Foundations of Artificial Intelligence
G1. Board Games: Introduction and State of the Art

Malte Helmert

University of Basel

May 14, 2025

Foundations of Artificial Intelligence
May 14, 2025 — G1. Board Games: Introduction and State of the Art

Board Games: Overview

chapter overview:

- G1. Introduction and State of the Art
- G2. Minimax Search and Evaluation Functions
- G3. Alpha-Beta Search
- G4. Stochastic Games
- G5. Monte-Carlo Tree Search Framework
- G6. Monte-Carlo Tree Search Variants

G1.1 Introduction

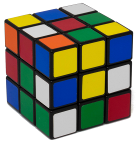
Why Board Games?

Board games are one of the oldest areas of AI
(Shannon 1950; Turing 1950).

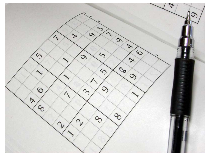
- abstract class of problems, easy to formalize
- obviously "intelligence" is needed (really?)
- dream of an intelligent machine capable of playing chess is older than electronic computers
- cf. von Kempelen's "Schachtürke" (1769), Torres y Quevedo's "El Ajedrecista" (1912)

Board Games

algorithms considered previously:



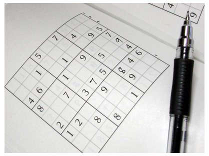
agent has full control over environment:
- agent is only acting entity
- effects of actions fully predictable



Board Games

algorithms considered previously:

agent has full control over environment:
- agent is only acting entity
- effects of actions fully predictable



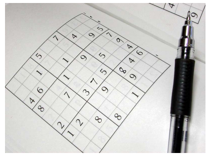
games considered now (Chapters G1–G3):

environment changes independently of agent:
- other agent (with opposing objective) acts

Board Games

algorithms considered previously:

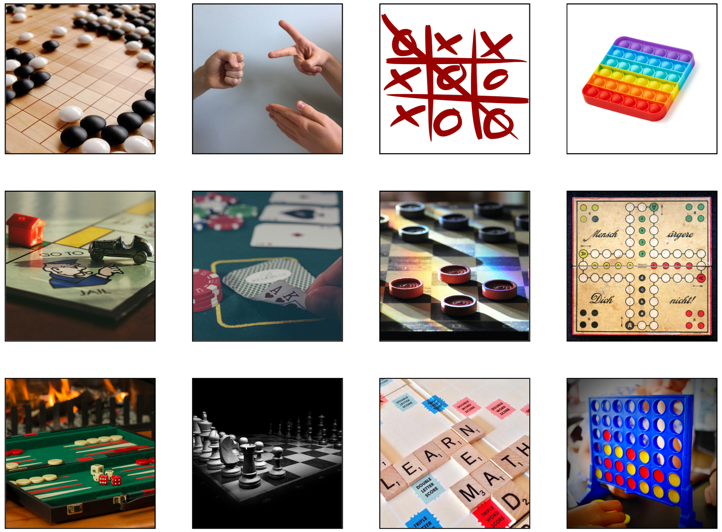
agent has full control over environment:
- agent is only acting entity
- effects of actions fully predictable



games considered later (Chapter G4):

environment changes independently of agent:
- other agent (with opposing objective) acts
- effects of actions underly chance

Applications



Game Applications Beyond Specific Board Games

video games

general game playing

cyber security

wildlife preservation

generative adversarial networks

auctions

Game Environments

game environments cover entire spectrum of properties
⇝ need some restrictions

important classes of games that we do not consider:

- with randomness (e.g., backgammon) (⇝ Chapter G4)
- with more than two players (e.g., poker)
- with hidden information (e.g., scrabble)
- with simultaneous moves (e.g., rock-paper-scissors)
- without turns (e.g., many video games)
- without zero-sum property (e.g., auctions)
- ...

many of these can be handled with similar/generalized algorithms

Properties of Games Considered (for Now)

- current situation representable by finite set of positions
- there is a finite set of moves players can play
- effects of actions are deterministic
- the game ends when a terminal position is reached
- a terminal position is reached after a finite number of steps (*)
- terminal positions yield a utility
- no randomness, no hidden information

(*) Our definitions do not enforce this, and there are some subtleties associated with this requirement which we ignore.

Properties of Games Considered (for Now)

- there are exactly two players called MAX and MIN
- both players observe the entire position (perfect information)
- it is the turn of exactly one player in each non-terminal position
- utility for MAX is opposite of utility for MIN (zero-sum game)
- MAX aims to maximize utility, MIN aims to minimize utility

Classification

classification:

Board Games

environment:

- static vs. dynamic
- deterministic vs. nondeterministic vs. stochastic
- fully observable vs. partially observable
- discrete vs. continuous
- single-agent vs. multi-agent (adversarial)

problem solving method:

- problem-specific vs. general vs. learning

Informal Description

objective of the agent:

- compute a strategy
- that determines which move to execute
- in the current position or in any (reachable) position

performance measure:

- maximize utility (given available resources)

To study board games, we need a formal model.

G1.2 Games

Example: Chess

Example (Chess)

- positions described by:
  - configuration of pieces
  - whose turn it is
  - en-passant and castling rights
- turns alternate
- terminal positions: checkmate and stalemate positions
- utility of terminal position for first player (white):
  - $+1$ if black is checkmated
  - 0 if stalemate position
  - -1 if white is checkmated

Terminology Compared to State-Space Search

Many concepts for board games are similar to state-space search.
Terminology differs, but is often in close correspondence:

- state $\rightsquigarrow$ position
- goal state $\rightsquigarrow$ terminal position
- action $\rightsquigarrow$ move
- search tree $\rightsquigarrow$ game tree

Definition

Definition (game)

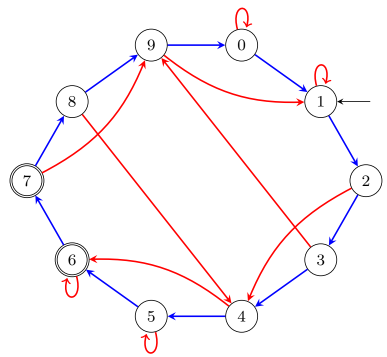
A game is a 7-tuple $\mathcal{S} = \langle S, A, T, s_\mathrm{I}, S_\mathrm{G}, \mathit{utility}, \mathit{player}\rangle$ with

- finite set of positions S
- finite set of moves A
- deterministic transition relation $T \subseteq S \times A \times S$
- initial position $s_\mathrm{I} \in S$
- set of terminal positions $S_\mathrm{G} \subseteq S$
- utility function $\mathit{utility} : S_\mathrm{G} \to \mathbb{R}$
- player function $\mathit{player} : S \setminus S_\mathrm{G} \to \{\mathrm{MAX}, \mathrm{MIN}\}$

Reminder: Bounded Inc-and-Square Search Problem

informal description:

- find a sequence of
  - increment-mod10 (*inc*) and
  - square-mod10 (*sqr*) actions
- on the natural numbers from 0 to 9
- that reaches the number 6 or 7
- starting from the number 1
- assuming each action costs 1.



Running Example: Bounded Inc-and-Square Game

informal description:

- Players alternatingly apply a
  - increment-mod10 (*inc*) or
  - square-mod10 (*sqr*) move
- on the natural numbers from 0 to 9
- starting from the number 1;
- if the game reaches the number 6 or 7
- or after a fixed number of 4 moves
- MAX obtains utility u (MIN: $-u$) where u is the current number.

formal model:

- $S = \{s_i^k \mid 0 \leq i \leq 9, 0 \leq k \leq 4\}$
- $A = \{inc, sqr\}$
- for $0 \leq i \leq 9$ and $0 \leq k < 4$:
  - $\langle s_i^k, inc, s_{(i+1) \bmod 10}^{k+1} \rangle \in T$
  - $\langle s_i^k, sqr, s_{i^2 \bmod 10}^{k+1} \rangle \in T$
- $s_I = s_1^0$
- $S_G = \{s_i^k \mid i \in \{6,7\} \vee k = 4\}$
- $utility(s_i^k) = i$ for all $s_i^k \in S_G$
- $player(s_i^k) = \text{MAX}$ if k even and $player(s_i^k) = \text{MIN}$ otherwise

Why are Board Games Difficult?

As in classical search problems, the number of positions of (interesting) board games is huge:

- Chess: roughly 10^{40} reachable positions; game with 50 moves/player and branching factor 35: tree size roughly $35^{100} \approx 10^{154}$
- Go: more than 10^{100} positions; game with roughly 300 moves and branching factor 200: tree size roughly $200^{300} \approx 10^{690}$

In addition, it is not sufficient to find a solution path:

- We need a strategy reacting to all possible opponent moves.
- Usually, such a strategy is implemented as an algorithm that provides the next move on the fly (i.e., not precomputed).

Strategies

Definition (strategy, partial strategy)

Let $\mathcal{S} = \langle S, A, T, s_I, S_G, utility, player \rangle$ be a game and let $S_{\text{MAX}} = \{s \in S \mid player(s) = \text{MAX}\}$.
A partial strategy for player MAX is a function

$$\pi : S'_{\text{MAX}} \mapsto A$$

with $S'_{\text{MAX}} \subseteq S_{\text{MAX}}$ and $\pi(s) = a$ implies that a is applicable in s.

If $S'_{\text{MAX}} = S_{\text{MAX}}$, then π is also called total strategy (or strategy).

We always take the viewpoint of MAX, but S_{MIN} and a (partial/total) strategy for MIN are defined accordingly.

Specific vs. General Algorithms

- We consider approaches that must be tailored to a specific board game for good performance, e.g., by using a suitable evaluation function.

⇝ see chapters on informed search methods

- Analogously to the generalization of search methods to declaratively described problems (automated planning), board games can be considered in a more general setting, where game rules (state spaces) are part of the input.

⇝ general game playing: regular competitions since 2005

Algorithms for Board Games

properties of good algorithms for board games:

- look ahead as far as possible (deep search)
- consider only interesting parts of the game tree
  (selective search, analogously to heuristic search algorithms)
- evaluate current position as accurately as possible
  (evaluation functions, analogously to heuristics)

G1.3 State of the Art

State of the Art

some well-known board games:

- Chess, Go: ⇝ next slides
- Othello: Logistello defeated human world champion in 1997;
  best computer players significantly stronger than best humans
- Checkers: Chinook official world champion (since 1994);
  proved in 2007 that it cannot be defeated
  and perfect game play results in a draw (game "solved")

Computer Chess

World champion Garry Kasparov was defeated by Deep Blue
in 1997 (6 matches, result 3.5–2.5).

- specialized chess hardware (30 cores with 16 chips each)
- alpha-beta search (⇝ Chapter G3) with extensions
- database of opening moves from millions of chess games

Nowadays, chess programs on standard PCs are much stronger
than all human players.

Computer Chess: Quotes

Claude Shannon (1950)

The chess machine is an ideal one to start with, since

1. the problem is sharply defined both in allowed operations (the moves) and in the ultimate goal (checkmate),
2. it is neither so simple as to be trivial nor too difficult for satisfactory solution,
3. chess is generally considered to require "thinking" for skillful play, [. . .]
4. the discrete structure of chess fits well into the digital nature of modern computers.

Alexander Kronrod (1965)

Chess is the drosophila of Artificial Intelligence.

Computer Chess: Another Quote

John McCarthy (1997)

In 1965, the Russian mathematician Alexander Kronrod said, "Chess is the drosophila of artificial intelligence."

However, computer chess has developed much as genetics might have if the geneticists had concentrated their efforts starting in 1910 on breeding racing drosophilae. We would have some science, but mainly we would have very fast fruit flies.

Computer Go

Computer Go

- The best Go programs use Monte-Carlo techniques (UCT).
- Until autumn 2015, leading programs Zen, Mogo, Crazystone played on the level of strong amateurs (1 kyu/1 dan).
- Until then, Go was considered as one of the "last" games that are too complex for computers.
- In October 2015, Deep Mind's AlphaGo defeated the European Champion Fan Hui (2p dan) with 5:0.
- In March 2016, AlphaGo defeated world-class player Lee Sedol (9p dan) with 4:1. The prize for the winner was 1 million US dollars.

G1.4 Summary

Summary

- Board games can be considered as classical search problems extended by an opponent.
- Both players try to reach a terminal position with (for the respective player) maximal utility.
- very successful for a large number of popular games
- Deep Blue defeated the world chess champion in 1997. Today, chess programs play vastly more strongly than humans.
- AlphaGo defeated one of the world's best players in the game of Go in 2016.