Artificial Intelligence

School of Electronic and Computer Engineering
Peking University

Wang Wenmin

Contents:

Contents:

Objectives 教学目的

- To examine the problems that arise when we try to plan ahead in a world where other agents are planning against us.

  去考察这样一些会发生的问题，当我们试图在某个环境预先规划时，其他智能体也正在针对我们做规划。

Contents:

Search vs. Adversarial Search 搜索与对抗搜索

Search 搜索	Adversarial Search 对抗搜索
Single agent 单智能体	Multiple agents 多智能体
Solution is (heuristic) method for finding goal. 解是寻找目标的（启发式）方法	Solution is strategy (strategy specifies move for every possible opponent reply). 解是策略（指定对每个可能对手回应的行动策略）
Heuristics can find *optimal* solution. 启发式法可以找到最优解	Time limits force an *approximate* solution. 时间受限被迫执行一个近似解
Evaluation function: estimate of cost from start to goal through given node. 评价函数：穿过给定节点从起始到目标的代价估计	Evaluation function: evaluate “goodness” of game position. 评价函数：评估博弈局势的“好坏”

Adversarial Search often Known as Games 对抗搜索通常称为博弈

- Definitions of Game theory 博弈论的定义
  - Study of strategic decision making. Specifically, study of mathematical models of conflict and cooperation between intelligent rational decision-makers.
    研究战略决策制定。具体来说，研究智能理性决策者之间的冲突与合作的数学模型。
  - An alternative term is interactive decision theory.
    一个可替代的术语是交互式决策理论。
- Applications of Game theory 博弈论的应用
  - Economics, political science, psychology, logic, computer science, and biology.
    经济学、政治学、心理学、逻辑、计算机科学、以及生物学。
  - Behavioral relations and decision science, including both humans and non-humans (e.g. computers).
    行为关系与决策科学，包括人类与非人类（如计算机等）。

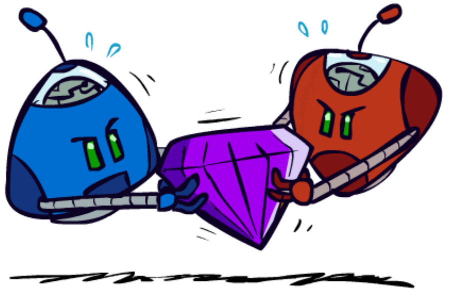
Games are Good Problems for AI 博弈是AI研究的好材料

- Machines (players) need “human-like” intelligence.
  机器（玩家）需要“类人”的智能。
- Requiring to make decision within limited time.
  要求在有限的时间内进行决策。
- Features of games: 博弈的特征：

Two, or more players (agents)	两个、或多个玩家（智能体）
Turn-taking vs. simultaneous moves	轮流、与同步行动
Perfect information vs. imperfect information	完全信息、与不完全信息
Deterministic vs. stochastic	确定性、与随机
Cooperative vs. competitive	合作式、与对抗式
Zero-sum vs. non zero-sum	零和、与非零和

Zero Sum vs. Non-zero Sum 零和与非零和博弈

- Zero sum games 零和博弈
  - Agents have *opposite utilities*.
    智能体之间是对立的方式。
  - Pure competition: win-lose, its sum is zero.
    纯竞争：输赢、其和为零。



- Non-zero sum games 非零和博弈
  - Agents have *independent* utilities.
    智能体之间是自主的方式。
  - Cooperation, indifference, competition, ...
    合作、中立、竞争、…
  - Win-win, win-lose or lose-lose, its sum is not zero.
    双赢、输赢、或双输，其和不为零。

Example: Prisoner's Dilemma 囚徒困境

- Two members of a criminal gang are arrested and imprisoned. Each prisoner is given the opportunity either to: betray the other by testifying that the other committed the crime, or to cooperate with the other by remaining silent. Here is the offer:

  有两个犯罪集团的成员被逮捕和监禁。每个囚徒只有二选一的机会：揭发对方并证明其犯罪，或者与对方合作保持沉默。惩罚方式如下：

  - If A and B each betray the other, each of them serves 2 years in prison.

    若A和B彼此揭发对方，则每个囚徒监禁2年。

  - If A betrays B but B remains silent, A will be set free and B will serve 3 years in prison (and vice versa).

    若A揭发B而B保持沉默，则A被释放而B监禁3年（反之亦然）。

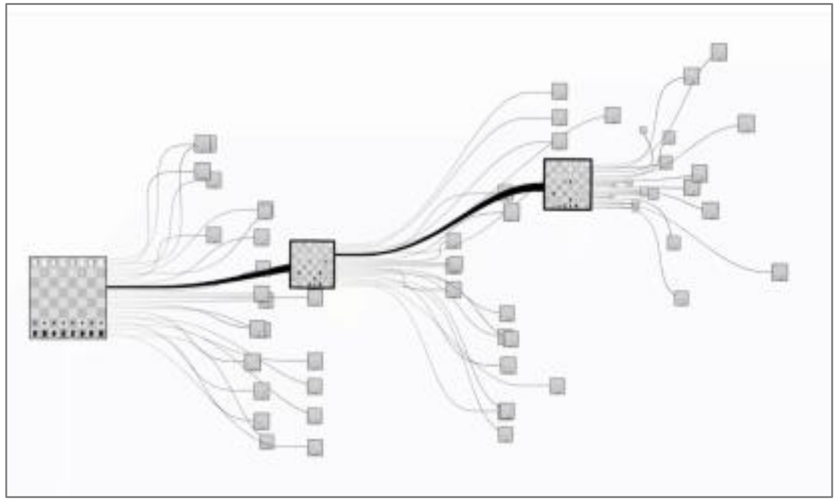
  - If A and B both remain silent, both of them will only serve 1 year in prison.

    若A和B都保持沉默，则他们仅被监禁1年。

Games are Interesting but Too Hard to Solve 博弈有趣但难以求解

E.g., Chess: average branching factor ≈ 35, each player often go to 50 moves, so search tree has about 35^{100} or 10^{154} nodes!

例如，国际象棋：平均分支数约等于35，每个对弈者常常走50多步，故该搜索树约有35^{100}或10^{154}个节点！



- Games, like the real world, therefore require the ability to make *some* decision even when calculating the *optimal* decision is infeasible.
  博弈，与现实世界相似，因而当无法算计出最优决策时，需要某种决策的能力。
- Game-playing research has spawned a number of interesting ideas on how to make the best possible use of time.
  博弈的研究已经产生了大量的有趣思想，即如何尽可能的利用时间。

Types of Games 博弈的类型

	Deterministic 确定性	Stochastic 随机性
Perfect information (fully observable) 完全信息（可完全观测）	Chess 国际象棋 Checkers 西洋跳棋 Go 围棋 Othello 黑白棋	Backgammon 西洋双陆棋 Monopoly 大富翁

(a) Checkers
西洋跳棋

(b) Othello
黑白棋

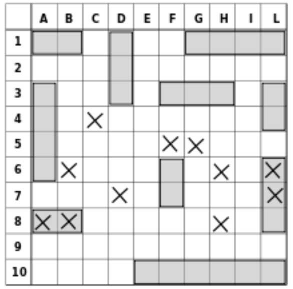
(c) Backgammon
西洋双陆棋

(d) Monopoly
大富翁

Types of Games 博弈的类型

	Deterministic 确定性	Stochastic 随机性
Imperfect information (partially observable) 不完全信息（不可完全观测）	Stratego 西洋陆军棋 Battleships 海战棋	Bridge 桥牌 Poker 扑克 Scrabble 拼字游戏

(e) Stratego
西洋陆军棋



(f) Battleships
海战棋

(g) Scrabble
拼字游戏

Origins of Game Playing Algorithms 博弈算法的起源

1912	Ernst Zermelo 恩斯特·策梅洛	Minimax algorithm 最小最大算法
1949	Claude Shannon 克劳德·香农	Chess playing with evaluation function, selective search 用评价函数和选择性搜索下国际象棋
1956	John McCarthy 约翰·麦卡锡	Alpha-beta search Alpha-beta搜索
1956	Arthur Samuel 亚瑟·塞缪尔	Checkers program that learns its own evaluation function 学习自身的评价函数的西洋跳棋程序

- Ernst Zermelo (1871–1953), a German logician and mathematician.
- Claude Shannon (1916–2001), an American mathematician, and cryptographer known as “the father of information theory”.
- John McCarthy (1927-2011), an American computer scientist and cognitive scientist, and one of the founders of AI.
- Arthur Samuel (1901-1990), an American pioneer of computer gaming, AI, and ML.

Game Playing Algorithms Today 博弈算法的进展

Computers are better than humans 计算机优于人类

Checkers 西洋跳棋	Solved in 2007 2007年已解决
Chess 国际象棋	IBM Deep Blue defeated Kasparov in 1997 IBM深蓝于1997年战胜了卡斯帕罗夫
Go 围棋	Google AlphaGo beat Lee Sedol, a 9 dan professional in Mar. 2016 谷歌AlphaGo于2016年3月战胜了9段职业棋手李世乭

Computers are competitive with top human players 计算机与顶级人类玩家媲美

Backgammon 西洋双陆棋	TD-Gammon used reinforcement learning to learn evaluation function TD-Gammon使用了强化学习方法来得到评价函数
Bridge 桥牌	Top systems use Monte-Carlo simulation and alpha-beta search 顶级的系统使用蒙特卡罗仿真和alpha-beta搜索

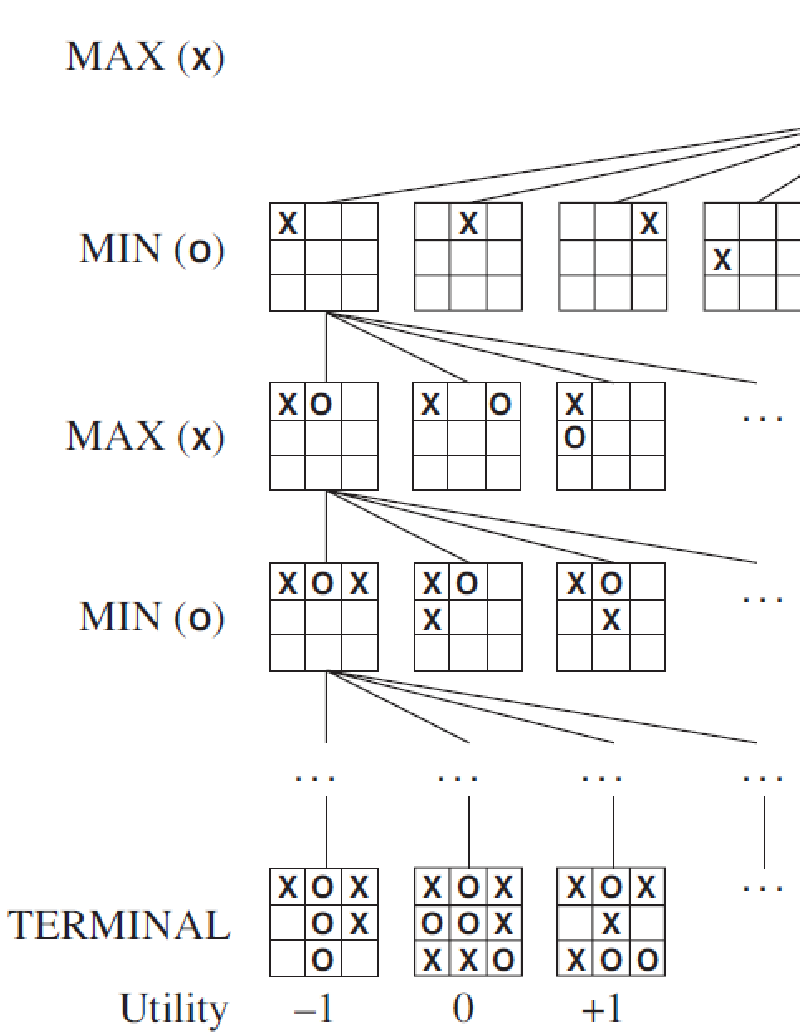
Two Players Games 两个玩家博弈

- Feature 特点
  deterministic, perfect information, turn-taking, two players, zero-sum.
  确定性、完整信息、轮流、两个玩家、零和。
- Calling the two players: 将两个玩家称为：
  MAX, MIN.
- MAX moves first, and then they take turns moving, until the game is over.
  MAX先走棋，然后轮流走棋，直到博弈结束。
- At game end 博弈结束时
  - winner: award points
    胜者：奖励点数
  - loser: give penalties.
    败者：给予处罚

Formally Defined as a Search Problem 形式化定义为搜索问题

S_0	Initial state, specifies how the game is set up at the start. 初始状态，指定博弈开始时的设定。
PLAYER(s)	Defines which player has the move in a state. 定义哪个玩家在某状态下动作。
ACTIONS(s)	Returns the set of legal moves in a state. 返回某个状态下的合法动作。
RESULT(s, a)	Transition model, defines the result of a move. 转换模型，定义一步动作的结果。
TERMINAL-TEST(s)	Terminal test, *true* when the game is over and *false* otherwise. 终止检测，博弈结束时为*true*，否则为*false*。
UTILITY(s, p)	Utility function, defines the value in state s for a player p. 效用函数，定义在状态*s*、玩家为*p*的值。

Example: Game Tree of Tic-tac-toe 井字棋的博弈树



- Part of the tree, giving alternating moves by MIN(O) and MAX(X).
  部分博弈树，MIN(O) 和 MAX(X)之间交替走子
- MAX moves first.
  MAX先走
- The game tree is relatively small, fewer than 9! = 362,880 nodes.
  该博弈树相对较小，少于9! = 362,880个节点