Computing Science (CMPUT) 657

Algorithms for Combinatorial Games

Martin Müller

Department of Computing Science
University of Alberta
mmueller@ualberta.ca

Fall 2025

Last Class

- One slide summary of rest of Coupons and TDS
- Coursework update and reminder
- Map of CG algorithms
- Quick overview of “other things” in CG algorithms
- Research questions and ongoing work

Coupons and TDS - One slide summary

- Detailed slides are on website
- TDS: Play coupons C + game G - can determine both mean and temperature of G
- $V(G + C, p) = \mu(G) + V(C, p)$
- $V(C, p)$ known, $V(G + C, p)$ computed by search
- Compute mean $\mu(G) = V(G + C, p) - V(C, p)$
- Compute temperature $t(G)$ by finding the “last possible time” that play in G is optimal - may require multiple searches with different coupon stacks
- Exact if difference δ between coupons is “small enough”
- Pretty good approximations even for large δ

Coursework update and reminder

- Almost done with missing evaluations - try to mail them out tomorrow
- Also includes summary of all evals so far - please check
- Please re-read the project section in `https://webdocs.cs.ualberta.ca/~mmueller/courses/657-Fall2025/assessments.html#Project`
- Please follow the guidelines “A good report will contain...”

Map of CG Algorithms

- Huge space of possible algorithms
- Only a few case studies so far
- Different possible classifications
  - Exact solver vs heuristic player
  - Impartial vs partisan
  - Size of subgames - to what degree can we solve them?
  - Number of subgames, ease of decomposition

Some Other Topics We did not Discuss

- Kao, mean and temperature search
  - Direct search algorithm for means and temperatures
  - Assume we recognise and stop search in numbers
  - Re-implementation of the simple version in a 2022 course project
- Algorithms for loopy games:
  - Thermographs for Ko in Go
  - Locally, one player plays two or more moves to win a ko
  - Opponent gets the tax several times
  - Can compute means and temperatures for many but not all loopy Go positions

Future work - Use Reversible Moves

Reversible moves

- In CGT, part of conversion to canonical form
- Reduces the search depth, like “macro sequence”
- Can we use minimax search to recognise reversible cases
- In principle, yes, but it may be very expensive?
- Mostly, an open research problem

Future work - Use incentives with MCGS style boolean minimax search

- Incentive based pruning
- Game G, moves to L1 and L2
- Incentives L1-G, L2-G
- Prune if one incentive dominates the other
- Within the same subgame, can just search L1-L2
- With moves from different subgames Gi and Gj, must search (Gi-L1) - (Gj-L2)
- Is it feasible? Is there something simpler?

Future work - Real Go Endgames

- Go endgames - bridge the gap from puzzles to real endgames
- Make assumptions about independence, then verify
- Use strong engine (e.g. KataGo) for local search
- Evaluate a heuristic local game tree to get approximate temperatures
- Streamlined global search of such simplified local trees
- I believe such a system would be much stronger than Alpha Zero or KataGo in the endgame
- Lots of work, PhD-sized problem. But the path is relatively clear

Some Research Goals

- Understand strengths, weaknesses and limitations of each algorithm
- Combine algorithms
- Find the “boundary line” where to switch between algorithms
- Scaling to large problems, parallel search
- Integrate with ML, learned evaluation and move ordering

MCGS Research Goals

- Implement more of the existing algorithms within the same framework
- Apply to many different games
- Scale by number of subgames, subgame complexity, type of subgames
- Develop hybrid algorithms, e.g. boolean vs canonical form
- Approximation algorithms, between “full game” and “just temperature” (e.g. cooling)

Summary

- Introduction to CG algorithms
- "Just enough" of the math background
- Many good ideas, not a coherent framework at this time
- Lots of room for future research!