Computing Science (CMPUT) 657

Algorithms for Combinatorial Games

Martin Müller

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

Fall 2025

Approximation Algorithms for Combinatorial Games

Approximation Algorithms

- Goal of approximation algorithms: play sums well
- Use when exact methods are too slow
- Main approach: simplify subgames
- Biggest simplification: reduce information about game to single number = temperature
- Other possible approaches: make game tree smaller (e.g. cooling, selective search)
- If even computing temperature is too hard: approximate temperature

Mean, Temperature and Thermograph

- What are mean and temperature of a game?
- How to use *temperature* of games?
- Simple strategy: play in hottest subgame
- What are thermographs?
- Other sum game strategies: Sentestrat, Thermostrat - better in theory, not in practice
- Better in practice: combine global alphabeta search with local analysis based on temperature (Müller and Li 2004)
- We study temperature first,
  then come back to algorithms later

Two main questions of CGT

Two main questions about a game:

1. Who is ahead, and by how much?
   - Exact answer: value
     (a game in canonical form, often complicated)
   - New, approximate but simpler answer: **mean**
     (a rational number)
2. How big is the next move?
   - Exact answer: incentive
     (a game in canonical form, often complicated)
   - New, approximate but simpler answer: **temperature**
     (a rational number)

- **Thermograph** (TG): a data structure
  to compute mean and temperature *efficiently*

Review: Leftscore and Rightscore

- Minimax scores of a game, *assuming we stop playing at integers*
- Leftscore: Score if Left plays first
- Rightscore: Score if Right plays first
- Shorter notation:
  - LS(G) = Leftscore(G)
  - RS(G) = Rightscore(G)
- We can compute LS, RS for a sum by (global) minimax search
- Much faster: just compute LS, RS of a single subgame by (local) minimax
- The difference LS - RS tells us something about the value of moving first

LS and RS - Sums vs. Subgames

- $G = G_1 + ... + G_n$
- If each subgame is reasonably small:
- Can compute LS(G_i), RS(G_i) for each subgame (relatively) quickly
  - However, this is **not** enough information to compute LS(G) and RS(G)
- We can use LS(G_i) - RS(G_i) for approximating move value
- It is useful for understanding sums of hot games better, without adding them up
- Temperature is better than LS-RS

Example: LS and RS for Subgames vs Sum

- $G = G_1 + G_2 = \{2|0\} + \{5|4\}$
- LS(G_1) = 2, RS(G_1) = 0
- LS(G_2) = 5, RS(G_2) = 4
- What is LS(G) and RS(G)?
- Not 7 and 4!
- LS(G) = 6, RS(G) = 5 (from minimax play of G)
- 7 = LS(G_1) + LS(G_2) is a valid upper bound for LS(G)
- 4 = RS(G_1) + RS(G_2) is a valid lower bound for RS(G)
- Sometimes, such bounds are enough
- Bounds get weaker the more subgames we add

Mean

- First question:
  how good is a game G?
- Just G itself, without playing any moves
- Answer: Mean, written $m(G)$ or sometimes $\mu(G)$
- What **is** the mean?
- Different ways to define it, same result
- Intuition: "average expected result" of adding G to your sum
  - Example: play sum of many copies of the same game G
  - How much is it worth adding another copy of G?

Facts about Mean

- Fact: LS(G) $\geq$ m(G) $\geq$ RS(G)
- Notation: $nG = G + ... + G$ = sum of n copies of game G
- Theorem: Difference $LS(nG) - RS(nG)$ stays *bounded*
  - This is true no matter how large n gets
  - See proof of Theorem 5.17 in Siegel
- Consequence: First definition of mean
  - $m(G)$ is the limit of $LS(nG)/n$ ...
  - ...as n goes to infinity
- Second definition (later, much better in practice): use the *thermograph*

Examples for Mean

- Just examples - no proof here.
- $m(4) = 4$
- $m(4|-4) = 0$
- $m(6|-4) = 1$
- $m(4||-4|-10) = -3/2$
- $m(4||-4|-20) = -4$
- Sometimes we can "see" the mean by playing several copies of a game (do examples)

Main Property of Mean: Means Add

Theorem, e.g. (Siegel 3.26)

$$m(G + H) = m(G) + m(H)$$

- Intuition/Example:
- You have about 3 points advantage in G
- You have about 4 points advantage in H
- You have about 7 points advantage in G + H

Temperature

- Each game has a temperature - a rational number
- Idea: measure how much making a move is worth to the players
- Two ways to define it: by *tax*, or by using *coupons*
- We study tax first, coupons later

Temperature - Example

- Compare $G = 1|-1$ and $H = 10|-10$
- Clearly, playing in H is more valuable
- Whoever goes first gets 10 extra moves in H, only one in G
- We can use incentives to prove that: $H^L - H > G^L - G$
- However, incentives are games, can be complicated
- Temperatures are an easier way, “often” correct
- Idea: find out roughly how many extra moves does a move gain for the first player?
- Here, temperature of G is 1, but temperature of H is 10

Defining Temperature by Tax for Games

- Introduce a tax t on each move: $-t$ for Left, $+t$ for Right
- Game G changes to taxed game G_t
- At what tax rate t will players stop playing a game? That tax rate is the temperature of the game
- Principle:
  - Stop taxing at smallest t where G_t is “infinitesimally close” to a number (see next slide)
  - What does it mean?
  - Answer: LS(G_t) = RS(G_t)

Background: Infinitesimals, Infinitesimally Close

- What exactly is an infinitesimal?
- Answer (Siegel Proposition 4.3):
- Any game G for which LS(G) = RS(G) = 0
- Examples of infinitesimals:
  - $* = \{0|0\}$
  - 0
  - Any clobber position
  - Any nim value, e.g. *5
  - $\uparrow = \{0|*\} = \{0|\{0|0\}\}$
  - All tinies and minies: $+_n = 0||0| - n$, $-_n = n|0||0$
- Fact: if G is an infinitesimal, and $\epsilon > 0$ a number, then $-\epsilon < G < \epsilon$
- G and H are called *infinitesimally close* iff $G - H$ is an infinitesimal

Temperature Example

- Game $G = 1|0$
- Both players are eager to play in G
- New games G_t: pay a tax t on each move
  - $-t$ for Left
  - $+t$ for Right
  - stop when tax makes games “almost a number”
- $1 - t \,|\, 0 + t$
- At $t < 1/2$, LS(G_t) $>$ RS(G_t)
- At $t = 1/2$, the game becomes $1/2 \,|\, 1/2 = 1/2 + *$.
- LS(G_t) = RS(G_t) = 1/2
- Therefore, temperature t(G) = t($1| - 1$) = 1/2

Another Temperature Example

- Game $G = 10|-2$
- Temperature t(10$|-2$) = 6.
- Let's work out why.

Temperature - Discussion

- Temperature t(G): measures urgency of move in G
- Higher temperature *usually* means more urgent to play
- **Not** a strict ordering
  - In some sumgames, playing in a lower temperature subgame is better
  - Why? We'll discuss details later, but it has to do with getting more good moves than the opponent
- Temperature is *very often* a good measure of urgency
- In some sense, it is the *best possible* single number for ordering moves by urgency

More Examples for Temperature

- Temperature t(4) = -1 (Why?? See next slide)
- t(4|-4) = 4
- t(4$|\{-4|-10\}$) = 11/2
- t(4$|\{-4|-20\}$) = 8
- t(4$|\{-4|-100\}$) = 8
- (work out some cases in detail, using tax)

Temperature of an Integer

- Temperature t(4) = -1
- Why??
- $G = 4 = \{3|\,\}$
- Left loses one point from playing from 4 to 3
- With a positive tax, Left would lose even more
  - Example: At t = 1/2
  - Left option in G_t to 3 - 1/2 = 2.5 (worse than 3)
- Left loses something even with a *negative tax*
  - Example: t = -1/2
  - Left option to 3 - (-1/2) = 3.5, still worse than the original game 4
- Finally, with a negative tax of t = -1, Left becomes indifferent, no loss from playing
  - Example: t = -1, Left option to 3 - (-1) = 4

Temperature of Sums

- Theorem (Siegel 5.18): t(a+b) $\leq$ max(t(a), t(b))
- Example: a=4 | -4, b=5 | -5, c= $5 \,||\, -4 \,|\, -6$
- Temperatures of single games:
  t(a) = 4
  t(b) = 5
  t(c) = 5
- Temperature of sum of two games:
  t(a + b) = 5
  t(b + c) = 1
  t(b + b) = 0

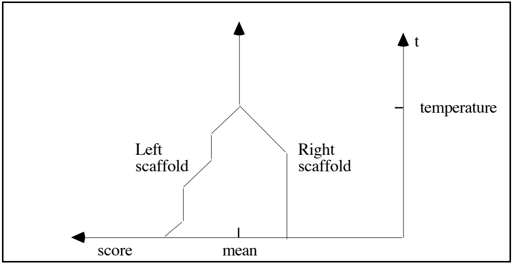
Switch (Review + Mean + Temperature)

- Game of the form a|b, $a \geq b$
- Each player has exactly one move
- Then game becomes a number
- $m(a|b) = (a + b)/2$
- $t(a|b) = (a - b)/2$
- Example: $G = 10| - 3$
- $m(G) = 7/2$
- $t(G) = 13/2$
- Note: $G + G = 7$

How to Compute Means and Temperatures?

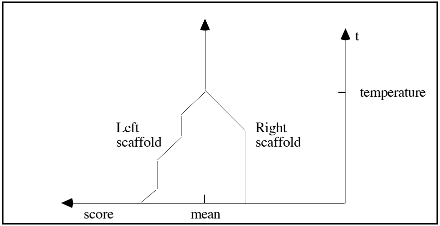
- Given a (maybe complicated) game G
- How to compute its mean and temperature?
- Trying out many tax rates is not practical
- We would like:
    - A recursive rule for $G = \{G^{\mathcal{L}} | G^{\mathcal{R}}\}$
    - Given: means, temperatures of all Left and Right options $G^{\mathcal{L}}$, $G^{\mathcal{R}}$
    - Compute the mean and temperature of G
    - **Problem**: this is *impossible* in general
    - Not enough information...
- Solution: compute *thermographs* instead
    - They do contain enough information...

Thermograph



- Temperature t on the y-axis
- Thermograph = two functions $LS(t)$ and $RS(t)$
  - They are the LS and RS of G_t
- Note: the graph is not shown like your usual function graph. t is the variable, and Left is positive

Thermograph (TG) Components



- Consists of left and right *scaffold*, $LS(t)$ and $RS(t)$
- Scaffolds first meet at value $m(G)$ and at $t = t(G)$, the mean and temperature of game G
- Left and right scaffolds = *mast* for all $t \geq t(G)$
- The mast starts at coordinate $(m(G), t(G))$

Thermograph - Motivation

- Big improvement in terms of representation:
  - A game (and its canonical form) may be very complex
  - Temperature and mean are only two numbers, not enough information
  - Thermograph has a “just right” amount of information
- Thermograph: describes characteristics of a game at an intermediate level of complexity
- Idea: simplify game by tax t on each move, plot $LS(t)$ and $RS(t)$ until they coincide
- Stop when the taxed game is “close” to a number

Tax and Thermograph Fundamentals

- The lowest tax rate is -1
- Negative tax means you actually get paid for making a move!
- All thermographs start at $t = -1$
- Thermographs of integers are vertical *masts*, starting at $t = -1$ and infinitely high
- Thermographs of all other games are constructed recursively from the thermographs of its options
- Remark: all you really need is the thermograph of 0. You can construct TG for all other integers from the (one) option using the *simplest number rule* (see later slide)

Thermograph Base Case: Integers

- Thermograph of integer n is a vertical *mast*: $LS(t) = RS(t) = n$ for all $t \geq -1$
- Used to construct other games

Constructing a Thermograph

- Example: `http://senseis.xmp.net/?Thermograph`
- Recursive construction:
- Base case: TG of integers are masts
- TG constructed from TG of all left and right options
- Tax right scaffold of each left option by t
  - Tilt to right: diagonal-left lines become vertical, vertical become diagonal-right
- Tax left scaffold of each right option by $-t$
  - Tilt to left: diagonal-right lines become vertical, vertical become diagonal-left

Constructing a Thermograph, Part 2

- Example: `http://senseis.xmp.net/?Thermograph`
- Compute a *single* scaffold for each player
- How? Take the max (min) over all left (right) options
- Find the intersection of the taxed left, right scaffolds
- Cut off scaffolds there, replace by vertical mast
- Let's do some examples

Sub-Zero Thermographs

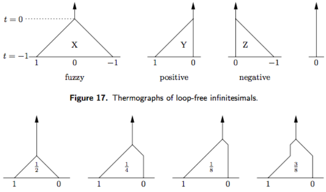
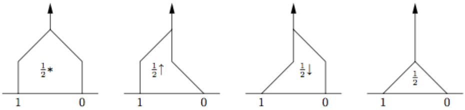


Image source: (Berlekamp 1996)

- In older texts, thermographs are defined only for $t \geq 0$.
- For computation, it is better to always start from $t = -1$
- “Sub-zero thermography” (Berlekamp 1996, Section 2.5)
- Numbers are exactly the games that have a temperature below 0
- Integers are exactly the games that have temperature -1
- My C++ implementation of sub-zero thermography is available for this course

Sub-Zero Thermograph Examples



- "Number-ish" games:
  - number plus non-zero infinitesimal
  - temperature 0
  - non-straight thermograph below 0

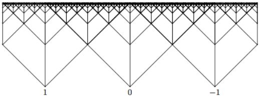


Figure 20. The foundations of number-ish thermographs.

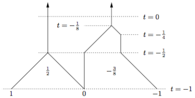


Figure 21. Underground thermographs of $\frac{1}{2}$ and $-\frac{3}{8}$.

Image source: (Berlekamp 1996)

Problems of Old-style (start from zero) Thermography

- Remember the definition of temperature by tax
- We magically needed to stop taxing when the game became infinitesimally close to a number
- How do we know when that happens?
- In practice, determining this is only feasible for simple games, and with much manual case-by-case analysis
- This is only practical for small games that are already in canonical form
- But we do not want to compute canonical forms all the time!
- So how can we compute thermographs in general, bottom-up, for any game (number or other)?

Advantage of Subzero Thermography

- With subzero thermographs all the problems go away
- We can construct everything bottom-up in a uniform way
- Numbers are dealt with automatically and correctly at some $t < 0$
- Example: $5/8 = \{1/2 \mid 3/4\}$ (do on whiteboard)

- But wait - there is still one thing missing
- In fact it was even missing for “classical” thermography

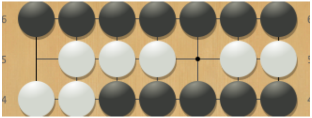
Thermographs and Simplest Number Rule

- We defined thermograph masts using the *intersection* point of taxed scaffolds
- What if they *never* intersect?
- Example: zugzwang games, e.g. $\{-5 \mid 3\}$ (do example on whiteboard)
- Solution: use the *simplest number rule*
- If the taxed scaffolds of G^L, G^R do not intersect, then:
  - G = simplest number between $RS(G^L)$ and $LS(G^R)$
  - Simplest = smallest birthday (it is unique)
  - The thermograph of G is the thermograph of this number
- Example: $G = \{-5 \mid 3\}$
  - $G = 0$, thermograph is mast at 0

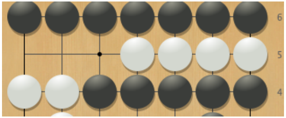
Sente and Gote

- Sente and Gote are terms from the game of Go
- Sente: the initiative
  - To *keep sente*: the opponent must answer our move, or loses something
- Gote: losing the initiative, opposite of sente
  - Playing a move that the opponent can ignore

Gote and Sente Examples in Go



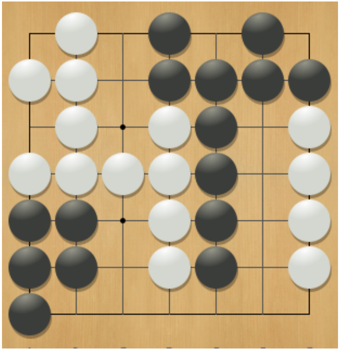
Note: all stones except the two white ones on the right are assumed safe here



Note: all stones except the four white ones on the right are assumed safe here

- Gote move for both players
  - G = 4|0
  - mean 2, temperature 2
  - LeftScore 4
  - RightScore 0
- Sente move for White, Gote for Black
  - G = 9||8|0
  - mean 8, temperature 1
  - LeftScore 9
  - RightScore 8
  - Move to 8|0 raises temperature to 4
  - Usually, Black will answer, move to 8

Double Sente Example



- Double sente
- G = 14|2 ||| -2||-14|-16
- mean -1/4, temperature 8 1/4
- LeftScore 2
- RightScore -2
- both sides have a big threat
- If opponent answers: free profit (2 vs -2)

Examples - Sente and Gote Thermographs

- Three examples:
  - Gote
  - One-sided sente
  - Double sente
- All examples will have LS = 4, RS = 0.
- They all appear the same to a local minimax search
- But they are very different games when played in a sum
- They have very different TG
- Let's draw them for practice

Gote

- Temperature drops after move (or move sequence)
- Example: `4 | 0`
- LS = 4, RS = 0
- mean = 2, temperature = 2

One-sided Sente

- One side has big threat, can *usually* force the opponent to answer
- Game: `22 | 4 | | 0`
- LS = 4, RS = 0
- mean = 4, temperature = 4
- *Reverse sente*: take away opponent's sente move
  - Right's move to 0 in the example is a reverse sente

Double Sente

- Game: 13|4 || 0|-11
- LS = 4, RS = 0
- Mean = 3/2, Temperature = 7
- With large threats, temperature can become arbitrarily high: `G = {{100002|4} | {0|-100000}}`
- Mean = 3/2, Temperature = 100003/2

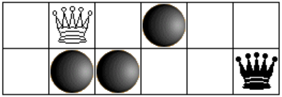
Why Temperature is Important

- Assume we want to play by local minimax searches only
- Compute LS(G), RS(G) for each subgame G
- Two minimax searches
- Then, play the subgame where the difference LS(G) - RS(G) is largest
- We call this *greedy* play
- It only considers each subgame separately, then chooses greedily
- It ignores the fact that we're playing a sum
- If we *keep sente*, we also get the next-biggest move

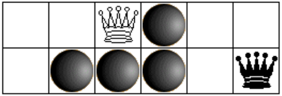
Why Greedy Play can be Bad

- Assume $G = G_1 + G_2$, Left to play
  $G_1 = 4|0$,
  $G_2 = 10|-3||-5$
- Minimax G_1 only: LS = 4, RS = 0, difference = 4
- Minimax G_2 only: LS = -3, RS = -5, difference = 2
- A "locally greedy" left player would play in G_1 first, and let right play in G_2.
  - Final score of playing $G_1 + G_2$ this way:
  - $4 - 5 = -1$
- Globally optimal play:
  - take the sente move in G_2 first, right should answer
  - Left still gets to play first in G_1
  - final score 4-3 = **+1**

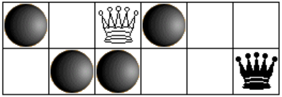
Why even Locally Greedy Play can be Bad



G



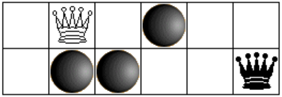
$G^{R1} = 0$



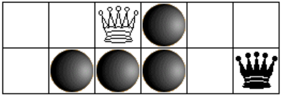
$G^{R2} = 1 || -2 | -4$

- $G = \{2 ||| 0, 1 || -2 | -4\}$,
- Black's only move is to 4-2=2
- White has two good moves
  - Move to $G^{R1} = 0$, so $LS(G^{R1}) = 0$
  - Move to $G^{R2} = 1 || -2 | -4$, $LS(G^{R2}) = 1$
- $RS(G) = min(0, 1) = 0$

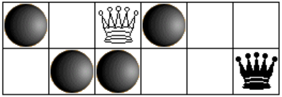
Why even Locally Greedy Play can be Bad



G



$G^{R1} = 0$



$G^{R2} = 1||-2|-4$

- The greedy move to 0 is only good at low temperatures
- With other games around, this move loses “on average” one point:
  - $m(0) = 0$
  - $m(\{1||-2|-4\}) = -1$
- At higher t, better to keep the option to invade open
- At low t, go for cash locally

Summary

- Mean and temperature approximately describe
  - The value of a (sub)game (mean)
  - The urgency of moving in it (temperature)
- Thermographs contain enough information to compute mean and temperature recursively from the options
- Can be computed **much** more efficiently than canonical form
  - size stays bounded in practice (rare to have more than 10 corner points in a graph)
- Very useful tool, especially with subzero thermographs
- Will be used in several algorithms
- Approximation algorithms also exist (TDS, TDS+ exist in both exact and approximate versions)