Computing and learning optimal strategies of two-player games

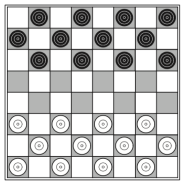
M2 Satoshi Tanaka
School of Computing
Tokyo Institute of Technology

Background: Solving Game

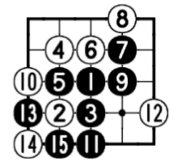
- Solving: outcome can be correctly predicted

From wikipedia

Connect-Four, **strongly solved**
Allen and Allis independently, 1988, ICGA [1]



Checkers, **weakly solved**
J Schaeffer, et al., 2007, science [2]



Go 5×5, **weakly solved**
ECD van der, et al., 2003, ICGA [3]

[1]. L.V. Allis, A Knowledge-Based Approach of Connect-Four, ICGA Journal, 11(4):165, 1988
[2]. Jonathan Schaeffer, et al., Checkers is solved, science, 2007
[3]. van der Werf, et al., Solving Go on small boards, icga Journal, 2003

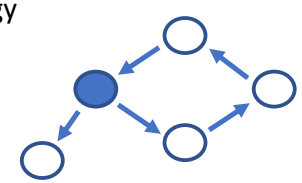
Game Length

- **Finite**: Connect-Four, Checkers, Tic-Tac-Toe
- **Not Finite**:
  - Loopy[4]: Backgammon, **Quixo**, **Novem**
  - Infinite[5]

[4] E. R. Berlekamp, Winning Ways for Your Mathematical Plays, Volume 2, secon edition, 2001
[5] Yurii Khomskii, Infinite games, Sofia University , 2010

Loopy Game

- Minmax may not be available
  - There are cycles in the search ``tree''
- Finding outcome does not mean finding a optimal strategy
  - Loop in win states[6]
  - Need a minimum step winning strategy



[6] Aviezri S Fraenkel and Ofer Rahat, Infinite cyclic impartial games, Theoretical Computer Science, 2001

Problem

- Does an optimal strategy exist?
- How to solve the game?

Goals

- **Find an optimal strategy**

From boardgamegeek

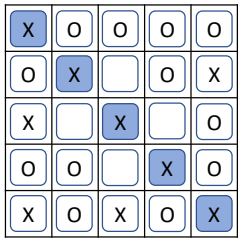
Case studies

- **Quixo**: Sequential game
- Novem[7]: Simultaneous game

[7] Fran¸cois Bonnet, Analyzing Novem, a Two-Player Multi-Stage Simultaneous Game, The 15th Workshop on Theoretical Computer Science, 2019

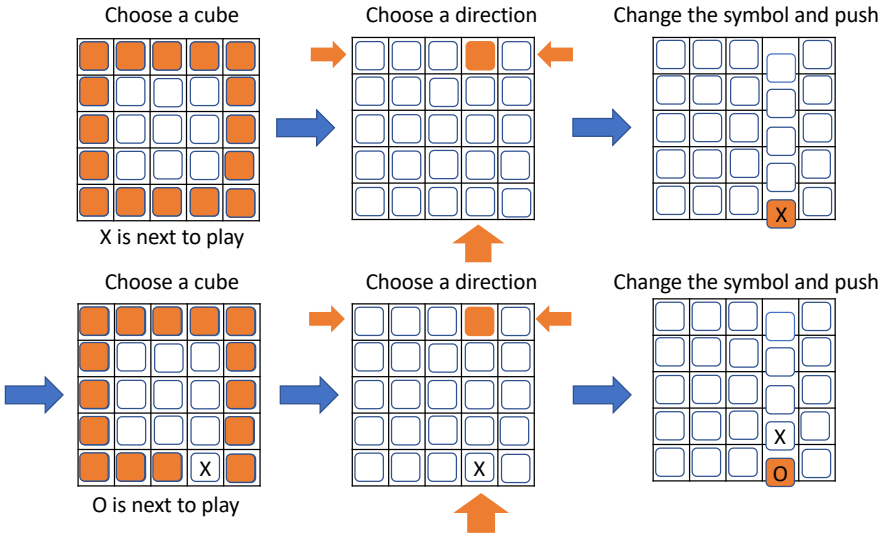
Quixo: Rule[8]

- 5×5 board
- 2 players: O and X
- Cube: {O, X, blank} O X
- **Goal: Make a line from 5 cubes** horizontal, vertical or diagonal
- Move: Choose a cube and move it
  - Choose blank or her symbol
  - Change the symbol to her symbol
  - Push the cube from the end of the row

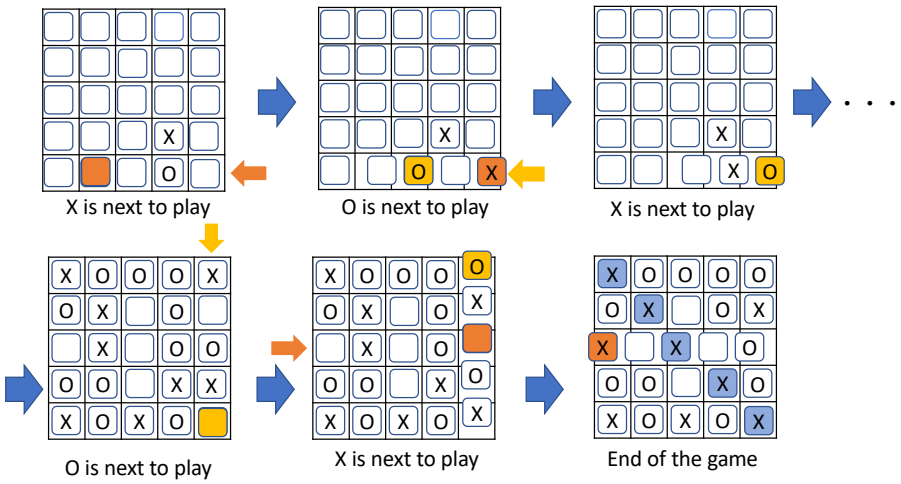


[8] Quixo page on Gigamic website. https://en.gigamic.com/game/quixo

Quixo: Rule - Move



Quixo: Rule - Move

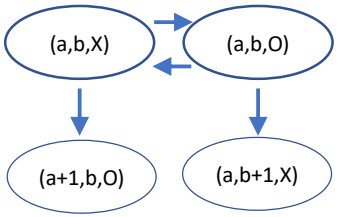


Quixo: Feature

- **Loopy game**
  - No draw rule like 50-move rule
- Sequential game
- The size of states: $3^{25} * 2 \approx 1.7 * 10^{12}$
- **The board is dynamic**
- Generalized Quixo is EXPTIME-complete[9]

[9] Shohei Mishiba and Yasuhiko Takenaga , *QUIXO is EXPTIME-complete*, 2020, Information Processing Letters

Quixo: Solving [10]



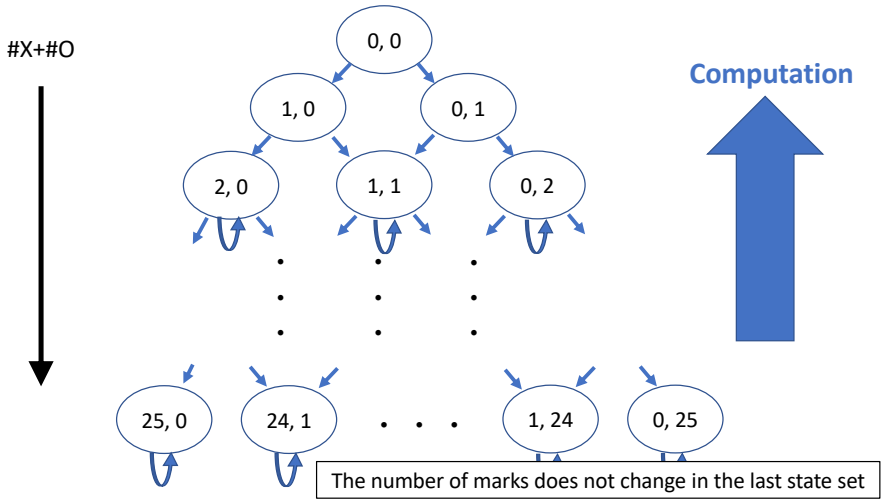
- **Backward Induction**[11][12]
  - (#X, #O, turn) is a Set of states
  - (a, b, X) → (a, b, O) or (a+1, b, O)
  - Compute from #X + #O = 25
- **Value Iteration**[13]
  - Repeat update until no update
  - Update:
    - Win if one next state is Loss
    - Loss if all next states are Win
- **Optimization**: **Memory** and **State transition**
- Parallelization

[10]Satoshi Tanaka, et al., *Quixoの強解決*, ゲームプログラミングワークショップ, 2020
Satoshi Tanaka, et al., *Quixo Is Solved*, arXiv , 2020
[11] Robert J. Aumann, Backward induction and common knowledge of rationality, Games and Economic Behavior, 1995
[12] Jonathan Schaeffer, Checkers is solved, science, 2007
[13] John W Romein and Henri E Bal, Solving awari with parallel retrograde analysis, Computer, 2003

Backward Induction

State set

※ The turn is omitted.
Both turns are included.



Value Iteration(basic algorithm)

1. Update the terminal states outcome to Win, Loss or Draw
2. Repeat update until no update

repeat
  for all states s such that outcome$[s] = Draw$ **do**
    if at least one child of s is Loss **then**
      outcome$[s] \leftarrow Win$
    else if all children of s are Win **then**
      outcome$[s] \leftarrow Loss$
until no update in the last iteration

Parallel computing

- **Parallelize Value Iteration part**
  - Need synchronization
- Other parallelization approach(not used)
  - **Parallelize Backward Induction part**
  - States set is large when #X = #O

Quixo: Result

※both players are perfect player

- 3×3 board: First player wins in 7 steps
- 4×4 board: First player wins in 21 steps
- 5×5 board: Draw (loop)

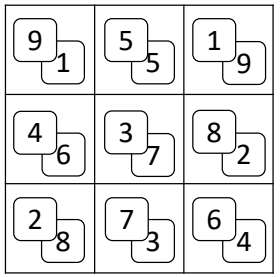
Number of States (containing unreachable states)

Board size	Win	Loss	Draw	Total
3×3	25,496	13,870	0	$3^9 * 2$
4×4	52,868,978	30,007,472	3,216,992	$3^{16} * 2$
5×5	883,630,314,618	559,492,455,912	251,454,448,356	$3^{25} * 2$

Quixo: computational time and resources

- Time
  - Single thread: **320 hours**
  - 32 threads: **32 hours**
- Resources
  - Ubuntu 18.04LTS server
  - RAM: 32GB
  - CPU: Intel Core i9-9960X

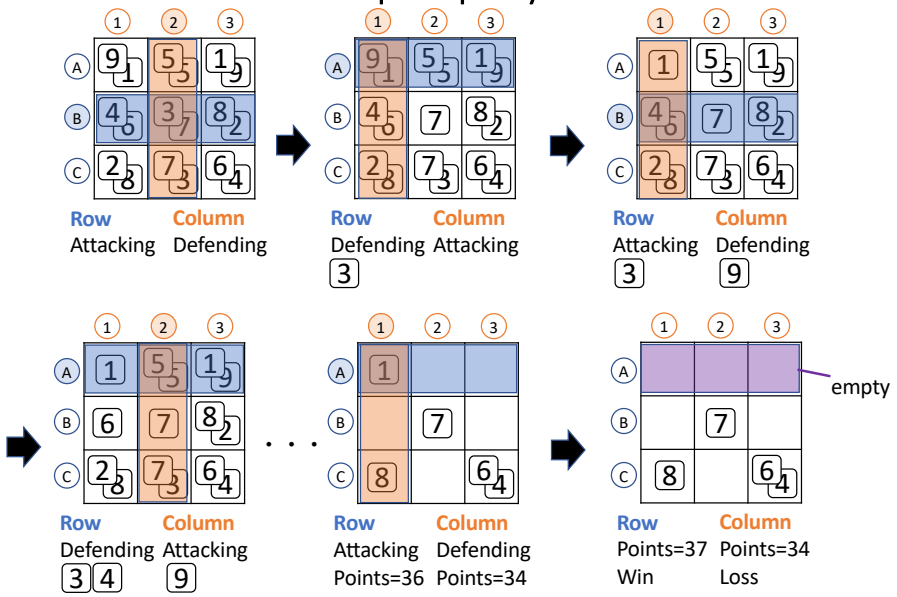
Novem: Rule[14]



- 5×5 board
- 2 players: **Row** and **Column**
- Tiles 1 to 9 on two layers
- Goal: get more points
- End: one row or column is empty
- Play:
  - Turn: attacking or defending
  - Action:
    - Row player select one row
    - Column player select one column
    - Attacking player collects the tile where the players chose row and column

[14] Novem rule website. http://www.tactic.net/site/rules/UK/02582.pdf

Novem: Example play



Novem: Feature

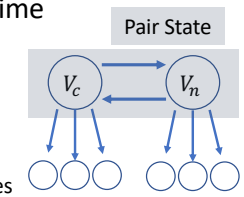
- **Simultaneous**: need mixed strategies
- Multi-stage
- **Loopy game**
- The number of states: $3^9 * 2 * 90 \approx 3.5 * 10^6$

turn

difference of points

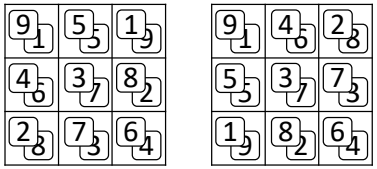
Novem: Solving

- Obviously Win state
  - One player collects more than half points
  - Sum of the remaining tiles in one row/column is smaller than the lead of the Row/Colum player
- **Linear Programing**[15]
- **Graphical solution** using **Pair State**
  - Compute 2 states outcome at the same time
  - Find the fixed point of f and g
    - $V_n = f(V_c)$
    - $V_c = g(V_n)$



[15] Branislav Bo˘sansk`y, et al., Algorithms for computing strategies in two-player simultaneous movegames, Artificial Intelligence, 2016

Novem: Result



There are only 2 initial states

Approximate **expected outcome** for the first attacking player.
Assuming that both players are perfect player.

	2-layer	1-layer
Left state	0.665	0.686
Right state	0.676	0.689

※ This is one game result.
In the official rule, there are two games changing the first turn.

Summarizing Up

- Solving Loopy Game
  - Cycles in the search ``tree''
- Problem:
  - Does an optimal strategy exist?
  - How to solve the game?
- Case studies:
  - ✓Quixo: Sequential game [9]
  - ✓Novem: Simultaneous game
- Future work
  - Other games: can the same approach be used?
    - Gobblet

From boardgamegeek

[10]Satoshi Tanaka, et al., *Quixoの強解決*, ゲームプログラミングワークショップ, 2020
Satoshi Tanaka, et al., *Quixo Is Solved*, arXiv , 2020