

CS10 : The Beauty and Joy of Computing

Lecture #16 : Computational Game Theory



UC Berkeley EECS
Summer Instructor
Ben Chun

2012-07-12

CHECKERS SOLVED!


A 19-year project led by Prof Jonathan Schaeffer, he used dozens (sometimes hundreds) of computers and AI to prove it is, in perfect play, a ... draw! This means that if two Gods were to play, nobody would ever win!

www.cs.ualberta.ca/~chinook/


Computational Game Theory

- History
- Definitions
 - Game Theory
 - What Games We Mean
 - Win, Lose, Tie, Draw
 - Weakly / Strongly Solving
- Gamesman
 - Dan's Undergraduate R&D Group
 - Demo!!
- Future




Chun, Summer 2012

UC Berkeley CS10 "The Beauty and Joy of Computing" : Computational Game Theory (2)

Computer Science ... A UCB view

www.eecs.berkeley.edu/Research/Areas/

- CS research areas:
 - Artificial Intelligence
 - Biosystems & Computational Biology
 - Computer Architecture & Engineering
 - Database Management Systems
 - Graphics
 - Human-Computer Interaction
 - Operating Systems & Networking
 - Programming Systems
 - Scientific Computing
 - Security
 - Theory
 - ...



Chun, Summer 2012

UC Berkeley CS10 "The Beauty and Joy of Computing" : Computational Game Theory (3)

The Turk (1770)

en.wikipedia.org/wiki/The_Turk

- A Hoax!
- Built by Wolfgang von Kempelen
 - to impress the Empress
- Could play a strong game of Chess
 - thanks to Master inside
- Toured Europe
 - Defeated Benjamin Franklin & Napoleon!
- Burned in an 1854 fire
 - Chessboard saved...



The Mechanical Turk (1770)


Chun, Summer 2012

UC Berkeley CS10 "The Beauty and Joy of Computing" : Computational Game Theory (4)

Claude Shannon's Paper (1950)

en.wikipedia.org/wiki/Claude_Shannon

- "Father of Information Theory"
 - Digital computer and digital circuit design theory
 - Defined fundamental limits on compressing/storing data
- Wrote "Programming a Computer for Playing Chess" paper in (1950)
 - All chess programs today have his theories at their core
 - His estimate of # of Chess positions called "Shannon #"
 • Now proved $< 2^{155} \approx 10^{46.7}$



Claude Shannon (1916-2001)

Chun, Summer 2012

UC Berkeley CS10 "The Beauty and Joy of Computing" : Computational Game Theory (5)

Deep Blue vs Garry Kasparov (1997)

[en.wikipedia.org/wiki/Deep_Blue_\(chess_computer\)](http://en.wikipedia.org/wiki/Deep_Blue_(chess_computer))

- Kasparov World Champ
- 1996 Tournament – Deep Blue
 - First game DB wins a classic!
 - But DB loses 3 and draws 2 to lose the 6-game match 4-2
 - In 1997 Deep Blue upgraded, renamed "Deeper Blue"
- 1997 Tournament – Deeper Blue
 - GK wins game 1
 - GK resigns game 2
 - even though it was draw!
 - DB & GK draw games 3-5
 - Game 6 : 1997-05-11 (May 11th)
 - Kasparov blunders move 7, loses in 19 moves. Loses tournament 3 1/2 - 2 1/2
 - GK accuses DB of cheating. No rematch.
- Defining moment in AI history



IBM's Deep Blue vs Garry Kasparov




Chun, Summer 2012

UC Berkeley CS10 "The Beauty and Joy of Computing" : Computational Game Theory (6)

www.cs.berkeley.edu/~dggarcia/eyawtkagtbwata

What is "Game Theory"?

Combinatorial

- Sprague and Grundy's 1939 Mathematics and Games
- Board games
- Nim, Domineering, dots and boxes
- Film: *Last Year in Marienbad*
- Complete info, alternating moves
- Goal: Last move

Computational

- R. C. Bell's 1988 Board and Table Games from many Civilizations
- Board games
- Tic-Tac-Toe, Chess, Connect 4, Othello
- Film: *Searching for Bobby Fischer*
- Complete info, alternating moves
- Goal: Varies**


Economic

- von Neumann and Morgenstern's 1944 *Theory of Games and Economic Behavior*
- Matrix games
- Prisoner's dilemma, auctions
- Film: *A Beautiful Mind* (about John Nash)
- Incomplete** info, simultaneous moves
- Goal: Maximize payoff

Chun, Summer 2018
UC Berkeley CS10 "The Beauty and Joy of Computing": Computational Game Theory (7)

What "Board Games" do you mean?

- No chance, such as dice or shuffled cards
- Both players have complete information
 - No hidden information, as in Stratego or Magic
- Two players (Left & Right) usually alternate moves
 - Repeat & skip moves ok
 - Simultaneous moves not ok
- The game can end in a pattern, capture, by the absence of moves, or ...

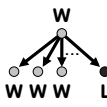


Chun, Summer 2018
UC Berkeley CS10 "The Beauty and Joy of Computing": Computational Game Theory (8)

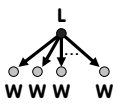
What's in a Strong Solution

- For every position
 - Assuming alternating play
 - Value ... (for player whose turn it is)
 - Winning (♠ losing child)
 - Losing (All children winning)
 - Tying (♠ losing child, but ♠ tying child)
 - Drawing (can't force a win or be forced to lose)
 - Remoteness
 - How long before game ends?

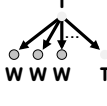
W



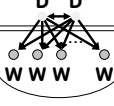
L



T



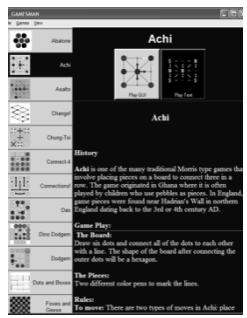
D D



Chun, Summer 2018
UC Berkeley CS10 "The Beauty and Joy of Computing": Computational Game Theory (9)

GamesCrafters

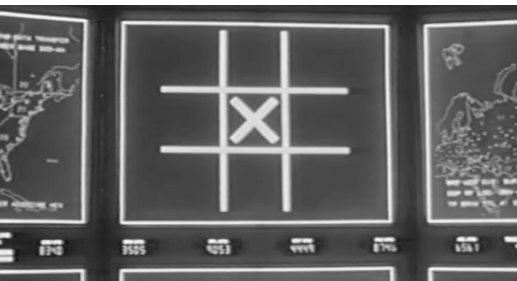
- A groups that strongly solves abstract strategy games and puzzles
 - 70 games / puzzles in our system
 - Allows perfect play against an opponent
 - Ability to do a post-game analysis



Chun, Summer 2018
UC Berkeley CS10 "The Beauty and Joy of Computing": Computational Game Theory (10)

<http://youtu.be/NHNj1Ca1rQo>

What did you mean "strongly solve"?

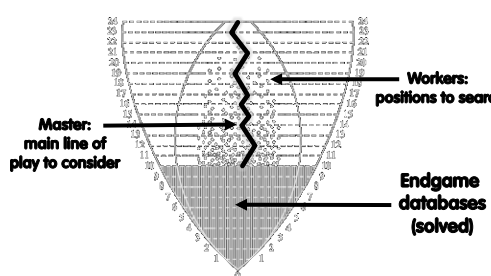


Wargames (1983)

Chun, Summer 2018
UC Berkeley CS10 "The Beauty and Joy of Computing": Computational Game Theory (11)

Thanks to Jonathan Schaeffer @ U Alberta for this slide...

Weakly Solving A Game (Checkers)




Log of Search Space Size

Chun, Summer 2018
UC Berkeley CS10 "The Beauty and Joy of Computing": Computational Game Theory (12)

Strong Solving Example: 1,2,...,10

- Rules (on your turn):**
 - Running total = 0
- Rules (on your turn):**
 - Add 1 or 2 to running total
- Goal**
 - Be the FIRST to get to 10
- Example**
 - Ana: "2 to make it 2"
 - Bob: "1 to make it 3"
 - Ana: "2 to make it 5"
 - Bob: "2 to make it 7" → photo
 - Ana: "1 to make it 8"
 - Bob: "2 to make it 10" I WIN!

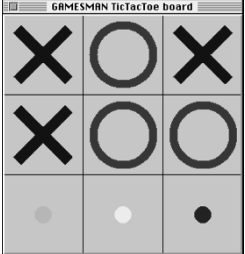


7 ducks (out of 10)

Chen, Summer 2012
UC Berkeley CS10 "The Beauty and Joy of Computing": Computational Game Theory (13)

Example: Tic-Tac-Toe

- Rules (on your turn):**
 - Place your X or O in an empty slot on 3x3 board
- Goal**
 - If you make 3-in-a-row first in any row / column / diag, win
 - Else if board is full with no 3-in-row, tie
- Misère is tricky**
 - 3-in-row LOSES
 - Pair up and play now, then swap who goes 1st

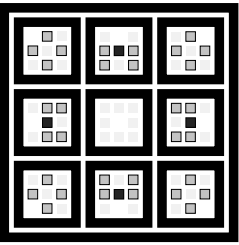


Values Visualization for Tic-Tac-Toe

Chen, Summer 2012
UC Berkeley CS10 "The Beauty and Joy of Computing": Computational Game Theory (14)

Tic-Tac-Toe Answer Visualized!

- Recursive Values Visualization Image**
- Misère Tic-tac-toe**
 - Outer rim is position
 - Inner levels moves
 - Legend
 - Lose
 - Tie
 - Win




Misère Tic-Tac-Toe 2-ply Answer

Chen, Summer 2012
UC Berkeley CS10 "The Beauty and Joy of Computing": Computational Game Theory (15)

GamesCrafters (revisited)

- Undergraduate Computational Game Theory Research Group**
- 300 students since 2001**
 - We now average 20/semester!
 - They work in teams of 2+
- Most return, take more senior roles (sub-group team leads)**
 - Maximization (bottom-up solve)
 - Oh, DeepaBlue (parallelization)
 - GUI (graphical interface work)
 - Retro (GUI refactoring)
 - Architecture (core)
 - New/ice Games (add / refactor)
 - Documentation (games & code)




Chen, Summer 2012
UC Berkeley CS10 "The Beauty and Joy of Computing": Computational Game Theory (16)

<http://nyc.cs.berkeley.edu:8080/gcweb/ui/game.jsp?game=connect4>

Connect 4 Solved, Online!

- Just finished a solve of Connect 4!!**
- It took 30 Machines x 8 Cores x 1 weeks**
- Win for the first player (go in the middle!)**
 - 3,5 = tie
 - 1,2,6,7 = lose
- Come play online!**



Chen, Summer 2012
UC Berkeley CS10 "The Beauty and Joy of Computing": Computational Game Theory (17)

Future

- Board games are exponential**
 - So has been the progress of the speed / capacity of computers!
 - Therefore, every few years, we only get to solve one more "ply"
- One by one, we're going to solve them and/or beat humans**
 - We'll never solve some
 - E.g., hardest game: Go
- Strongly solving (GamesCrafters)**
 - We visit EVERY position, and know value of EVERY position
 - E.g., Connect 4
- Weakly solving (Univ Alberta)**
 - We prove game's value by only visiting SOME positions, so we only know value of SOME positions
 - E.g., Checkers

17408965065903192790718
8238070564367946602724
950263541194828118706801
05167618464984116279288
98871493861209698881632
07806137549871813550931
2951480336966057289307
5468180597603

Go's search space ~ 3³⁶¹

Chen, Summer 2012
UC Berkeley CS10 "The Beauty and Joy of Computing": Computational Game Theory (18)