Solution Methods for Two Player Games

ACG 2023

Martin Müller, Computing Science and Amii, University of Alberta

November 2023

Solution Methods for Two Player Games

Goals of this talk

- Update on 2002 survey by v.d. Herik, Uiterwijk and v. Rijswijck
- Much progress since then…
- Very brief introduction
- Focus on the many successes in solving
  - Grouped by methods
- Current and future challenges

Credits: most illustrations are from the original publications

Games Solved: Now and in the Future

van den Herik, Uiterwijk and van Rijswijck, AI Journal 2002

- Survey of the field as it was 21 years ago
- Focus on two player, perfect information, zero sum games
- Terminology from Allis' 1994 PhD thesis
  - **ultra-weakly solved**: winner known, no strategy known
  - **weakly solved**: winner and strategy for standard starting position known
    - The default in this talk, if not stated otherwise
  - **strongly solved**: winner and strategy for all legal positions known
  - Classify games by both **state space** and **game tree complexity**

What Games were Solved in the last 20 Years?

- Much progress in both algorithms and specific games
- Some highlights:
  - Awari solved (2002)
  - Checkers solved (2007)
  - Heads-up limit poker solved (2015)
  - Othello solved (2023)
- Other games solved
- Many more games partly solved (on smaller boards, endgames, …)
- Review - still focus on two player, perfect information, zero sum games
- Short detour to imperfect information games

Basics of Solving Games

Setting: two players, zero sum, perfect information

- Solve = find result if both players play optimally
- Optimal play = follow minimax
  - Player **max**imizes their score
  - Opponent **min**imizes their score
  - Exact results in terminal states, determined by rules of the game
  - Recursive procedure to evaluate all earlier states, max(min(max(…),…),…)

Many Solution Methods

- Large-scale computation, brute force - find all legal states and their minimax values
  - Retrograde analysis
- **Forward search**
  - Alphabeta, proof number search, df-pn, Monte Carlo Tree Search
- Knowledge-based methods
  - Exact minimax values without search
  - Examples:
    - Use mathematical properties of game
    - Show that a state has the same value as another
  - Heuristics help speed up the search, as in players, such as move ordering in alphabeta

Alphabeta and Boolean Minimax Search

- Still a standard approach
- Many of the games in this talk were solved with this algorithm
- Huge number of enhancements and heuristics
  - Iterative deepening
  - Transposition tables
  - A general solution to the graph history interaction problem (Kishimoto and Müller, AAAI 2004)
  - Move ordering heuristics
  - Many heuristics for game-playing also speed up an exact solver

Proof Number Search and Df-pn

- Proof-number search (Allis, van der Meulen, and van den Herik, AI Journal 1994)
- Depth-first Proof-number Search (df-pn) algorithm (Nagai GPW 1999, PhD thesis 2002)
- Powerful family of best-first solving algorithms
  - Proof number = optimistic estimate of proof cost
- Df-pn can run in very little memory
- Many enhancements and variants, very successful
- Survey: Kishimoto, Winands, Müller and Saito. Game-Tree Search Using Proof Numbers: The First Twenty Years. ICGA 2012.

Monte Carlo Tree Search Solver

Winands, Björnsson and Saito, CG 2008

- Modify Monte Carlo Tree Search (MCTS)
- In-tree backup of proven wins and losses using minimax
- Extra proof information is stored in nodes
- Can be better than Alphabeta, df-pn in some cases
- Exact performance differences still unclear (future work)

Specific Solving Techniques and Examples

Large Scale Computation
Databases

- Strongly solve game
- Brute force
  - Compute all possible states and their values
- Bottom up computation
  - “Retrograde analysis” - see next slide
- Example shown two slides later: Awari (solved 2002)
- Another example: Irving (2014), Pentago is a first player win, massive computation on Cray supercomputer, $3x10^{15}$ states, *https://naml.us/paper/pentago/, https://perfect-pentago.net/*

Retrograde Analysis

Basic procedure here, many variations

- Strongly solve (part of) a game
  - Build complete database - all states and their minimax values
- Search proceeds *backwards* from end of game towards the start
- Initialize database with all known values, such as terminal states
- Compute and store all wins and losses in n=1 move
- Compute and store all wins, losses in n=2 moves, etc
- Continue until no more wins or losses in n moves found (fixed point in computation)
- Which states are left? Use game rules to assign value, e.g. all remaining are draws

Awari is Solved (2002)

Romein, ICGA Journal 2002

Romein and Bal, Solving Awari with Parallel Retrograde Analysis, Computer, 2003

- 889,063,398,406 reachable board positions
- Best move and minimax score computed for all - start position is **draw**
- Retrograde analysis - solved all

  positions with 1, 2, 3, …48 stones
- Parallel system: 144 processors,

  72 Gb memory, fast interconnect
- Smart parallel algorithm, how to use

  memory vs disk space

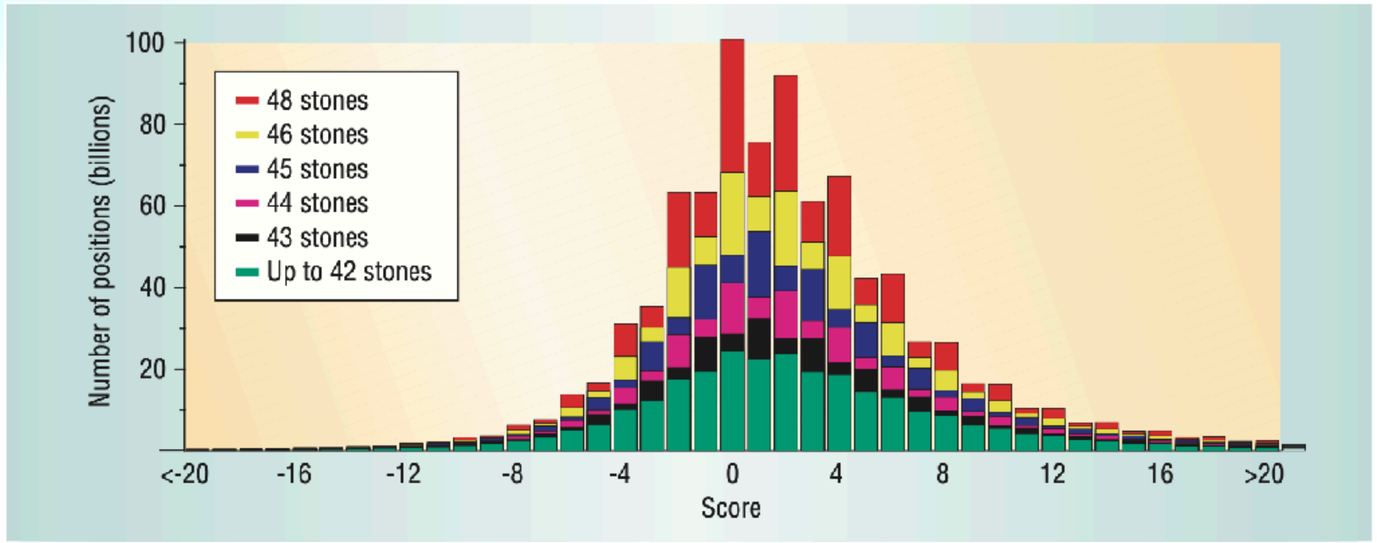


Figure 6. Distribution of the awari position scores. The greatest number of positions have a score of 0, the average is +2.29. Sixty percent of the positions are wins (all bars with a score of at least 1), and 11.3 percent are draws (including the opening position).

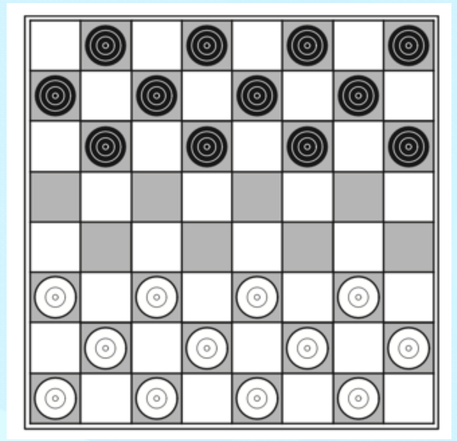
Large Scale Computation

Forward Search plus Databases

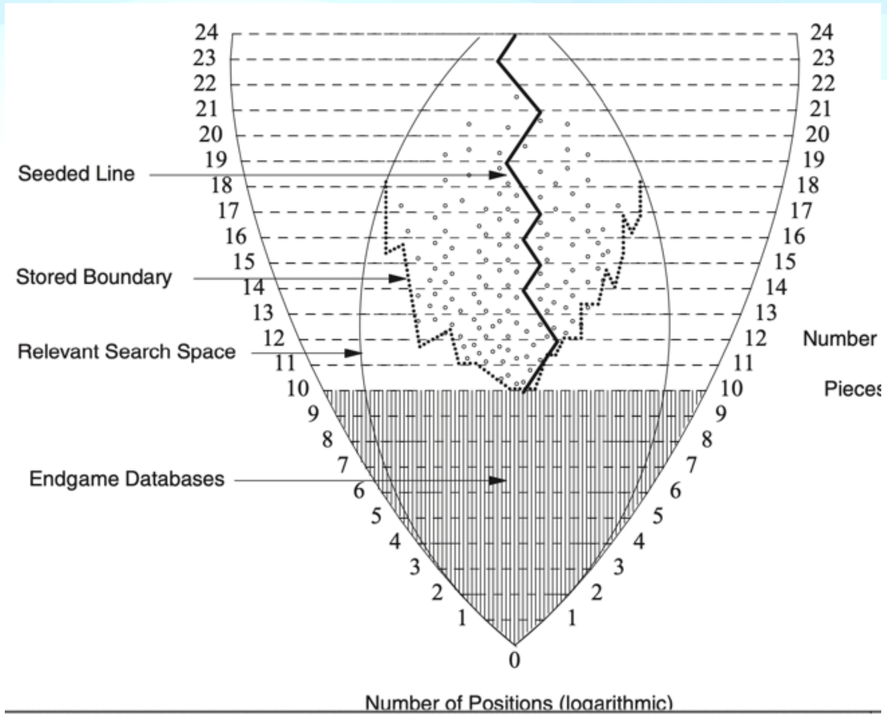
- Work on game from both ends
- Search from start of game
- Build large endgame databases to cover last phase of the game
  - Reduces search depth needed
- Often: parallel search, expand opening book on disk until proof is complete
- Examples: checkers, Othello

Checkers is Solved (2007)

Schaeffer, Burch, Björnsson et al, Science



- Checkers proven to be a draw
- Massive endgame databases, all up to 10 pieces
- Massive opening book, with strong human-seeded lines
- Large-scale independent parallel searches to bridge the gap from book to endgame database
- Two strong solvers, alphabeta and df-pn
  - intermediate searched positions not stored
- Compute enough exact values and bounds to show that the initial position is a draw



Othello is Solved (2023)

Takizawa, preprint

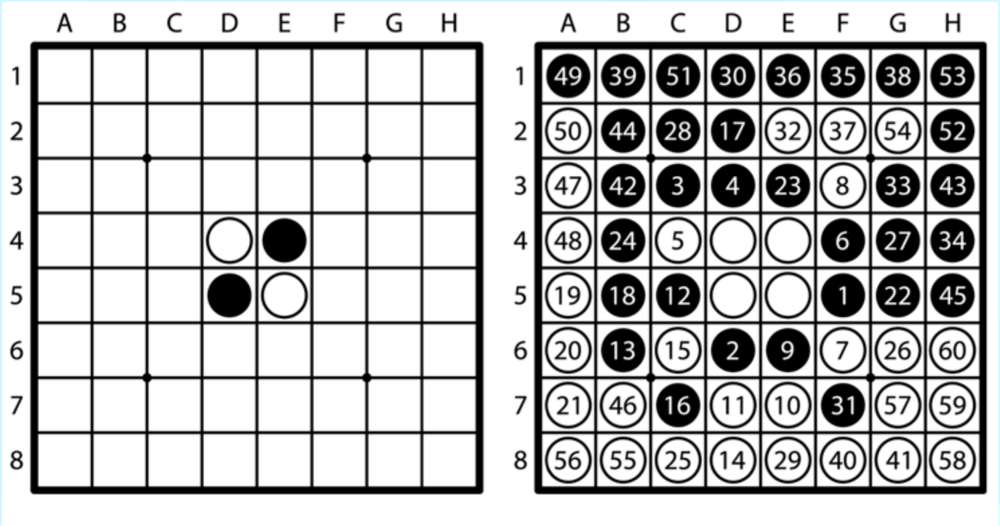


Figure 1: (Left) The initial board position of 8×8 Othello. (Right) A diagram of an optimal game record designated by our study. The game record is "F5D6C3D3 C4F4F6F3 E6E7D7C5 B6D8C6C7 D2B5A5A6 A7G5E3B4 C8G6G4C2 E8D1F7E2 G3H4F1E1 F2G1B1F8 G8B3H3B2 H5B7A3A4 A1A2C1H2 H1G2B8A8 G7H8H7H6". The numbers in stones indicates the order of moves, and the colors of stones indicates the final result. Our study confirms that if a deviation from this record occurs at any point, our software, playing as the opponent, is guaranteed a draw or a win.

- Othello is a draw
- Massive computation
- 10^{28} state space, 10^{58} possible games, searched 10^{18} positions
- 3-step search: Evaluate all legal positions with 50 empty by a strong heuristic
- Choose a subset that would prove a draw, if those 2587 positions are proven
- Search from there to about 1.5 billion positions with 36 empty squares
- Use strong solver Edax to solve all these positions on three supercomputers MN-2A, MN-2B, and MN-3

Factor	Description	Number
Problems	Number of solved problems to weakly solve Othello	$\sim 1.5 \times 10^9$
CPU	Searching capability using Edax (positions/GHz/core/sec)	$\sim 1.2 \times 10^7$
Positions	Number of searched positions (reported by Edax)	$\sim 1.5 \times 10^{18}$

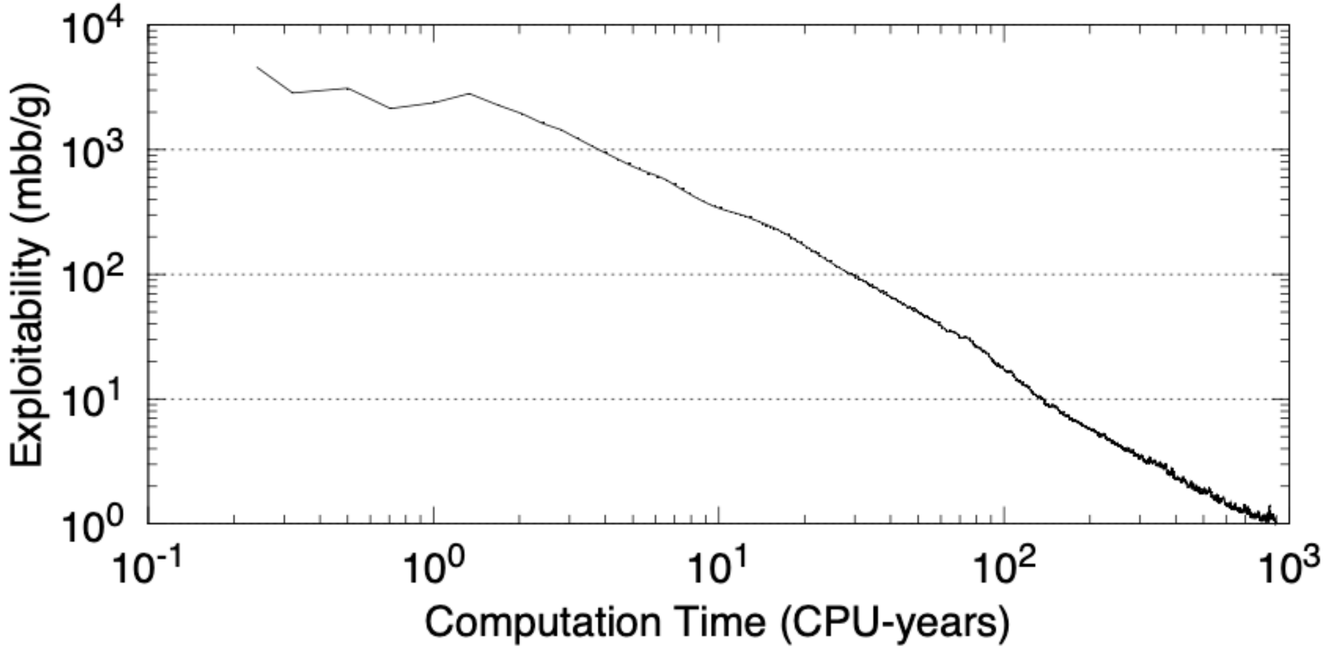
Games with Imperfect Information

- Imperfect information: parts of state are hidden from player
- Example: opponent's private information, such as cards in hand
- Example: "dark" games - opponent's moves is hidden from player
- Example: simultaneous move games
- Core theoretical principle: **Nash equilibrium**
- Two player zero sum games: all Nash equilibria have the same value
  - Expected value if two perfect players play very many games
  - Requires probabilistic, **mixed strategies** in general, as in rock-paper-scissors
- Two highlight examples: poker, Sumo game

Heads-up Limit Hold'em Poker is Solved

Bowling, Burch, Johanson, Tammelin, Science 2015

- Poker: family of card games, with incomplete information, bluffing
- Heads-up: two players
- Limit: size of bets is controlled, small number of actions
- About 3×10^{14} "decision points"
- Standard iterative algorithm: counterfactual regret minimization (CFR)
- Improvement CFR+ converges towards Nash Equilibrium in practice
- Exploitability = by how much can an opponent beat you if they "know everything"
- "Essentially solved" = "a lifetime of play is unable to statistically differentiate it from being solved at 95% confidence"



- 200 node cluster, 200*24 cores, 200*32Gb memory
- Split game into 110,565 subgames
- 1579 CFR+ iterations
- 68.5 days, 900 core-years of computation
- 10.9 TB of disk space

Solving the Oshi-Zumo Game

Buro, ACG 2004

- Each player has 50 units of energy
- Players start in the middle of the ring
- Simultaneous moves - select energy use for current round (minimum bid M)
- If use more energy than opponent - push one square forward
- Goal: push opponent off the board, or advance more by the end
- Very complex mixed strategies

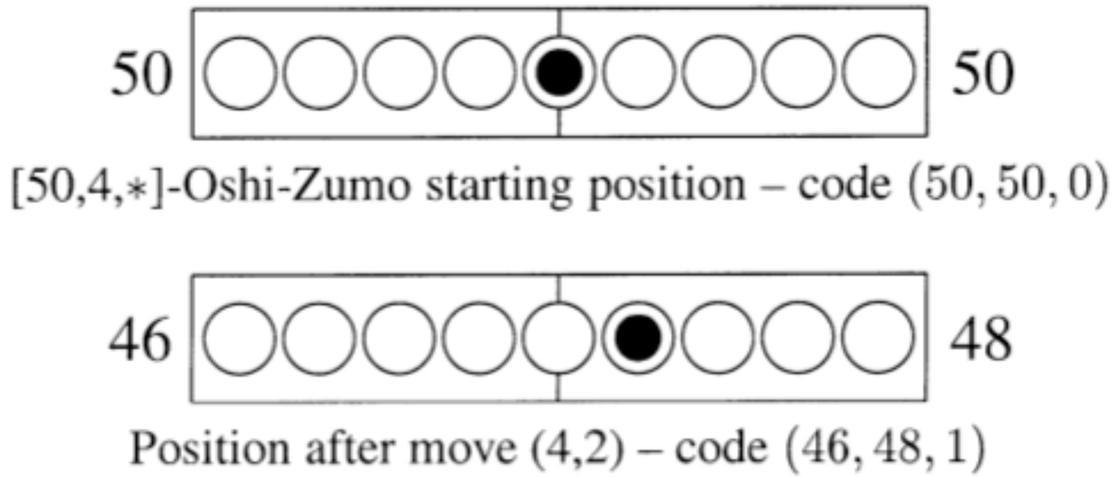


Figure 1. Oshi-Zumo positions and their triple representation.

$M = 0$ position=$(50, 50, 0)$ value= 0.0

bids:	0	1	2	3	4	5	6	7	8	9	10
prob:	.083	.077	.088	.083	.092	.088	.097	.092	.099	.094	.101

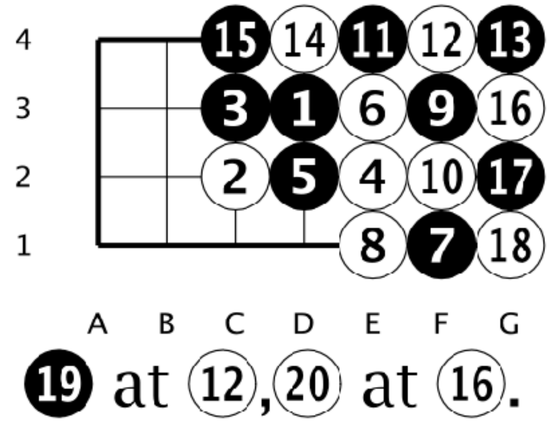
$M = 1$ position=$(50, 50, 0)$ value= 0.0

bids:	1	2	3	4	5	6	7	8	9
prob:	.139	.053	.146	.060	.152	.067	.156	.068	.156

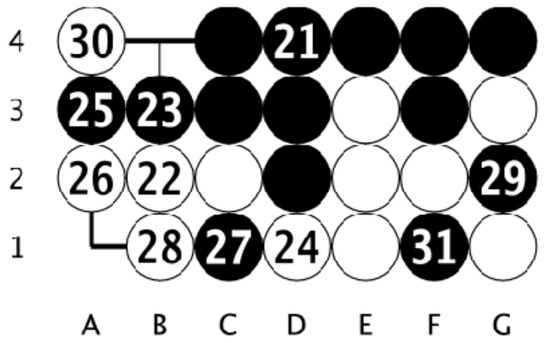
Solving Go On Small Boards (2003, 2009)

Van der Werf, Van den Herik and Uiterwijk, ICGA Journal 2003
Solving Go for Rectangular Boards, V. d. Werf and Winands, ICGA Journal 2009

- Game-specific knowledge: Search with early termination by recognizing safe stones and territory
- Found several mistakes in previous human analysis
- In Go, search can be deeper than number of points on board, due to capture
- Different rule variants also affect difficulty
- Boards solved under at least one rule set:
  - 1xn up to 1x12, 2xn up to 2x10
  - 3xn up to 3x9, bound on 3x10
  - 4xn up to **4x7**, 5x5 (in 2003), **5x6**



(a) Optimal 1–20 (B+28).



(b) Continuation.

Hex

- Hex - connect two opposite sides
- Ultra-weakly solved
  - Hex is a first player win on all nxn boards (Nash 1952)
  - Mathematical proof, no concrete strategy
- Weakly solved board sizes
- Up to 6x6, strongly solved (van Rijswijk 2000)
- 7x7 some first moves (Yang 2001), all first moves (Hayward et al 2003)
- 8x8 center (Yang 2002), all (Henderson, Arneson and Hayward, IJCAI 2009)
- 9x9 center (Yang 2003), many more: Solving Hex: Beyond Humans (Arneson, Hayward and Henderson, CG 2010), all (Pawlewicz 2012)
- 10x10 center (Pawlewicz and Hayward 2013)

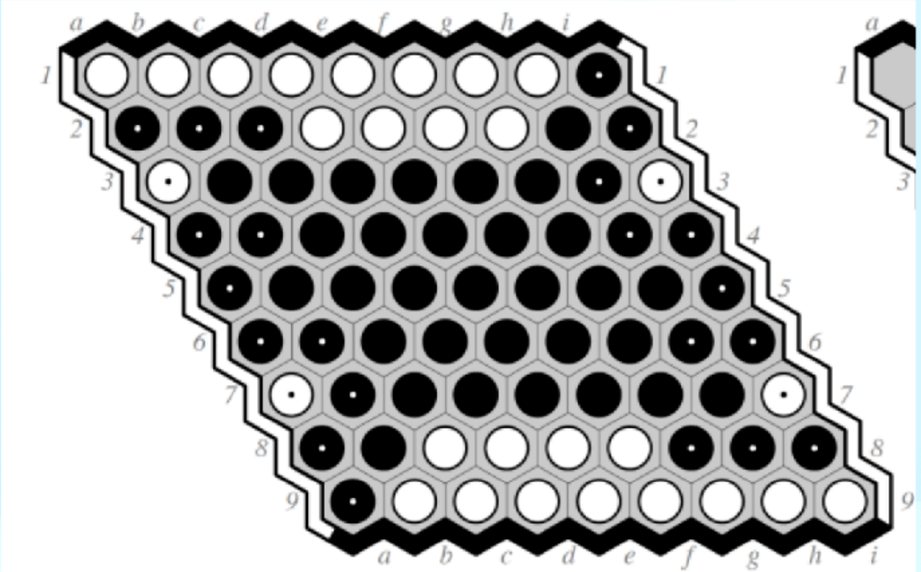
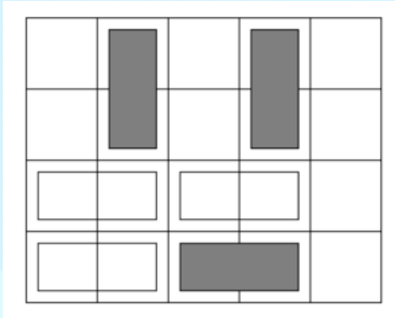


Fig. 1. Newly solved 9×9 opening values (dots), winner if black opens there.

Game-specific Knowledge

- Domineering
  - Place domino, one player horizontally, other vertically
  - Very strong game-specific rules
  - Massive search reduction
  - 10x10: first player win (Bullock 2002)
  - 11x11: first player win (Uiterwijk 2016)
- Safe stones and territories in Go
  - Several generations of strong game-specific knowledg
  - Latest: Randall et al, Improving Search in Go Using Bounded Static Safety, CG 2022
  - Helps game solver by finding wins/losses (much) earlie.



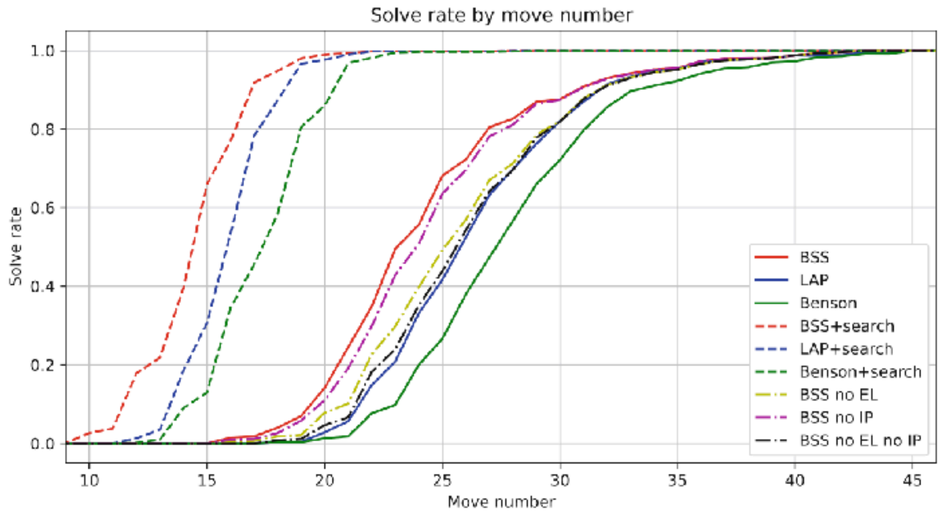
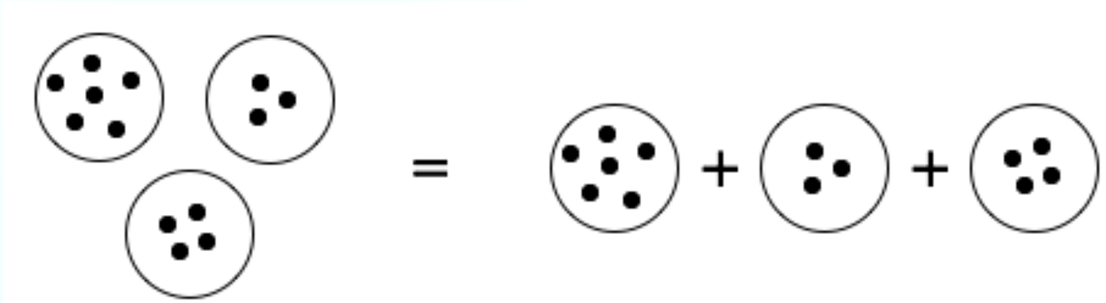


Fig. 5. Solve rate of the positions in 600 6 × 6 games by the move number of each position for each safety evaluation type. (Color figure online)

Search and Combinatorial Game Theory (CGT)

- Combinatorial games: break up a game into a *sum of independent subgames*
- Use CGT concepts to speed up search



An Enhanced Solver for The Game of Amazons (2015)

Song and Müller, IEEE Transactions on Computational Intelligence and AI in Games

- Solve Amazons on small rectangular boards
- 4 amazons each, as in standard 10x10 game
- 4x5, 5x4, 5x5 (Müller in 2001), 4x6, 5x6, and 4x7 all first player wins
- 6x4 is second player win (!)
- Approach
  - Massive databases for “territories” and local subgames
  - Means and temperatures for local subgames
  - Search algorithm uses databases to determine early wins/losses
  - Effective even though number of subgames is small

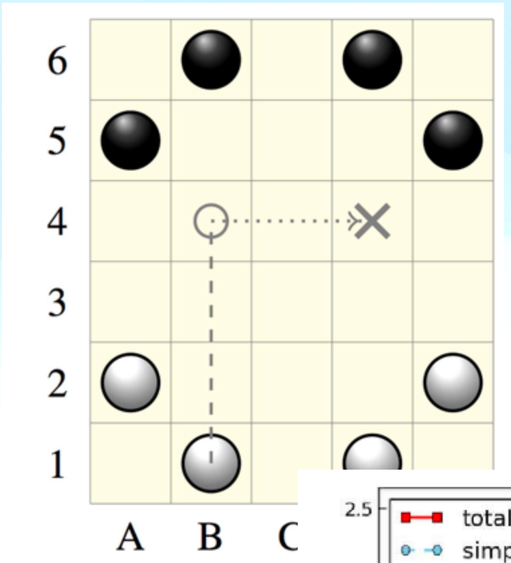
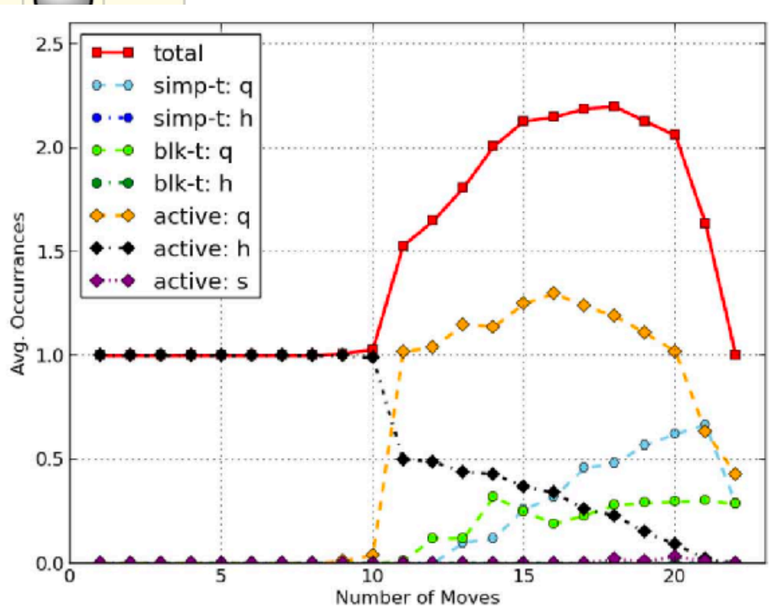


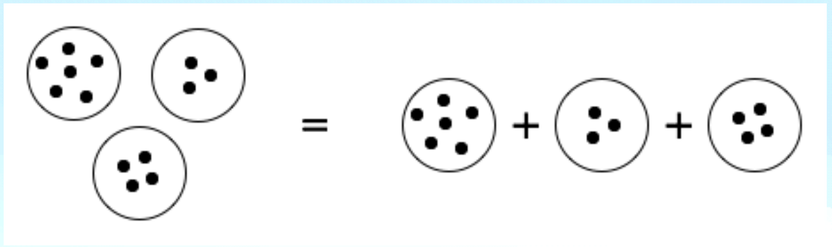
Fig. 1. 5×6 *Amazons* starting position an



Impartial Games: Nimbers are inevitable (2012)

Lemoine and Viennot, Theoretical Computer Science

- Impartial game: special type of combinatorial game
- Both players have the same move options in each state
- Each game is equivalent to some Nim value, “Nimber”
- Specialized algorithm
- Efficient way to compute Nimbers
- Big breakthrough, massive improvement on previous work
  - See next slide



- Recent work: Beling and Rogalski, On pruning search trees of impartial games, AIJ 2020

Impartial Games: Sprouts (2012)

- **Sprouts**:
  - Connect two dots
  - Add dot in between
  - Lines cannot cross
- Previous state of the art: solved up to 11 dots
- Lemoine and Viennot solved all Sprouts starting positions up to 44 points, plus 46, 47 and 53 (!!)
- Enormous progress, given the exponential growth in problem difficulty

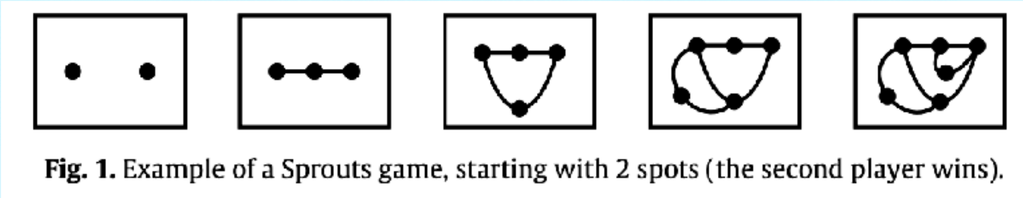


Fig. 1. Example of a Sprouts game, starting with 2 spots (the second player wins).

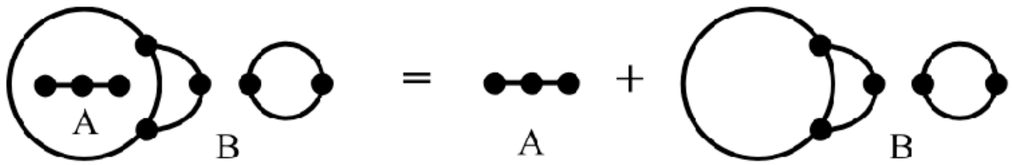


Fig. 3. Splittable Sprouts position.

Impartial Games: Cram (2012, 2019)

- Impartial domineering, no restriction on horizontal/vertical
- L+V: Exact Nim values for 3xn up to 3x18
  - Previously known: up to 3x9
- Also: 4×7, 4×9, 5×6, 5×8 boards, partial results for 5×9, 7×7, 6×7
- Uiterwijk, Solving Cram using Combinatorial Game Theory, ACG 2019
  - Win/loss results for Cram 3x19, 3x21, 5x11

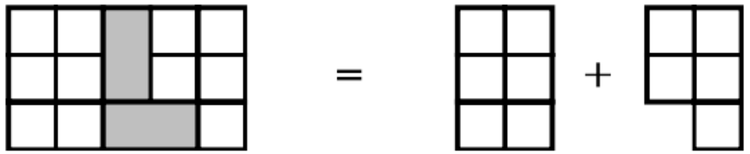
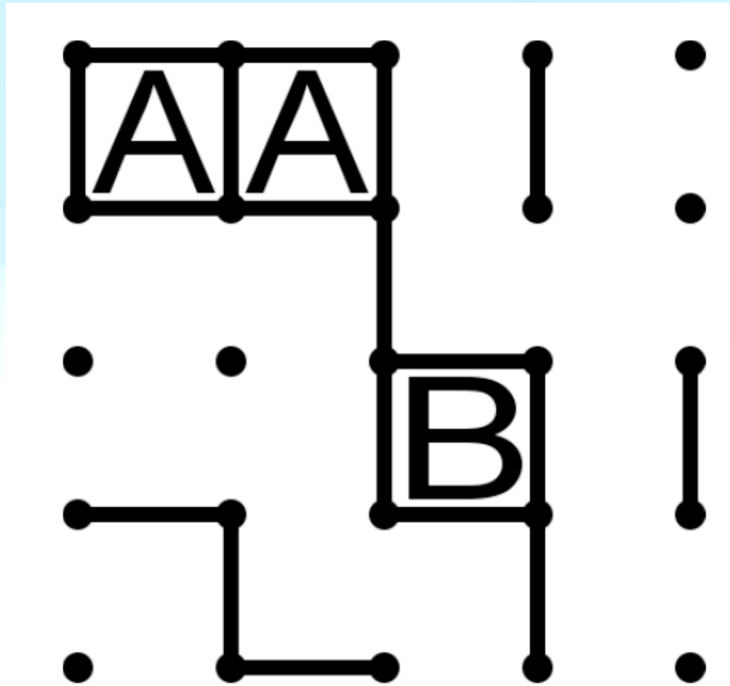


Fig. 4. Splittable Cram position.

Solving Dots-And-Boxes

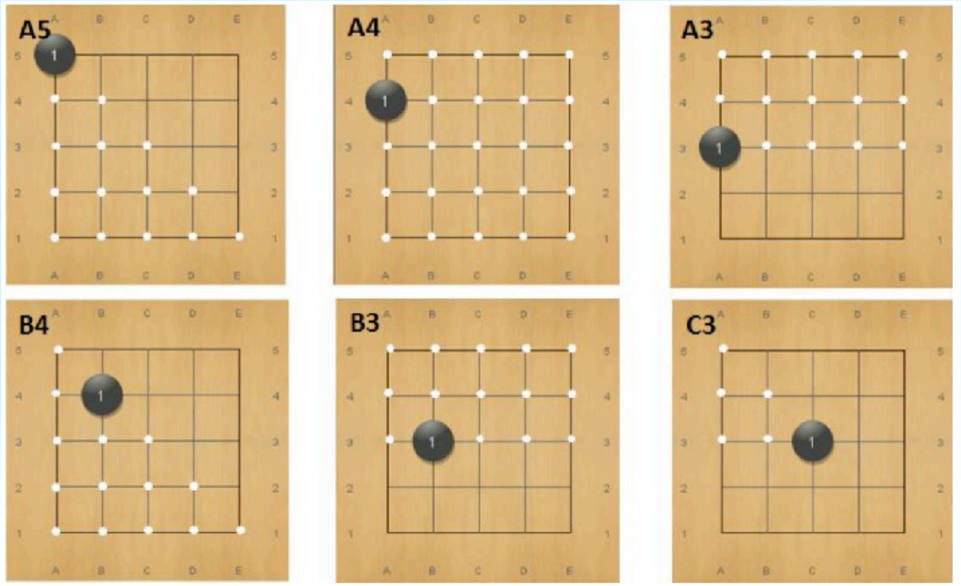
Barker and Korf, AAAI 2012

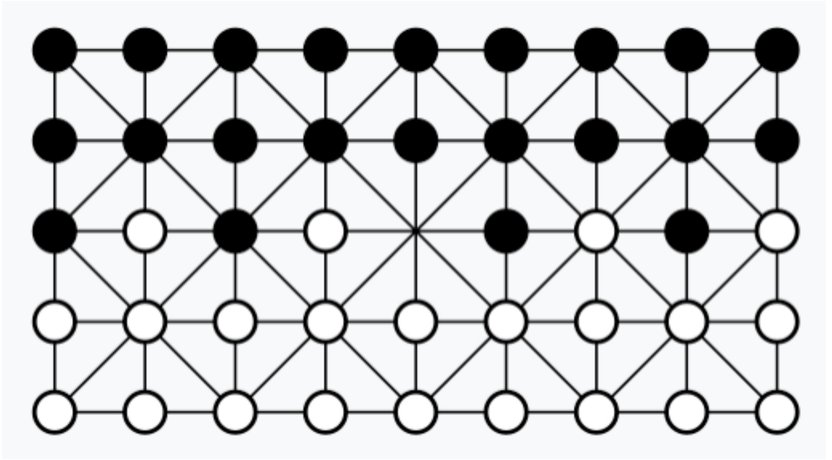
- Impartial game, quite popular
- Many connections to CGT, but not a simple sum
- Approach: Alphabeta search
  - Many generic and game-specific search improvements
- Solved boards up to 1x12, 2x7, 3x5, **4x5**
- **Result:** 4x5 is tie in terms of number of boxes
- Wilson (2010) solved many boards up to 4x4



NoGo and Fanorona

- **NoGo**
  - No-capture Go
  - 4x4 is second player win
  - 5x5 is first player win (She 2013)
  - All rectangular boards up to 27 points (Cazenave 2020; Du, Wei and Müller 2023 in this conference)
- **Fanorona**
  - Proven a draw (Schadd et al 2007)
  - Endgame database
  - Variant of Proof-number search





Some Games I Know Too Little About

Please tell me more…

- Draughts (10x10 checkers)
  - Large endgame databases exist?
- Chinese chess
  - Massive endgame databases exist
- Backgammon: Large endgame databases exist?
  - Expected value calculation, due to dice

Towards the Future

What About Deep Learning?

- Deep Learning and RL are extremely successful for game *playing*
  - AlphaGo, AlphaZero, KataGo etc.
  - Many successful related algorithms - AlphaStar, SuphX Mahjong, Gran Turismo, Student of Games, …
- Challenge: How to use this technology for solving?
  - Main problem: overcome speed penalty of network evaluation
  - Player: search can build narrow game tree **for both players**
  - Solver: **cannot ignore any moves** for losing side
    - Much worse scaling
    - Massive amount of extremely unlikely states must still be solved
    - Example: proof cost network

AlphaZero-based Proof Cost Network to Aid Game Solving

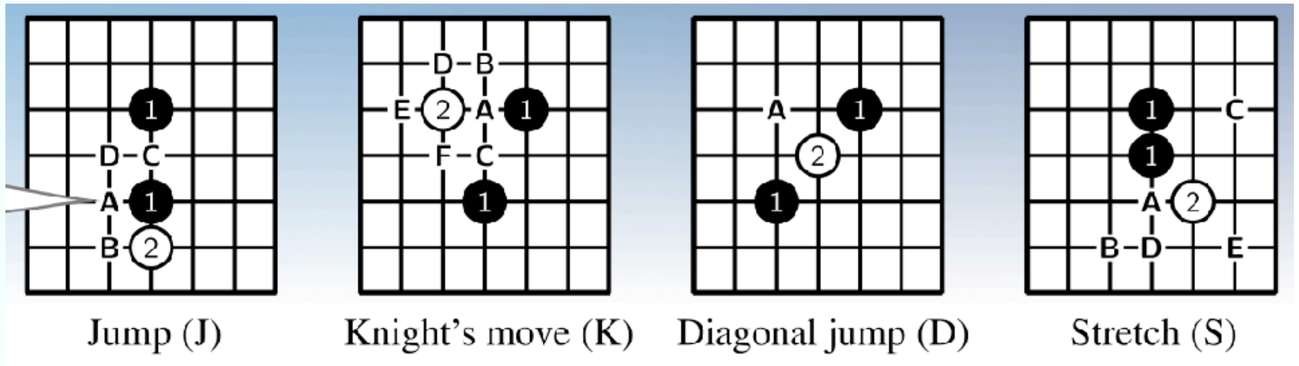
Wu et al, ICLR 2022

- Proof cost network
- Trained to estimate size of proof
- Used to control proof number search in a cheapest-first manner
- Killall Go: prevent opponent from making any living group
- 9x9, solved some KillAll Go openings

Opening Books

and Task Managers for distributed search

- Recent representative work
- Wu et al, Game Solving with Online Fine-Tuning, Neurips 2023
- Uses Proof cost network
- Fine-tunes it online, during search
- Distributed game solvers
- Solved more 7x7 killall Go positions



Monte Carlo Game Solver (2020)

Cazenave, Monte Carlo Search workshop IJCAI

- Idea: very strong move ordering for “traditional” solver
- Uses online learning of playout policies …
- Uses Monte Carlo Tree Search …
- …both for move ordering for the main solver
- Strong results on many games
  - Atari Go, Nogo, Go, Breakthrough, …

Top Challenges in Solving Games

Near Term Challenges

- Solve 6x6 Go (working on it…)
- Framework for search-based solving of combinatorial games (working on it…)
- Alpha Zero-based MCTS solver (with regular network, or specialised proof cost net)
- Solve 6x5 and 6x6 Amazons (see 2015 paper for instructions…)
- Extensive comparison of alphabeta, PNS, df-pn, MCTS solver
  - Common codebase
  - Which algorithms for which (types of) games?

Top 10 Challenges in Solving Games

Long Term Challenges

- Incomplete information games: more, better (search-based) algorithms
- Better search algorithms
  - Combine benefits of popular approaches - PNS, MCTS solver
  - Better handle on proof cost estimates - more deep learning?
- Automated discovery of game-specific techniques: knowledge for solving, proofs for early static evaluation
- Cheap vs expensive knowledge - tradeoffs?
  - When are neural nets useful?
  - Can they be compiled into cheaper knowledge (such as NNUE)?
- Which games will be solved next?

Exact Solutions - Beyond Games?

- Verify correctness of safety-critical systems
- Guarantee correctness of a knowledge-based system
- Successful example: Chemical synthesis by retrosynthetic analysis (RA)
  - Formulate as an AND/OR search problem
  - Solve by game tree search methods: df-pn, MCTS
  - Heifets and Jurasica (AAAI 2012), Heifets (PhD 2014)
    - Heifets started a company, Atomwise, to develop these methods
    - “raised over $174 million of venture capital”
- Related: Kishimoto et al, Depth-First Proof-Number Search with Heuristic Edge Cost and Application to Chemical Synthesis Planning, Neurips 2019

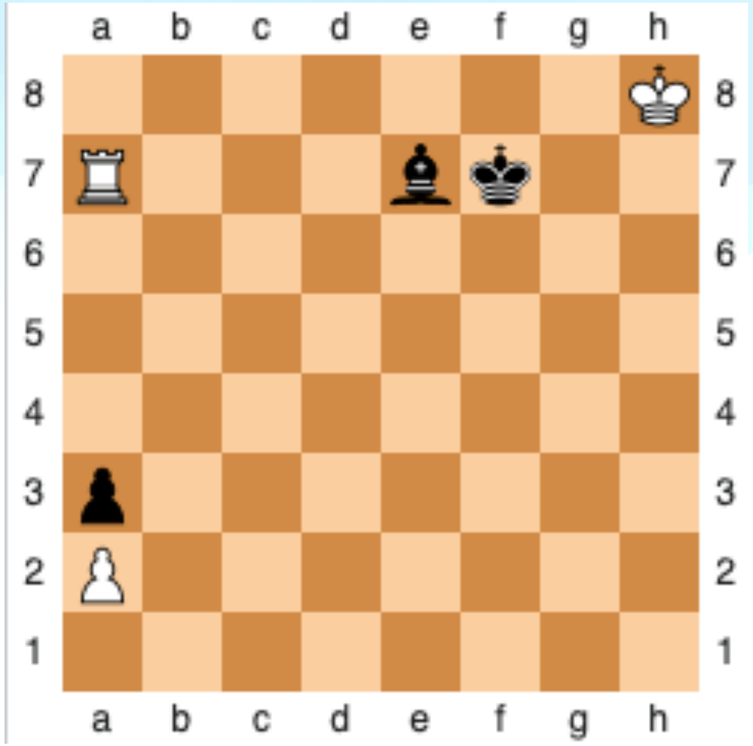
Summary and Conclusions

- Huge progress in last 20 years
- Stronger search algorithms and variants - df-pn, specialized algorithms for impartial and combinatorial games
- (Of course) much better hardware
- Much work on game-specific heuristics and improved knowledge of games
- Impact of deep learning just starting to be felt
- Another exciting 20 years ahead!

Extra Slides

Chess Endgame Tablebases

- https://en.wikipedia.org/wiki/Endgame_tablebase
- First computation, plus available for download
- Pioneers, 3,4,5 piece tablebases: Ströhlein, Thompson, …
- Some 6-piece: Stiller 1991, all 6-piece: Nalimov, syzygy (2005, 2013)
- 7-piece: Makhnychev and Zakharov (Lomonosov, 2012), syzygy 2018
- 8-piece: ongoing



Temperature Discovery Search

Müller, Enzenberger and Schaeffer, AAAI 2004
Zhang and Müller: TDS+: Improving Temperature Discovery Search, AAAI 2015

- The opposite approach
- Not using CGT to help games solving..
- Use game solving to **exactly compute a property of combinatorial games** - temperature
- Alphabeta search of game plus *coupon stack*
- From *principal variation* of search, can determine both the **mean** and **temperature** of game