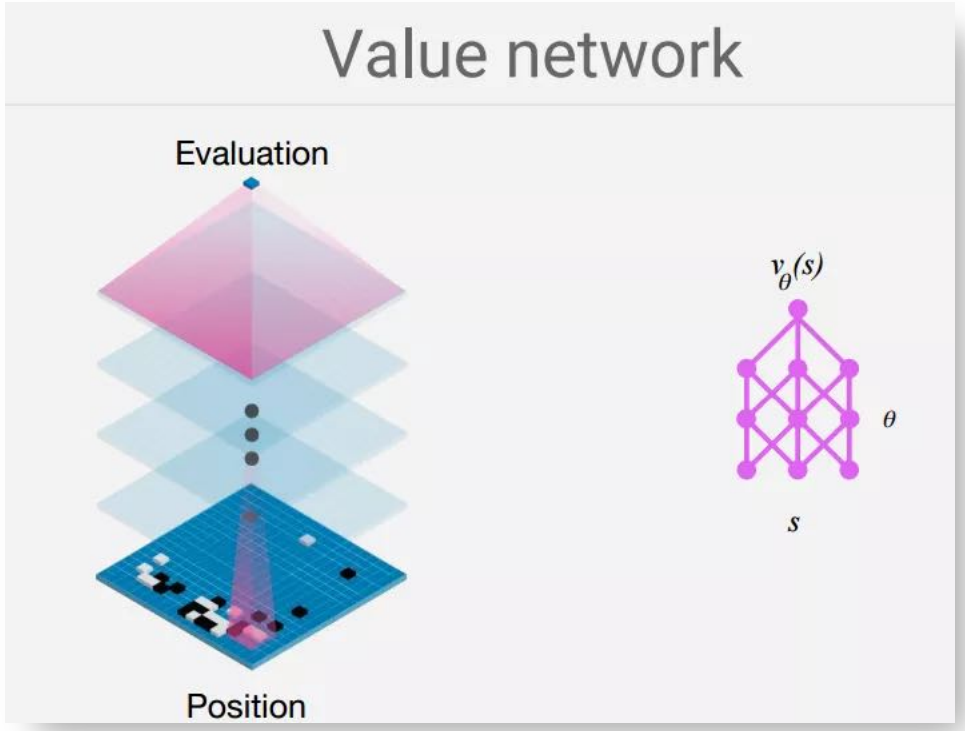


进一步改进



- 采用多步搜索策略
- 提高搜索的深度
- 尽量接近搜索过程的终止状态
- 搜索配合剪枝提高效率

进一步改进



- 改进估值函数
- 针对不同棋类添加不同的估值规则
- 通过神经网络等方式让AI自己学得策略

最大最小值法

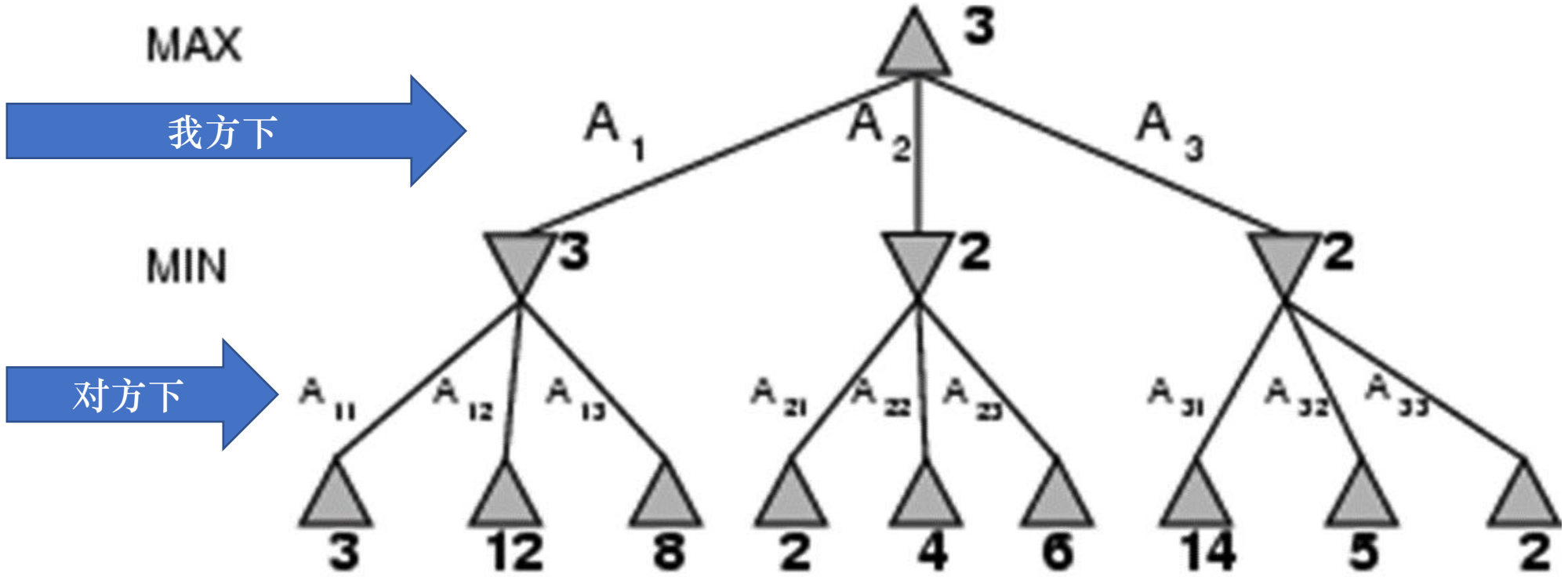
- 零和游戏中
- 玩家在可选的选项中选择将其优势最大化的选择
- 也就是说要选择令对手优势最小化的方法

最大最小值法

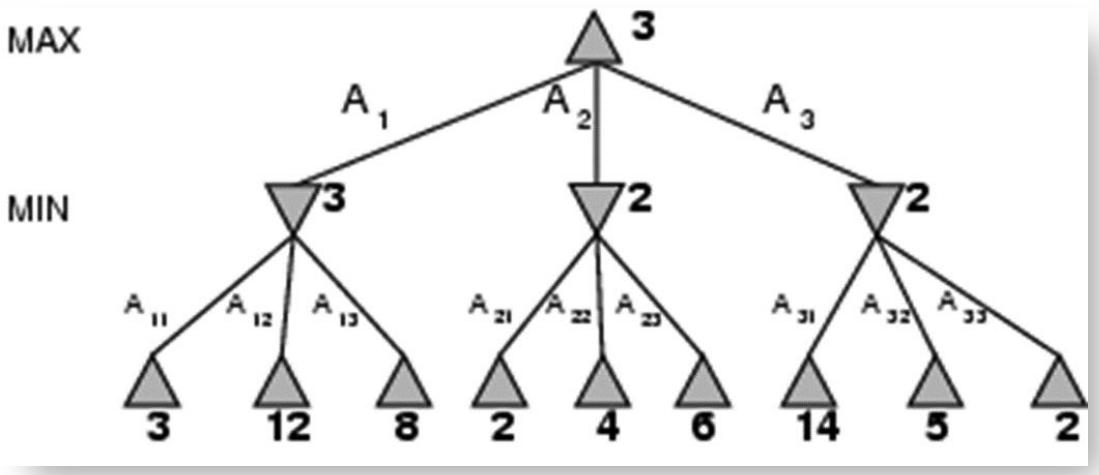
- 回合制的游戏
- 双方都很聪明，都会采用最优策略
- 用最大最小值法统一表示
  - 轮到我方下，选择我方的最大估值
  - 轮到对方下，选择我方的最小估值

minimax值

- 一个minimax决策树，包括max结点、min结点和终止结点

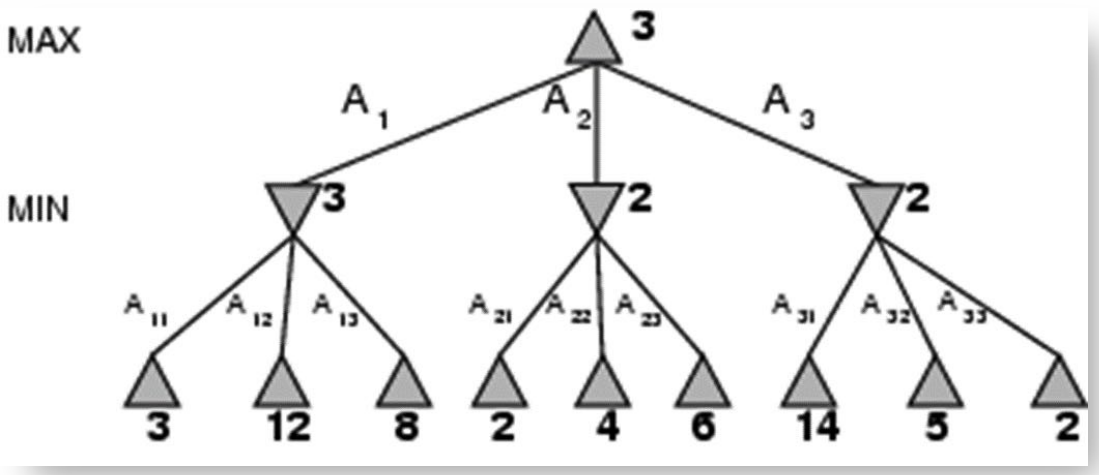


minimax值：表示决策树上结点的估值



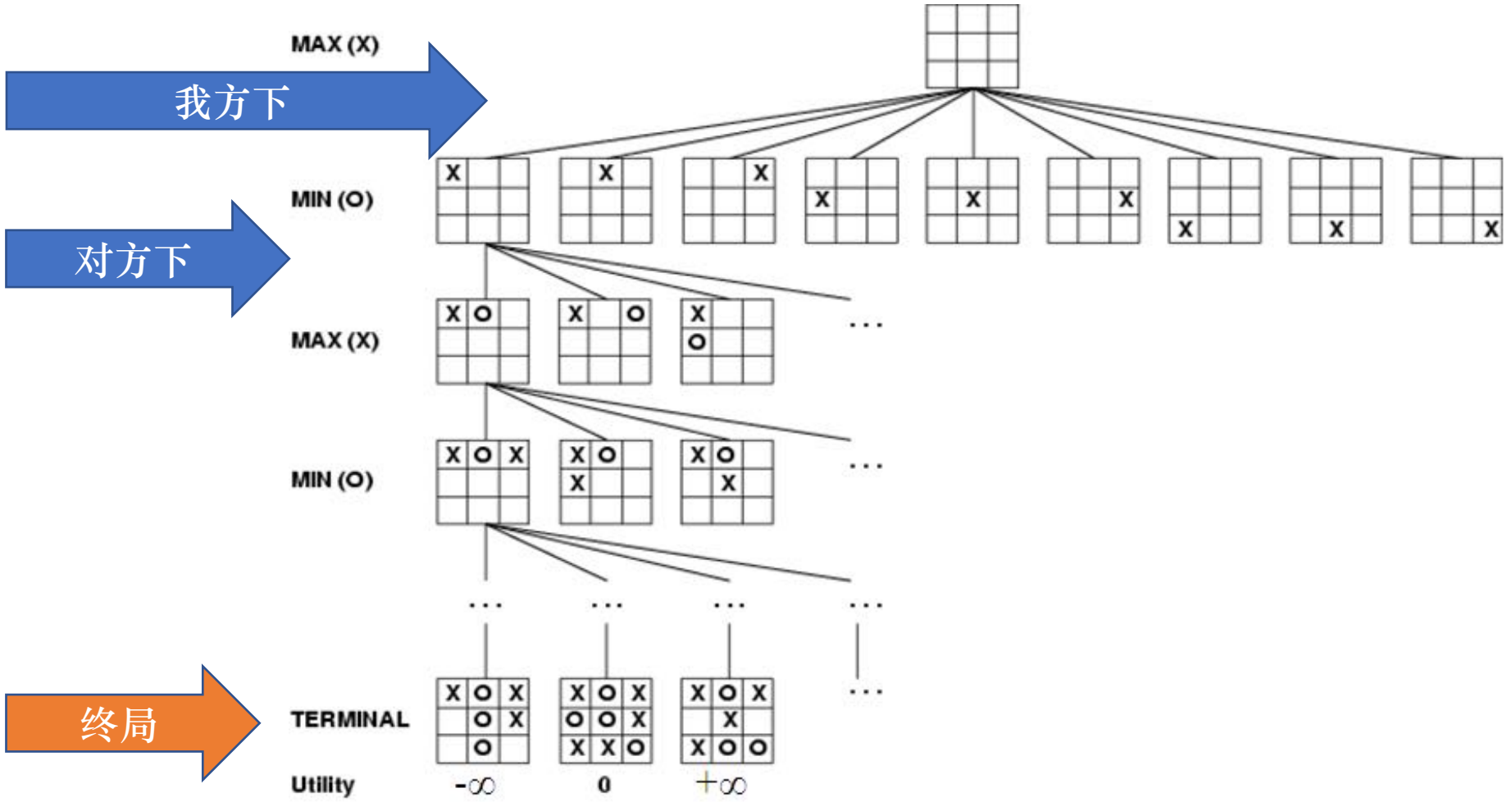
- 对于终止结点， minimax值等于直接对局面的估值
- 对于MAX结点，选择minimax值最大的子结点的值作为MAX结点的值
- 对于MIN结点，选择minimax值最小的子结点的值作为MIN结点的值

算法过程



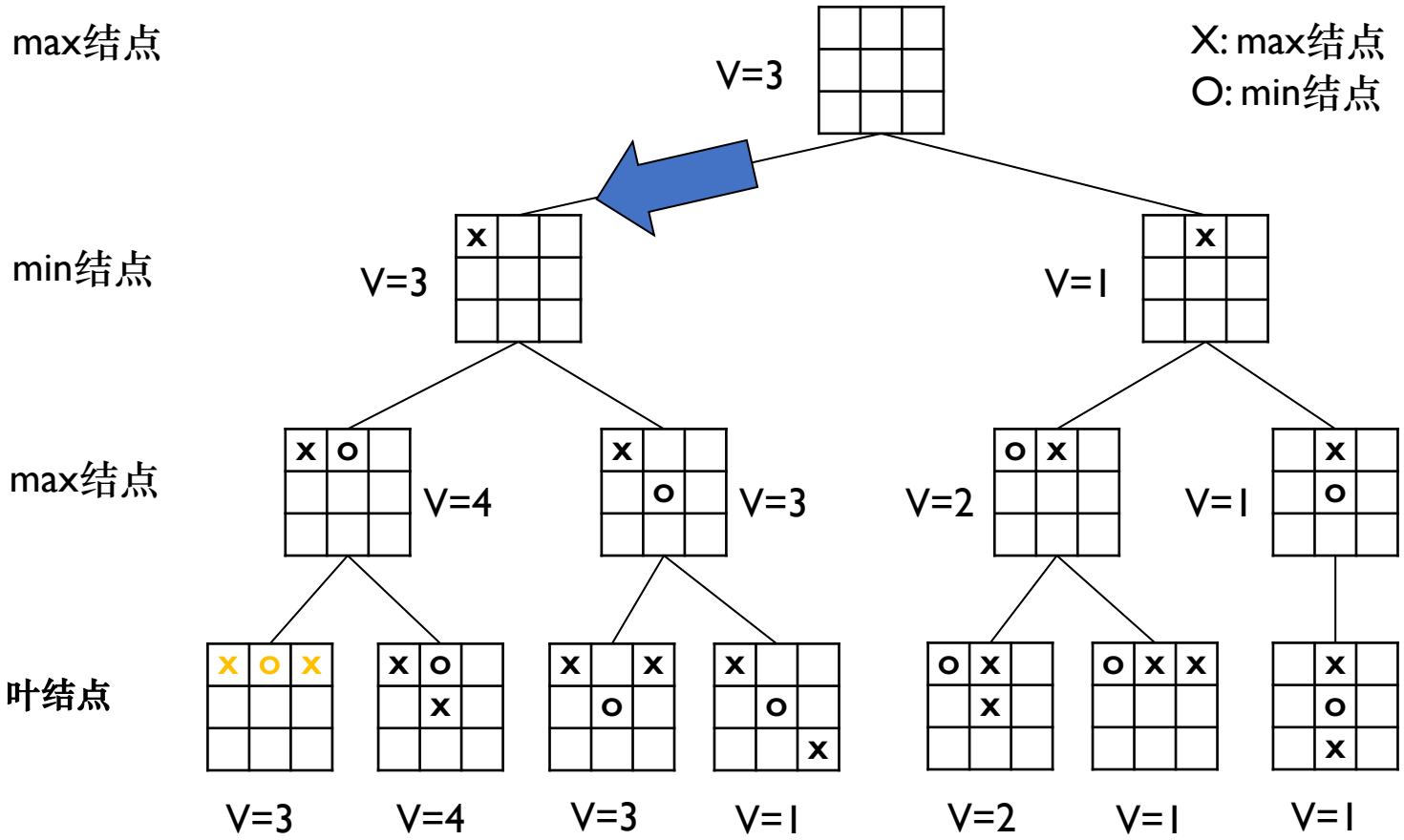
1. 构建决策树
2. 将评估函数应用于终局的叶子结点
3. 自底向上计算每个结点的minimax值
4. 从根结点选择minimax值最大的分支，作为行动策略

构建决策树，叶子结点估值



北京大学 陈斌 gischen@pku.edu.cn

算法过程演示

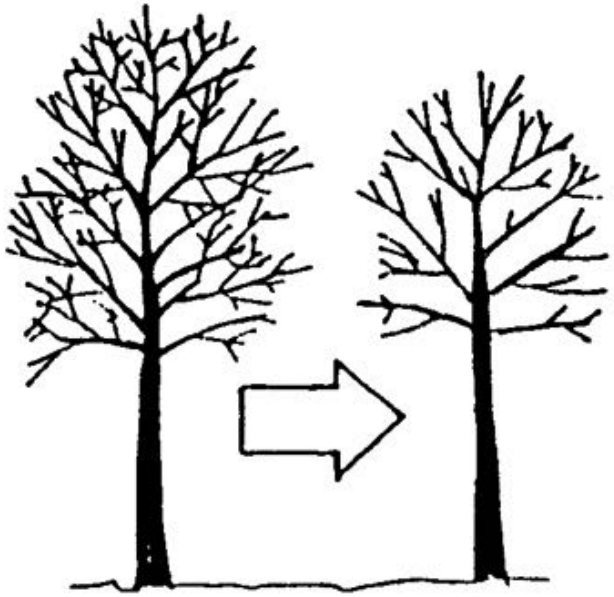

max结点
min结点
max结点
叶结点
V=3
V=3
V=1
V=4
V=3
V=2
V=1
V=3
V=4
V=3
V=1
V=2
V=1
V=1
X: max结点
O: min结点

计算minmax值主要流程总结

- 如果结点是终止结点：
  - 应用估值函数求值
- 如果结点是max结点：
  - 找到每个子结点的值
  - 其中最大的子结点值作为这个结点的值
- 如果结点是min结点：
  - 找到每个子结点的值
  - 其中最小的子结点值作为这个结点的值

Minimax的优化：剪枝

- Minimax需要展开整个决策树
- 对于局面复杂的问题，需要很大的空间
- 有部分结点跟最后的结果无关，无需展开局面和计算估值
- 不计算这些结点可节省算法搜索的时间
- 去掉这些节点的过程叫做“剪枝”

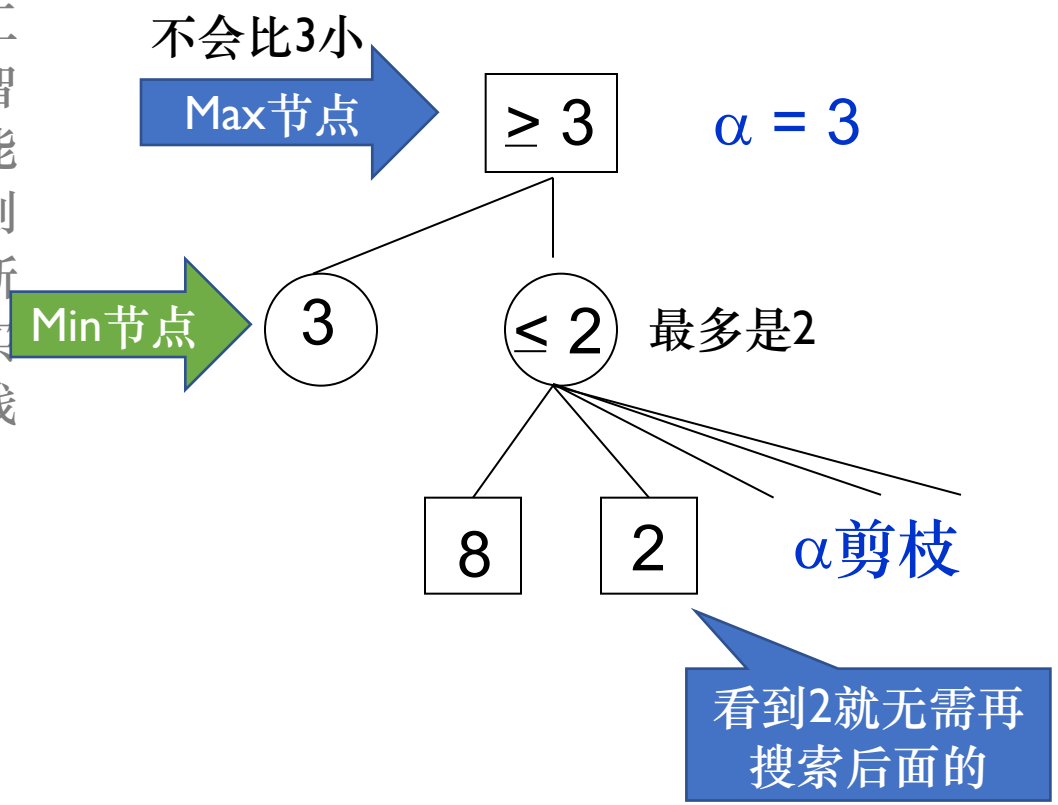


北京大学 陈斌 gischen@pku.edu.cn

Alpha-Beta剪枝

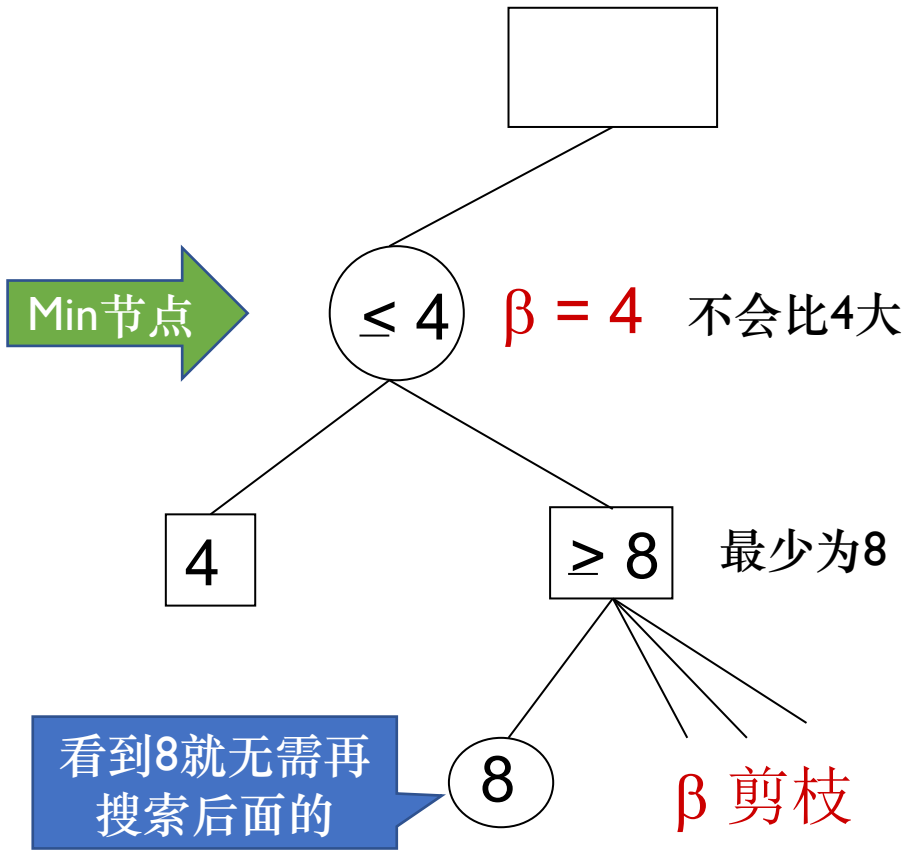
- 加速minimax搜索过程
- 每个结点存储局面估值之外，还存储可能取值的上下界
- 估值：minimax值
- 下界（最小可能值）：Alpha值
- 上界（最大可能值）：Beta值

Alpha剪枝



- Max结点的子节点搜索时
- 受到已经搜索过的值比其可能取值大的兄弟结点影响

Beta剪枝

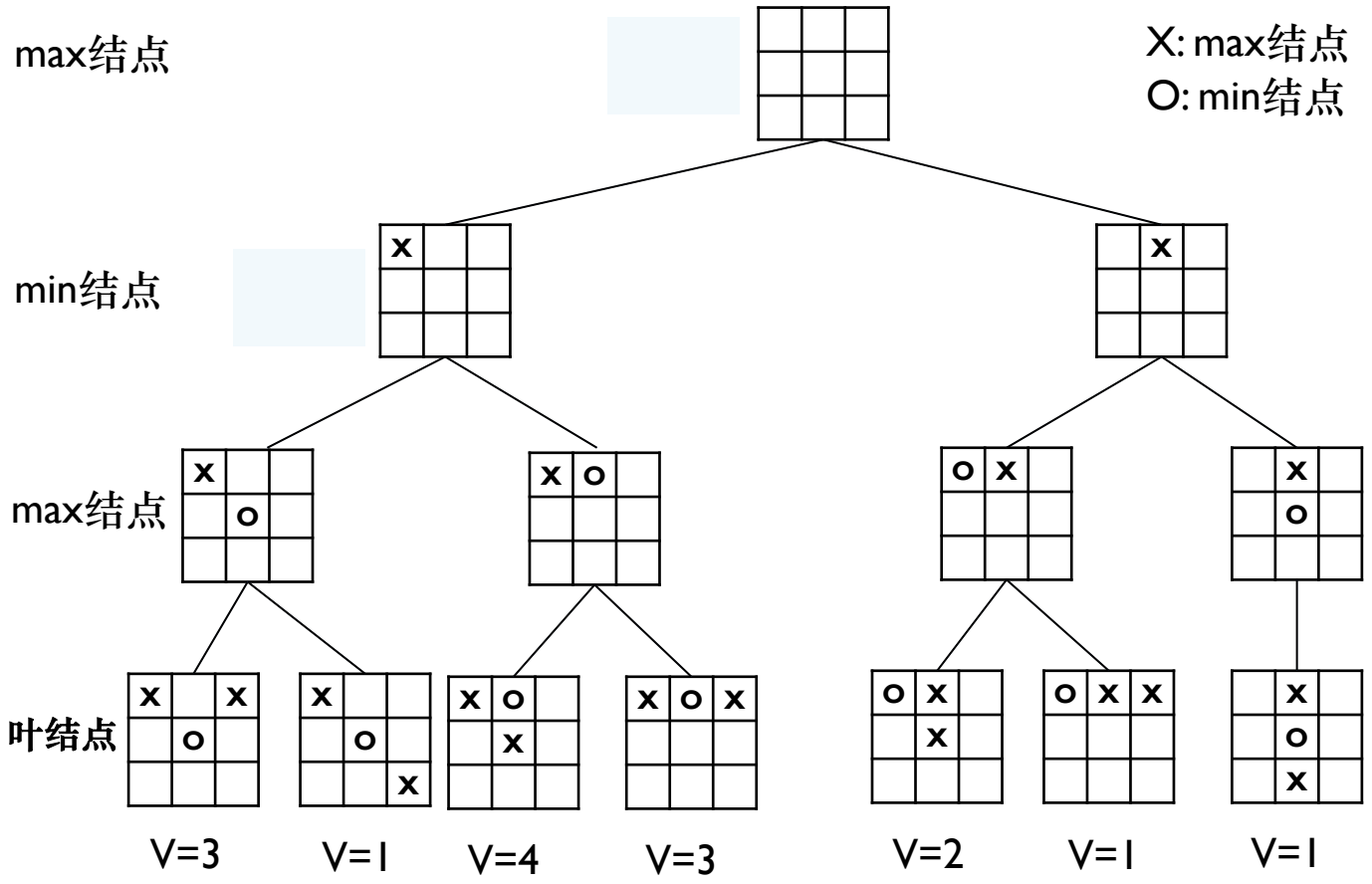


- Min结点的子节点搜索时
- 受到已经搜索过的值比其可能取值小的兄弟结点影响

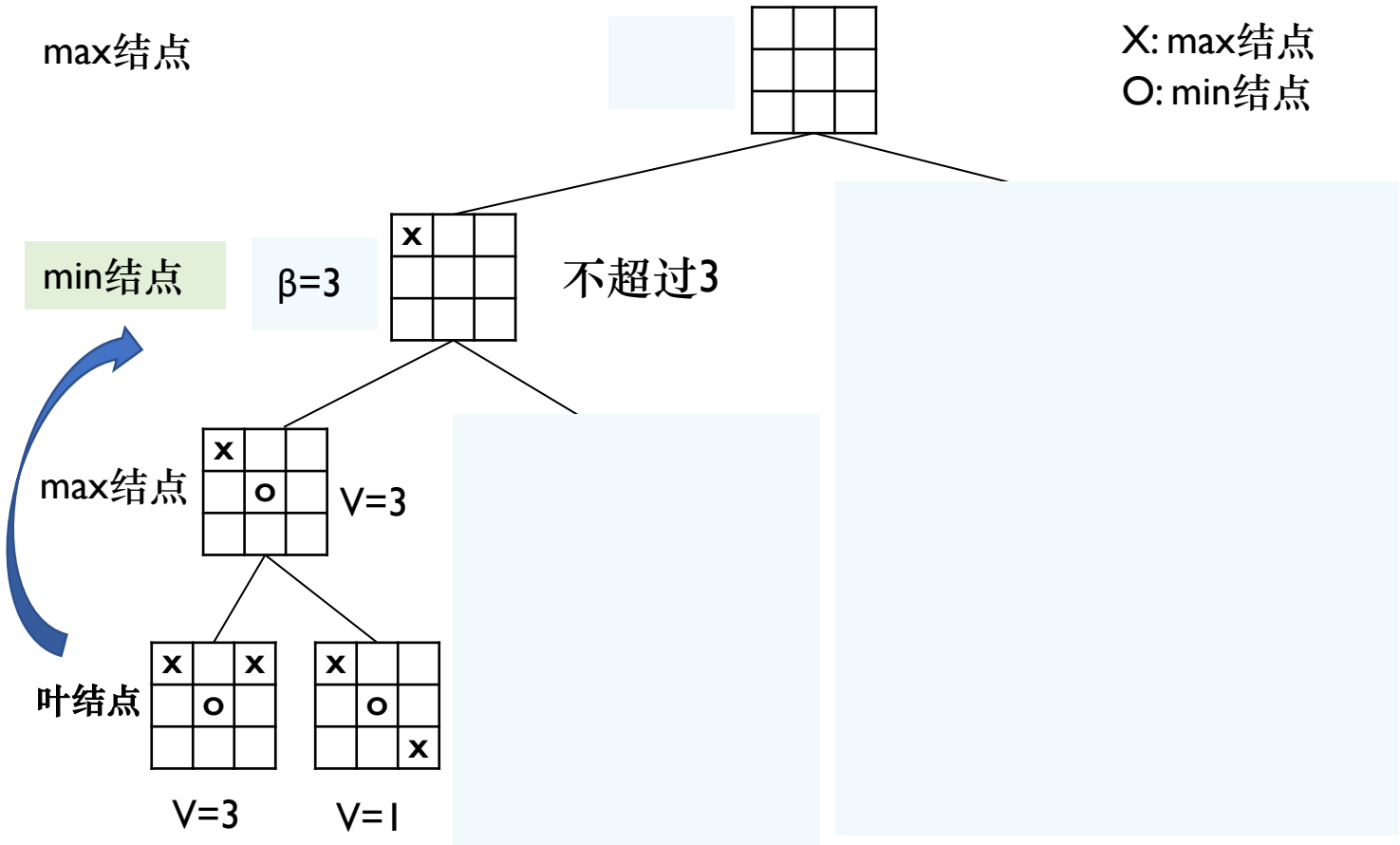
算法过程

1. 开始构建决策树
2. 将估值函数应用于叶子结点
3. 深度优先搜索，传递并更新α、β、结点值
   - Max结点更新α值(下限)
   - Min结点更新β值(上限)
4. 从根结点选择评估值最大的分支，作为行动策略

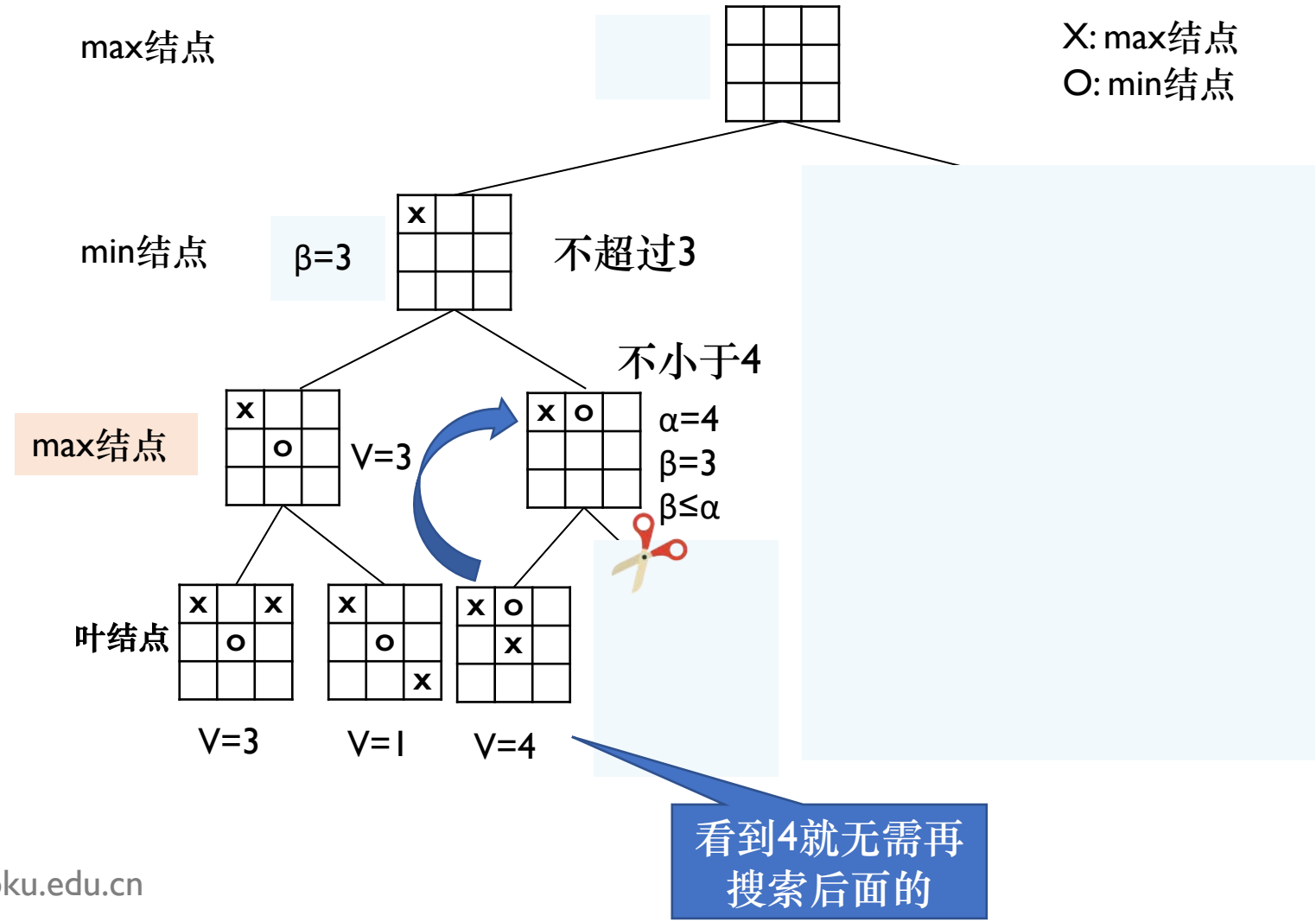
算法过程



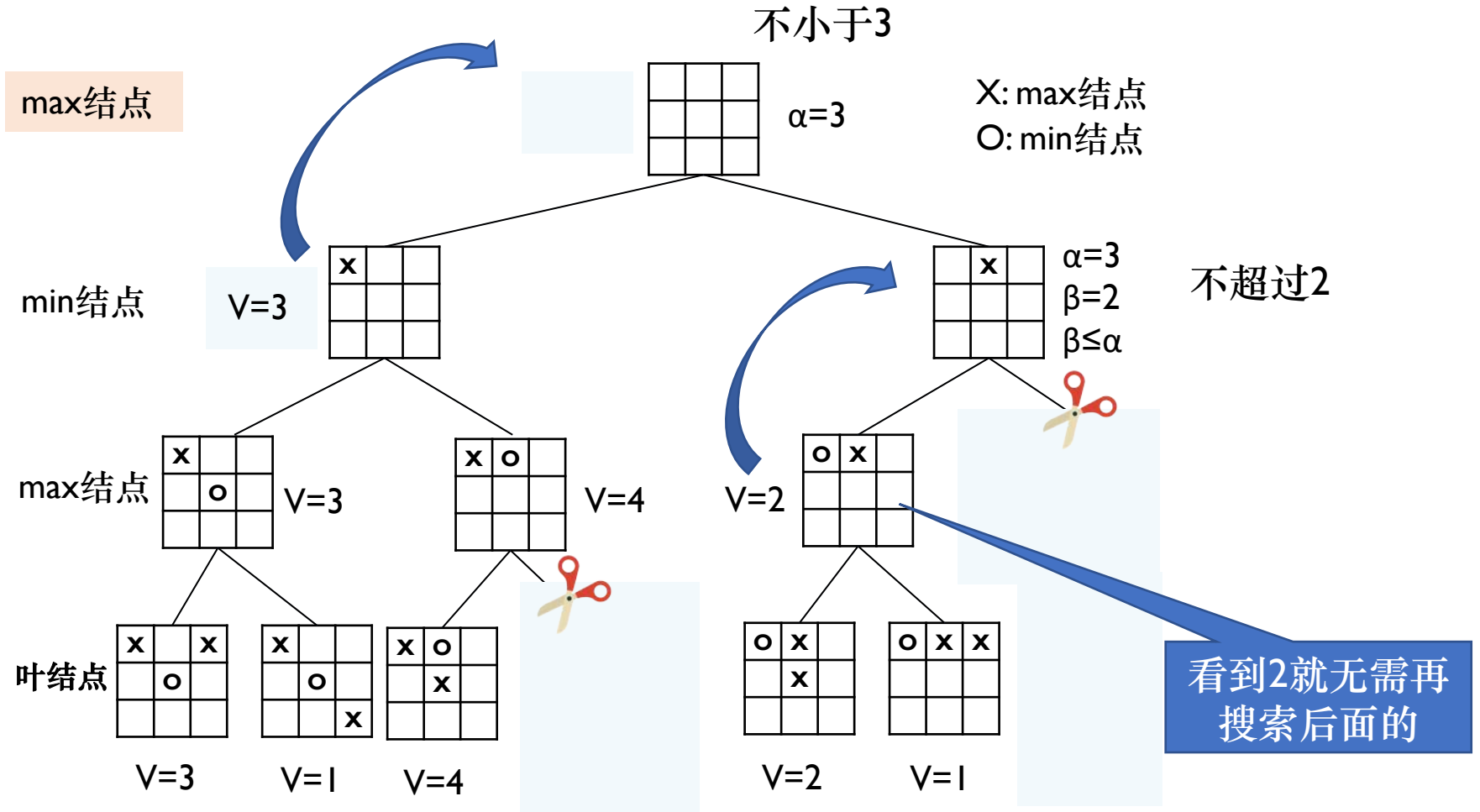
算法过程



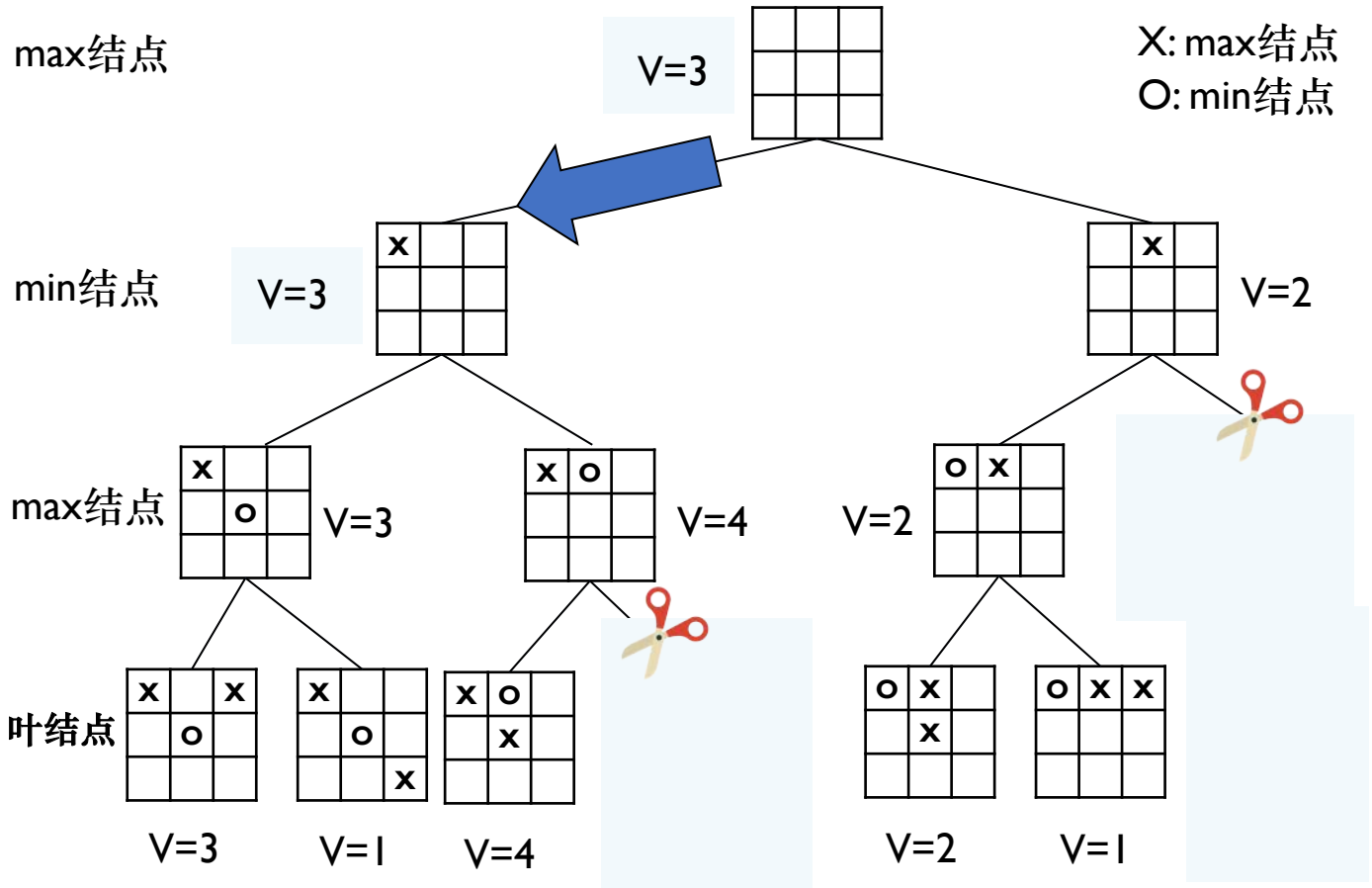
算法过程


max结点
X: max结点
O: min结点
min结点
β=3
不超过3
不小于4
max结点
V=3
α=4
β=3
β≤α
叶结点
V=3
V=1
V=4
看到4就无需再搜索后面的

算法过程


不小于3
max结点
α=3
X: max结点
O: min结点
min结点
V=3
α=3
β=2
β≤α
不超过2
max结点
V=3
V=4
V=2
叶结点
V=3
V=1
V=4
V=2
V=1
看到2就无需再搜索后面的

算法过程


max结点
V=3
X: max结点
O: min结点
min结点
V=3
V=2
max结点
V=3
V=4
V=2
叶结点
V=3
V=1
V=4
V=2
V=1

启发式算法

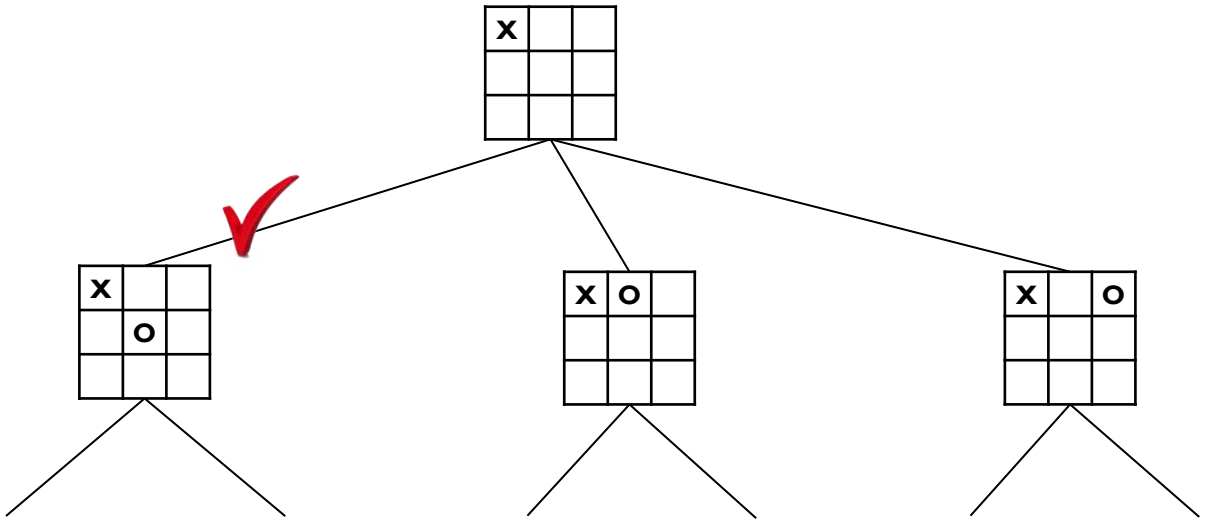
- 在搜索过程中，启发式算法被定义成一系列额外的规则
- 经验法则
- 利用一些特定的知识
- “高手怎么下，我也怎么下”

特点

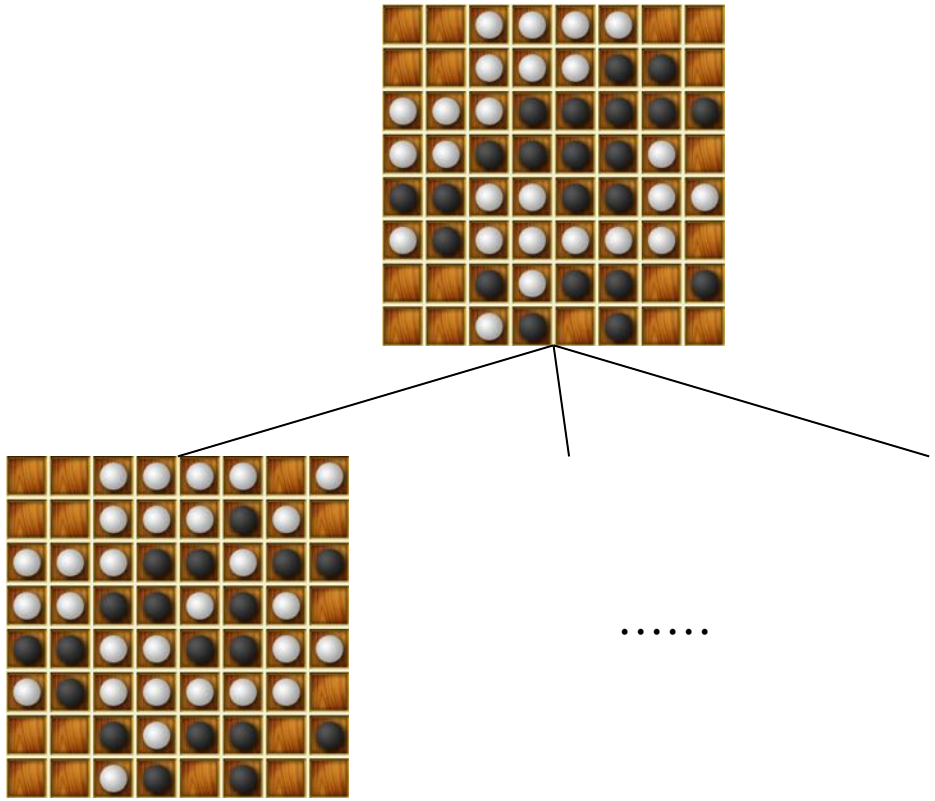
- 它常能发现很不错的解，但也没办法证明它不会得到较坏的解
- 它通常可在合理时间解出答案，但也没办法知道它是否每次都可以这样的速度求解

井字棋的启发式规则

- 当X下在角落的时候，O必须下在中心



黑白棋

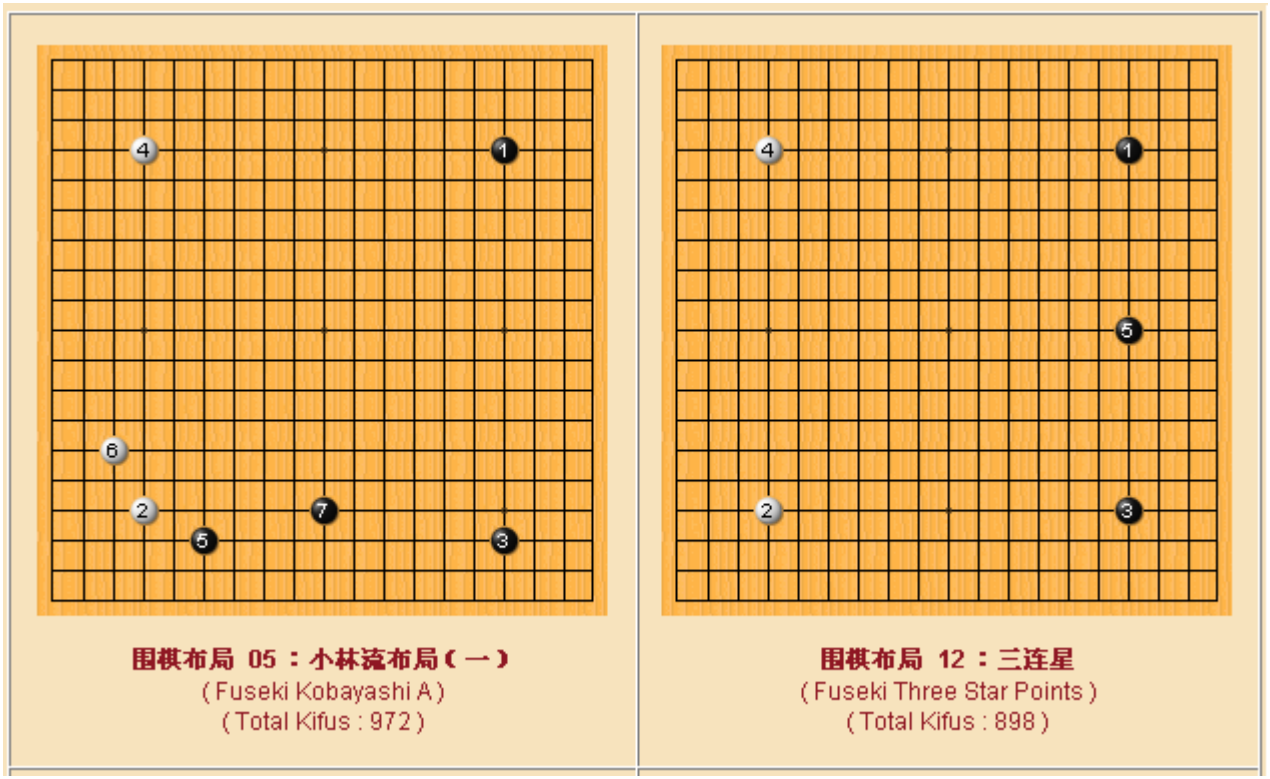


……

- 当能够下在角落的时候，选择下在角落

围棋

- 根据经验积累出很多布局定式



额外的估值函数

$$f(n) = g(n) + h(n)$$

启发性规则估值

终止局面估值

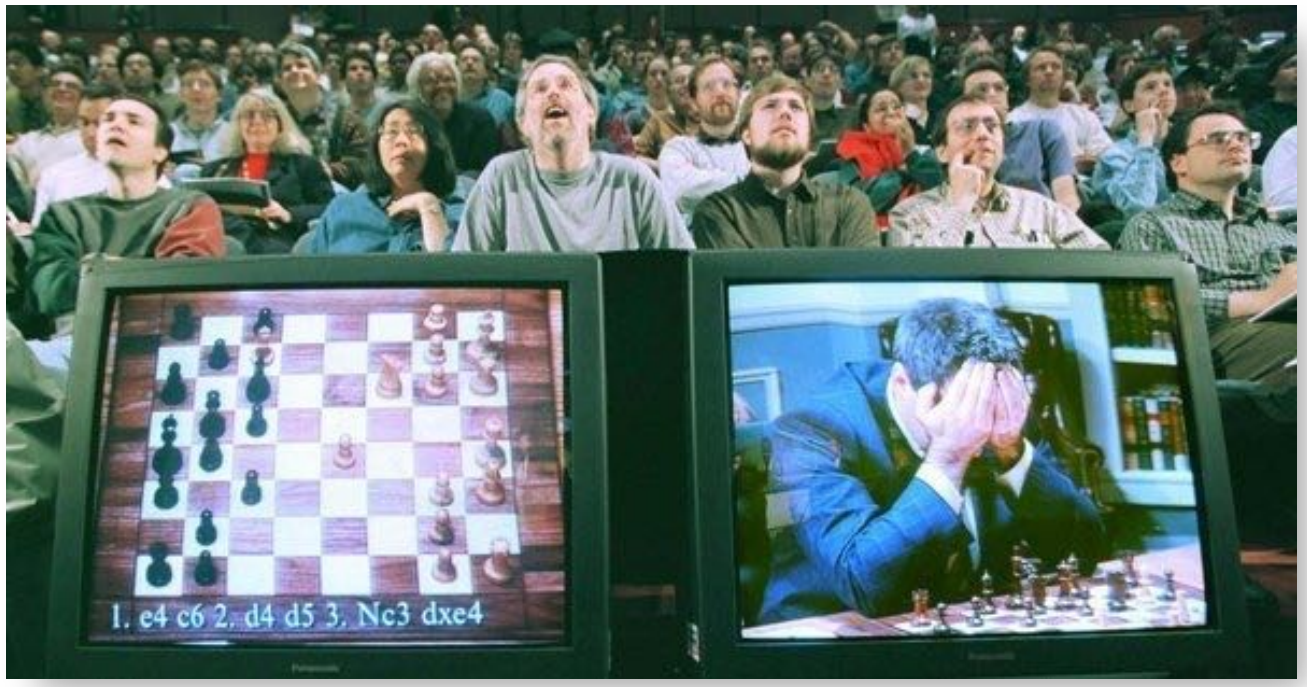
- 启发性规则反映在额外的估值函数中
- 可以和终止局面的估值函数一起计算

启发式算法总结

- 与Alpha-Beta剪枝不同，不用从叶节点自底向上计算估值
- 重点在于如何设计并实现启发函数，使我们能够更快地获得较优解

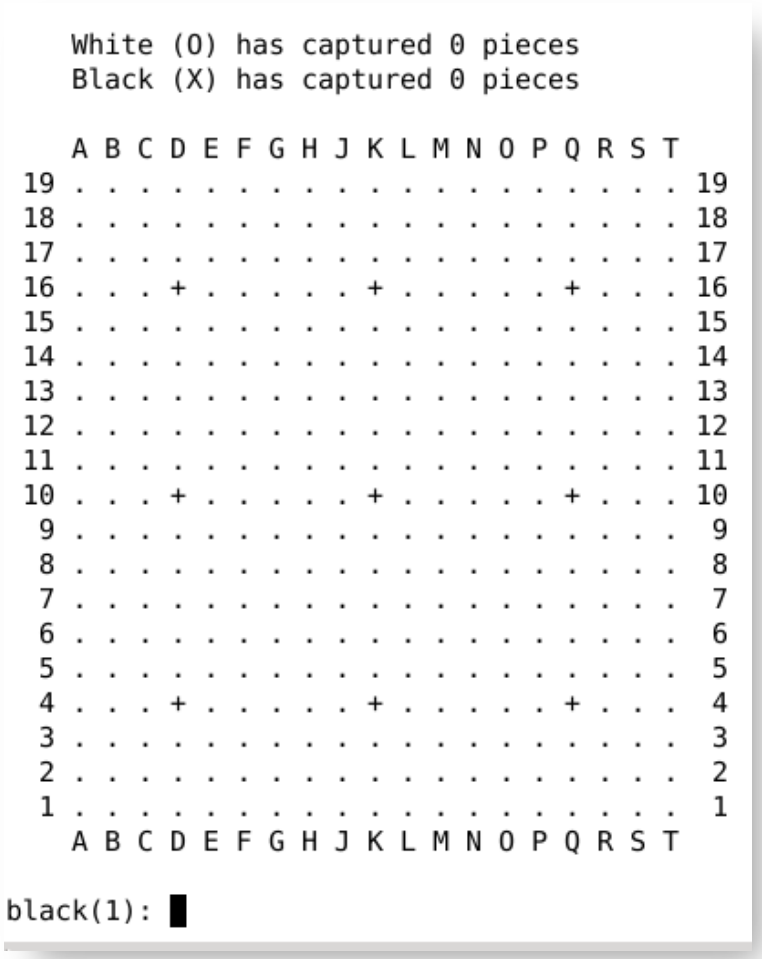
从象棋到围棋：深蓝的算法

- Alpha-Beta剪枝
- 残局库（增加搜索深度）
- 人类对局开局库



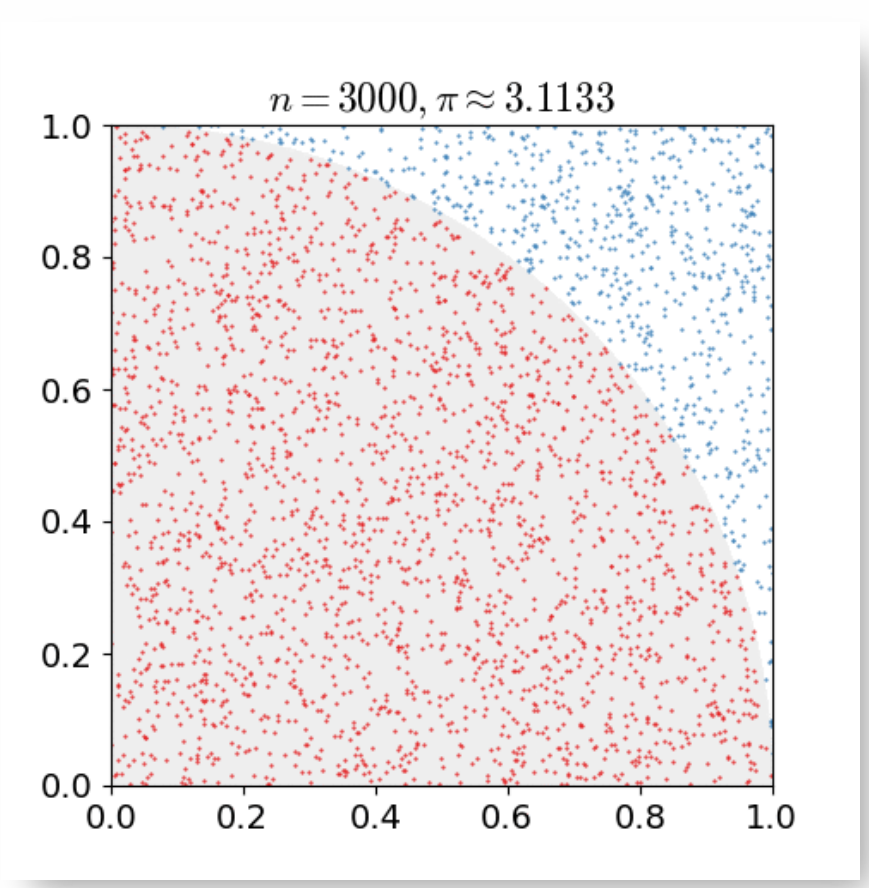
北京大学 陈斌 gischen@pku.edu.cn

Alpha-Beta剪枝之于围棋



- GNU Go
- 业余5~10级左右
- CGOS上，GNU Go被当作基准分数

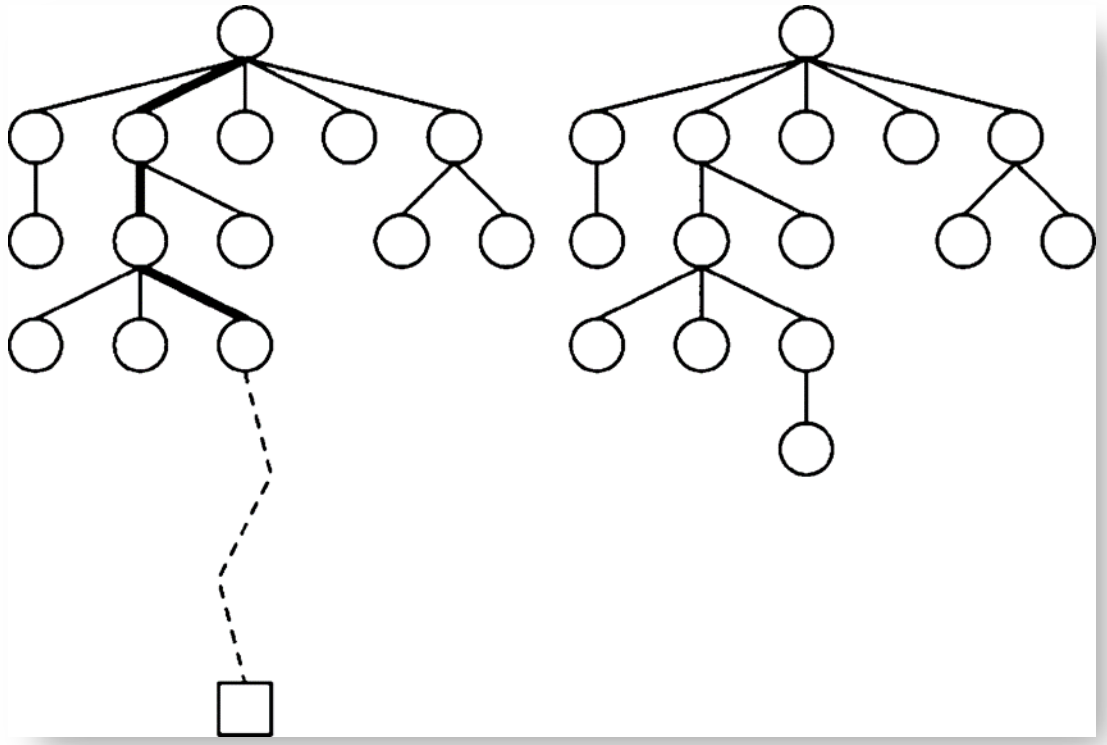
蒙特卡洛方法



- 通过随机采样计算得到近似结果
- 一种通用的计算方法

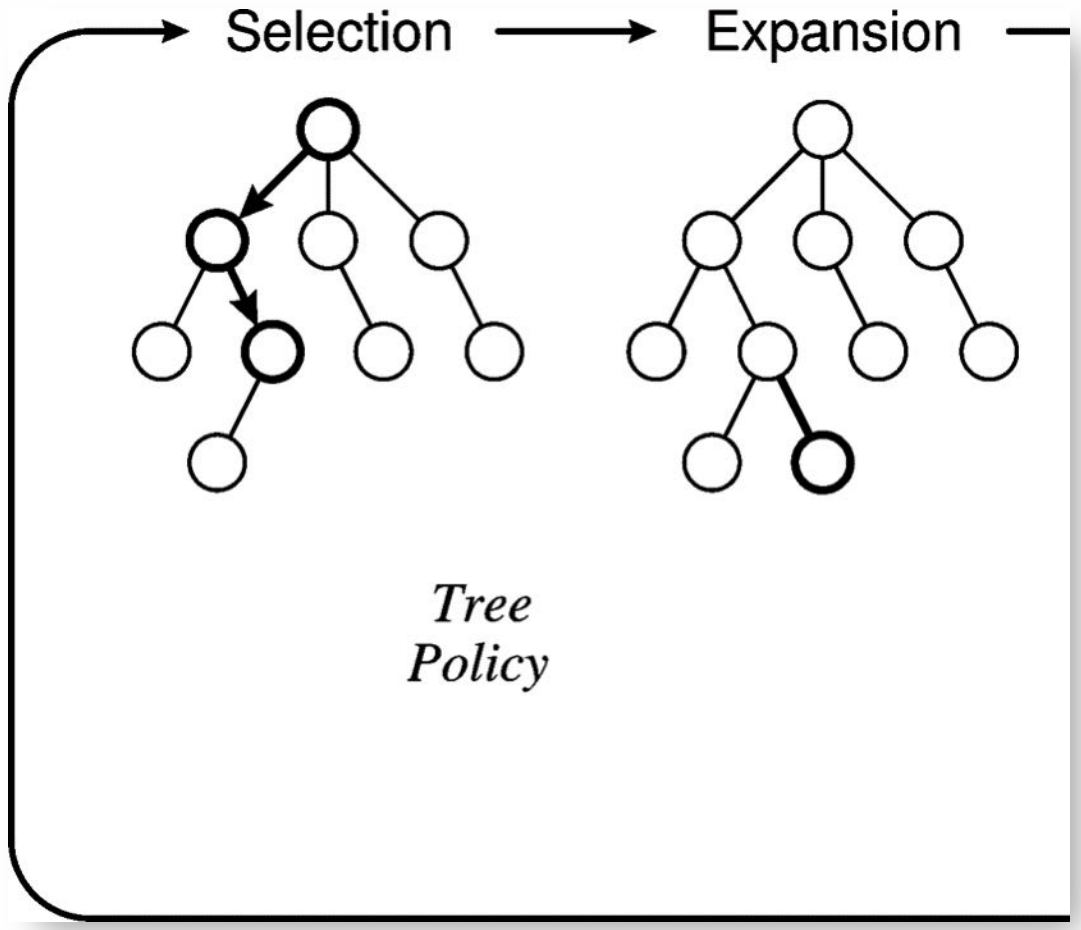
北京大学 陈斌 gischen@pku.edu.cn

蒙特卡洛树搜索（MCTS）



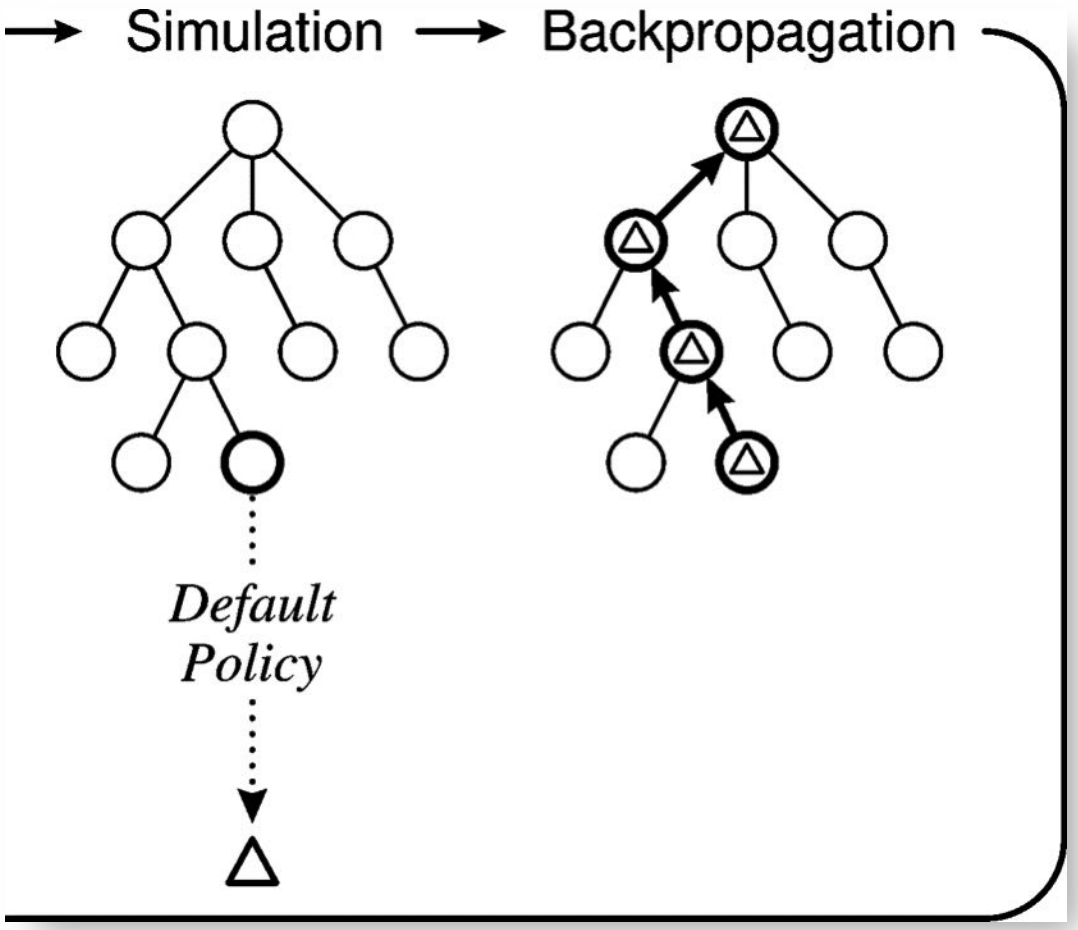
- 一种通过在决策空间中随机采样并根据结果构建决策树来寻找最优策略的方法
- 决策树的构建
  - 选择、扩张、模拟、反馈

选择、扩张



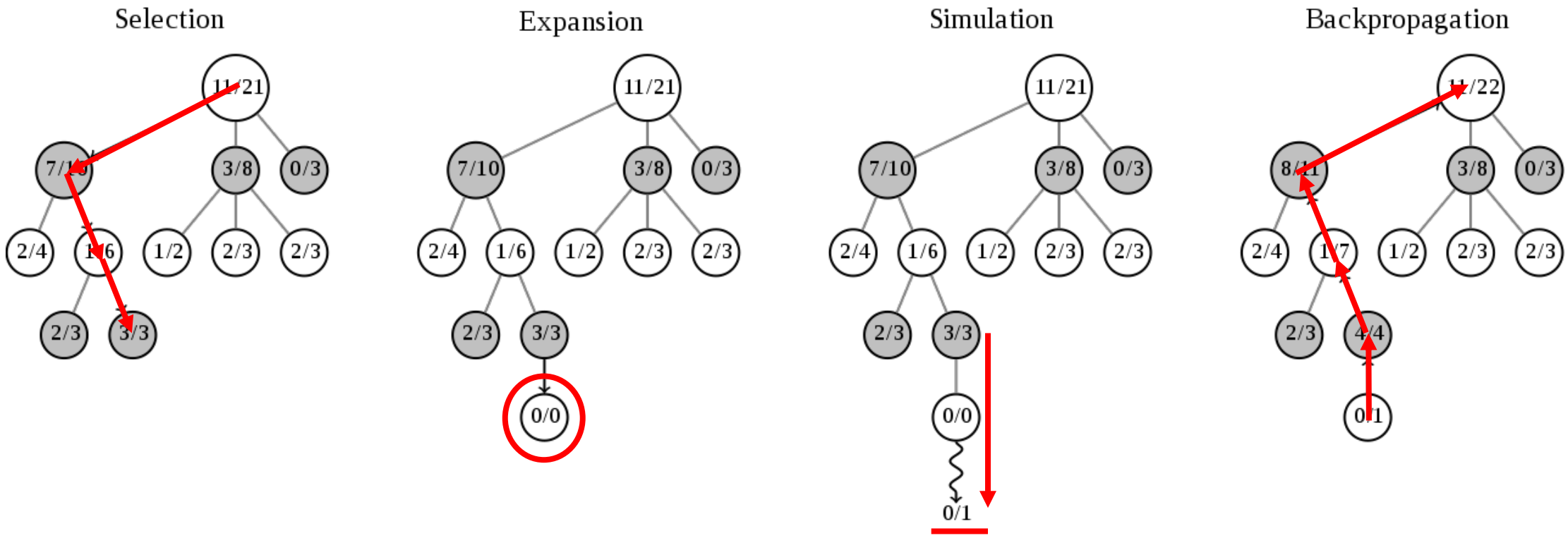
- 树策略：从决策树中选择并创建新的叶节点

模拟、反馈



- 默认策略：从一个非中止状态不断进行游戏并得到一个价值评估

MCTS例子

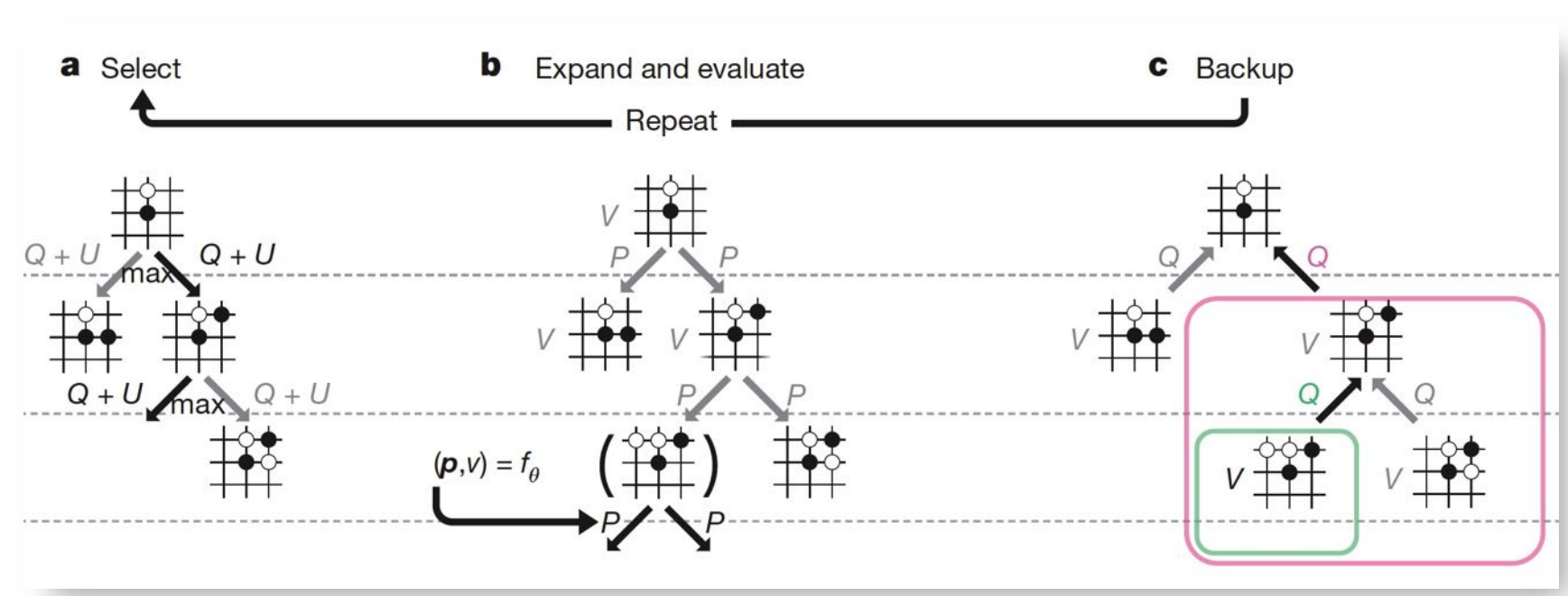

Selection
11/21
7/10
3/8
0/3
2/4
1/6
1/2
2/3
2/3
2/3
3/3
Expansion
11/21
7/10
3/8
0/3
2/4
1/6
1/2
2/3
2/3
2/3
3/3
0/0
Simulation
11/21
7/10
3/8
0/3
2/4
1/6
1/2
2/3
2/3
2/3
3/3
0/0
0/1
Backpropagation
11/22
8/11
3/8
0/3
2/4
1/7
1/2
2/3
2/3
2/3
4/4
0/1

MCTS之于围棋

- MoGo 第一个使用使用蒙特卡洛树搜索的围棋程序（2006年），在9×9的棋盘上击败了职业选手
- DeepZenGo是AlphaGo之前最强的围棋程序之一，可以达到与职业棋士差距3~4子的水平

AlphaGo

- 在MCTS的基础上，通过神经网络进行训练得到更好的树策略和估值函数



结果比较

软件或人类	BayesElo
AlphaGo Zero（40 blocks版）	5422?
AlphaGo（Master版）	5231?
AlphaGo Zero（20 blocks版）	5022?
AlphaGo（Lee版）	4672?
朴廷桓	4592?
柯洁	4590?
井山裕太	4546?
李世乭	4514?
DeepZenGo	4269
AlphaGo（Fan版，176 GPU）	4122?
AlphaGo（Fan版，48 CPU与8 GPU）	3862?
GNU Go	1800

【H4】五子棋AI

- 如果规定五子棋下法如下
  - 黑棋先
  - 必须下在跟已有棋子挨着的位置
- 请画出4层的决策树
- 请定义一个局面估值函数
- 用minimax法来决策下几步棋
- 想出一个启发性规则估值来优化估值函数?