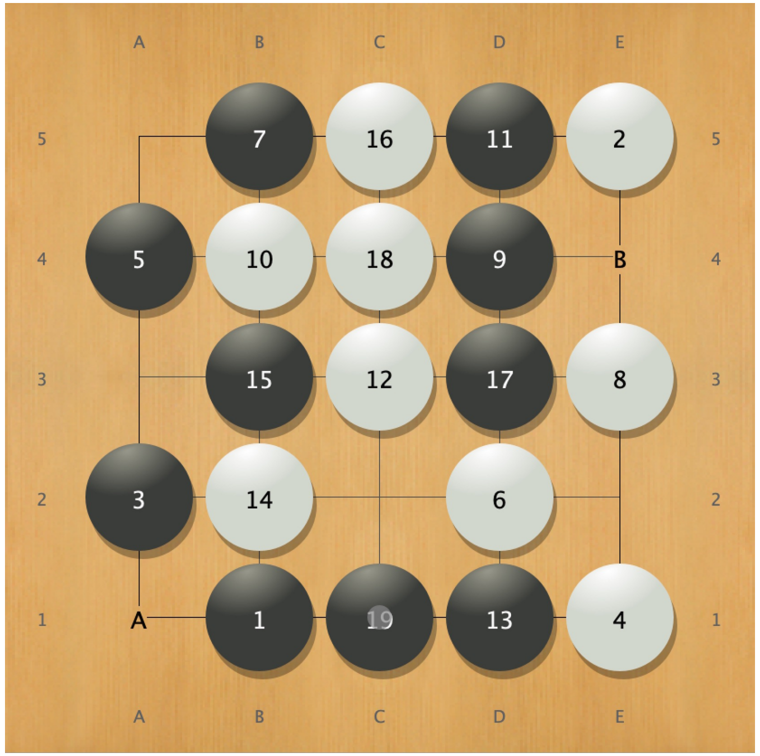
Solving NoGo

Henry Du

NoGo



- Blocks
- Liberties

- All blocks must have at least one liberty
- Capturing and suicide are not allowed

- Game ends when a player has no move to make

Publications

- Haoyu Du, Ting-Han Wei, and Martin Müller.
- Solving NoGo on Small Rectangular Boards.
- In *Advances in Computer Games*, 2023. → SBHSolver

- Haoyu Du and Martin Müller.
- Solving Linear NoGo with Combinatorial Game Theory.
- In *Computers and Games*, 2024. → CGTSolver

SBHSolver

Contributions

- Weakly solving NoGo

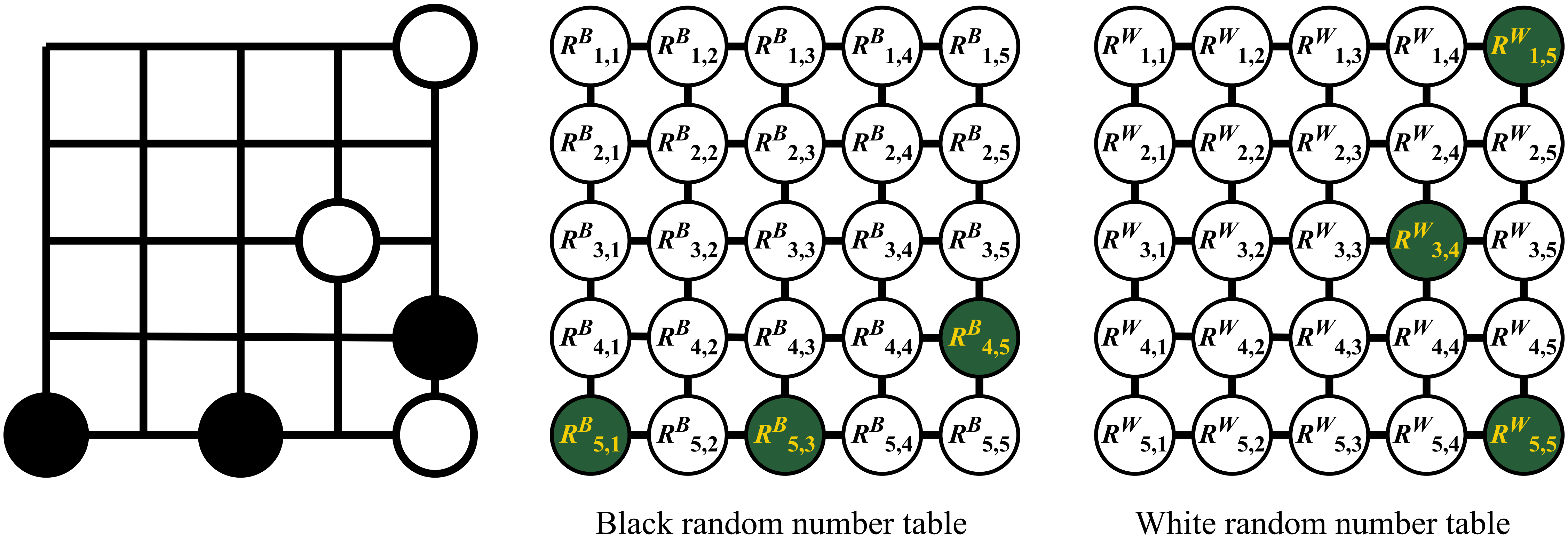
- Sorted Bucket Hash (SBH)
- Solutions of NoGo on boards with up to 27 points
- Statistics and human-understandable strategies of NoGo

Motivations

- Weakly solving – outcome and an explicit strategy
- The strategy can be used for game playing.

- Zobrist hashing has large memory footprint.
- Zobrist hashing may produce too many hash collisions.

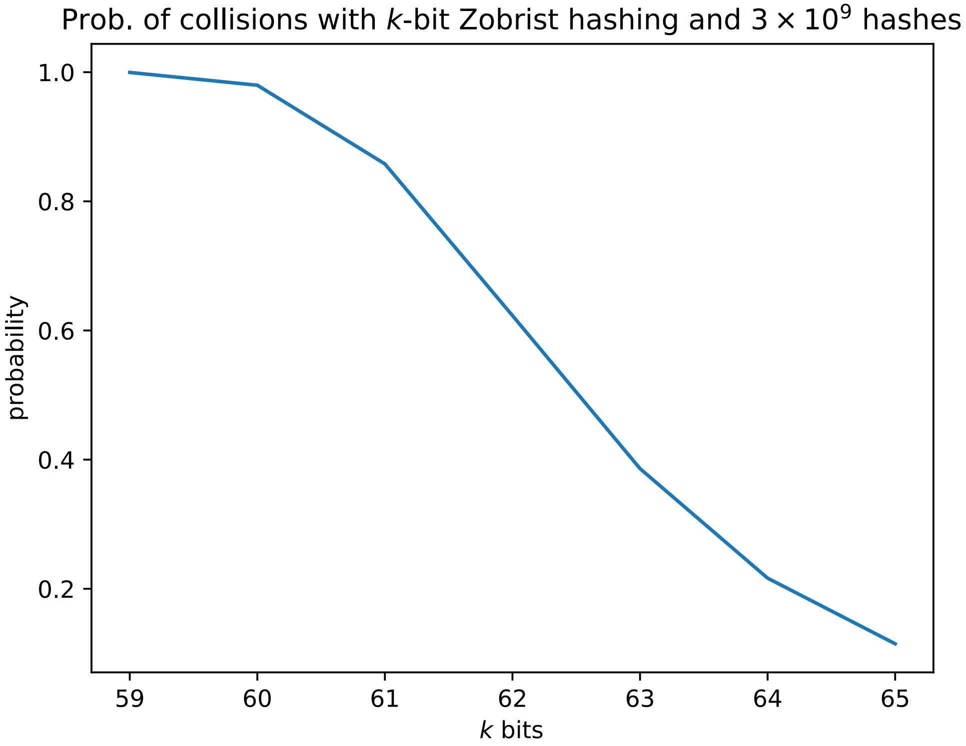
Zobrist Hashing

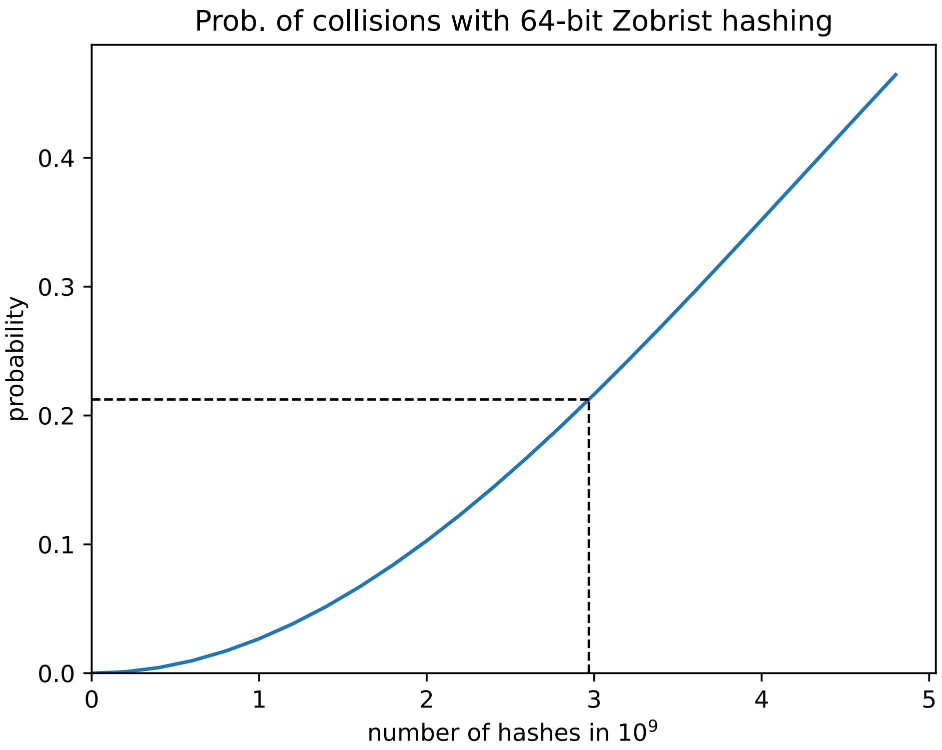


$$h(\text{board}) = R^W_{1,5} \oplus R^W_{3,4} \oplus R^B_{4,5} \oplus R^B_{5,1} \oplus R^B_{5,3} \oplus R^W_{5,5}$$

Prob. of Collisions

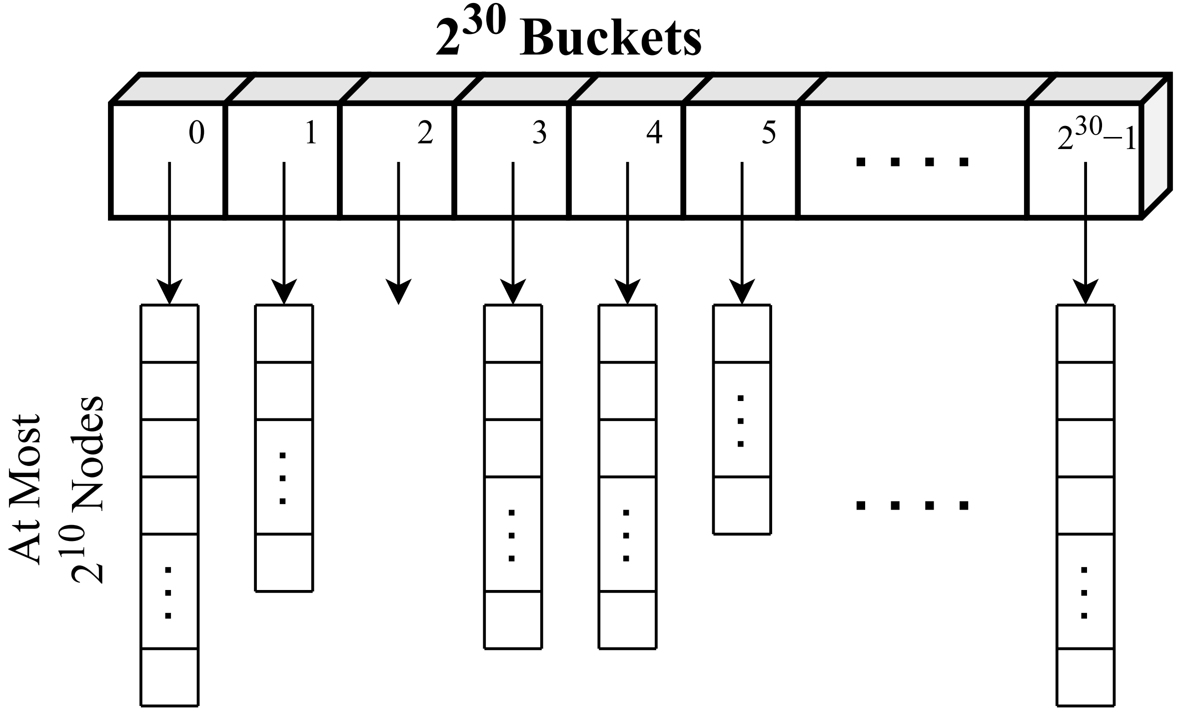
- Birthday problem



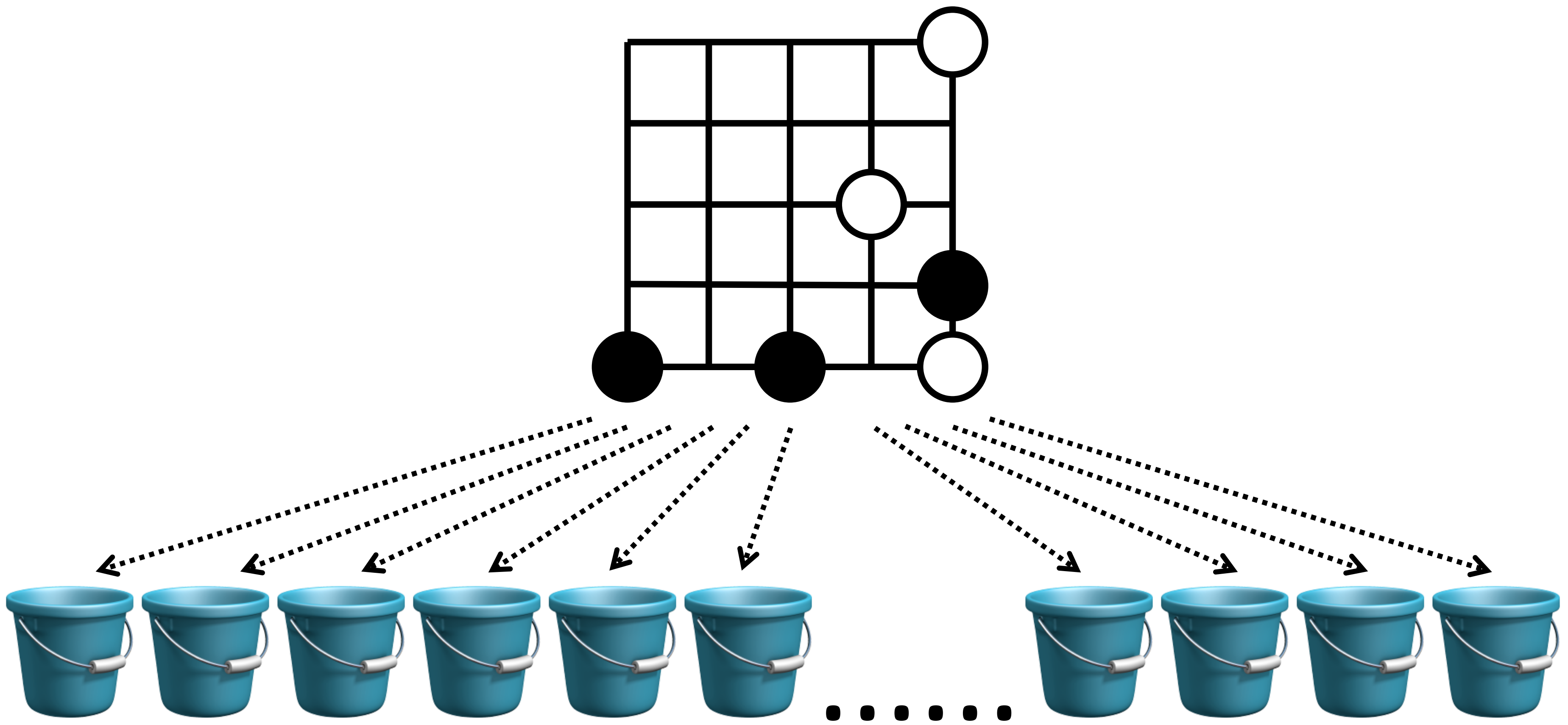


Sorted Bucket Hash (SBH)

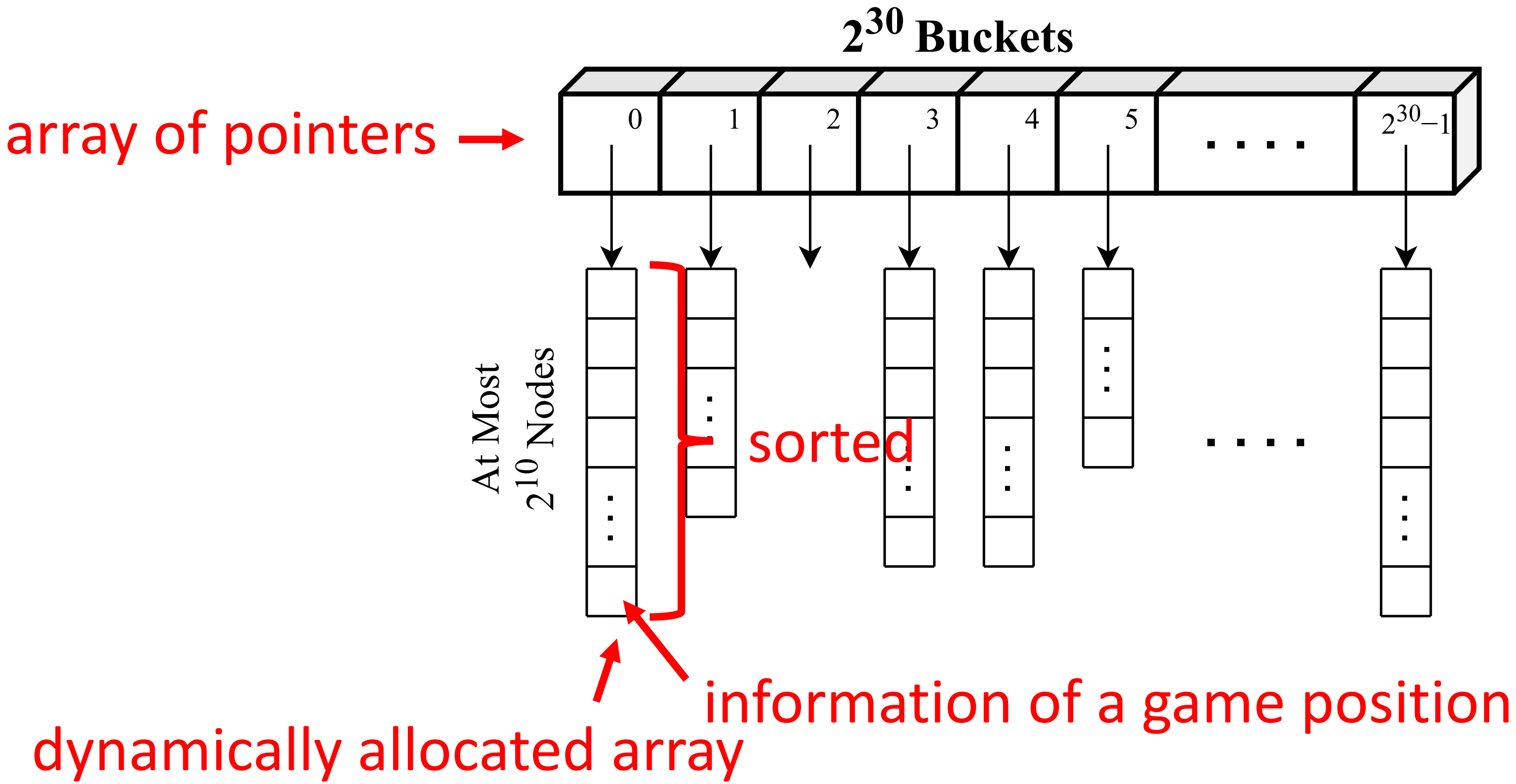
- Weakly solving games
- Perfect hashing
- Memory footprint (32GB)



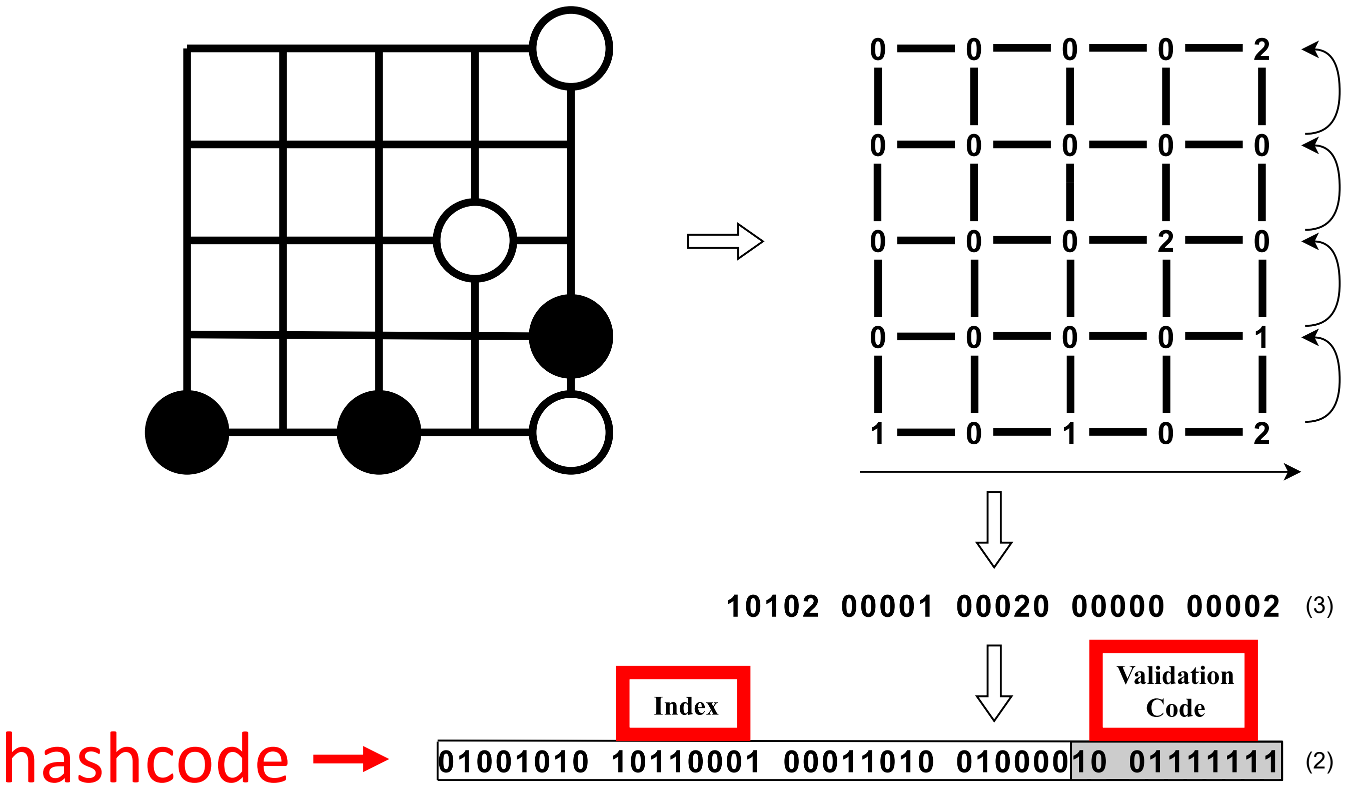
SBH – Intuition



SBH – Data Structure

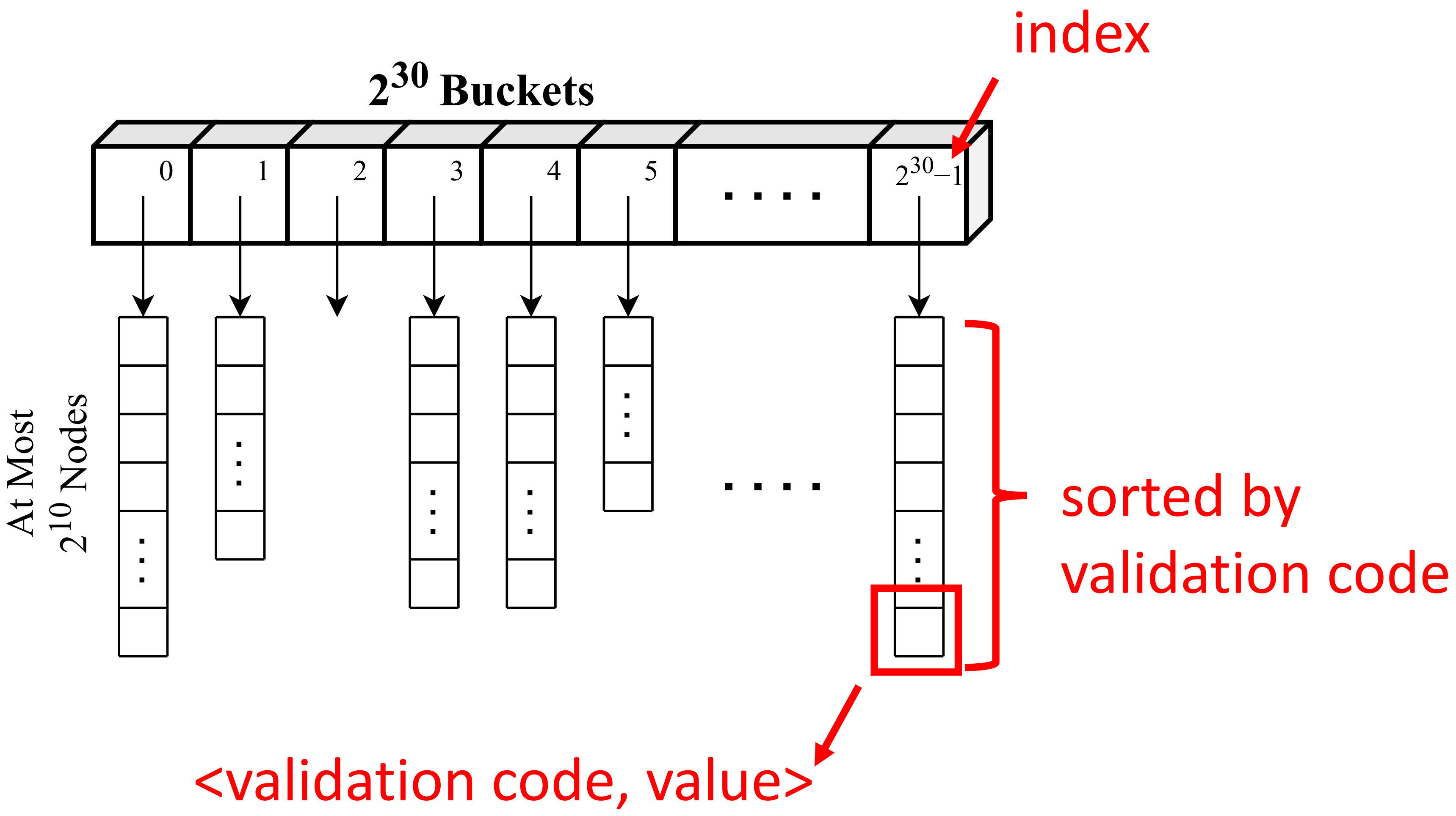


SBH – Calculating Hashcode



k-bit hashcode
m-bit index
n-bit validation
$k = m + n$

SBH – Transposition Table



SBHSolver

- Boolean Negamax

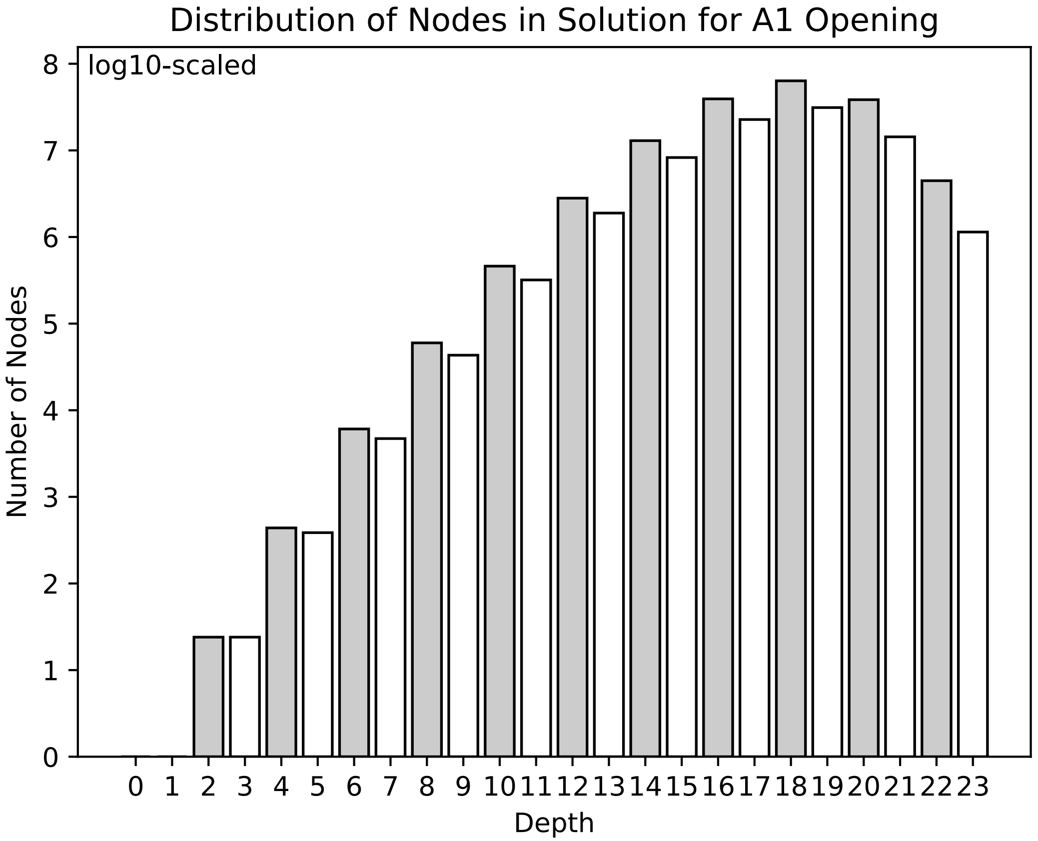
- History heuristic
- Enhanced transposition cutoff
- SBH transposition table

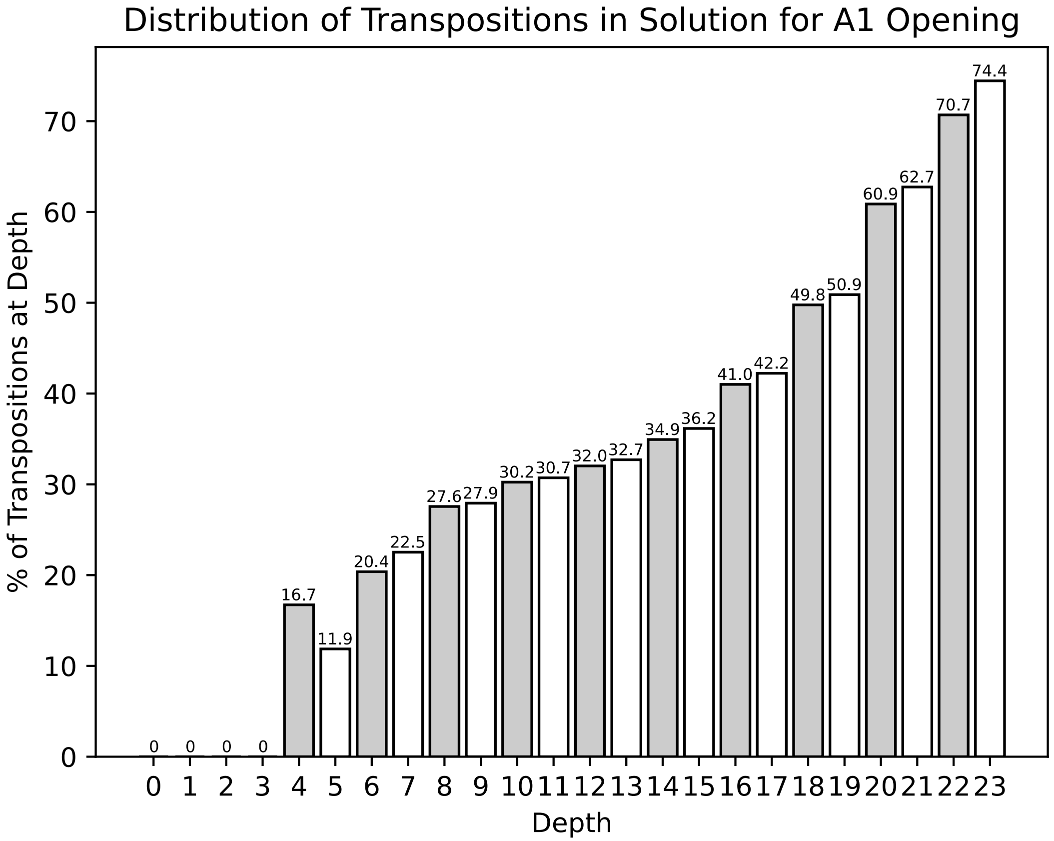
SBHSolver – Efficiency Gains

- 5x5 NoGo

- She, 2013 — 520×10^9 points
- Cazenave, 2020 — 46×10^9 moves
- **SBHSolver** — 3×10^9 game positions

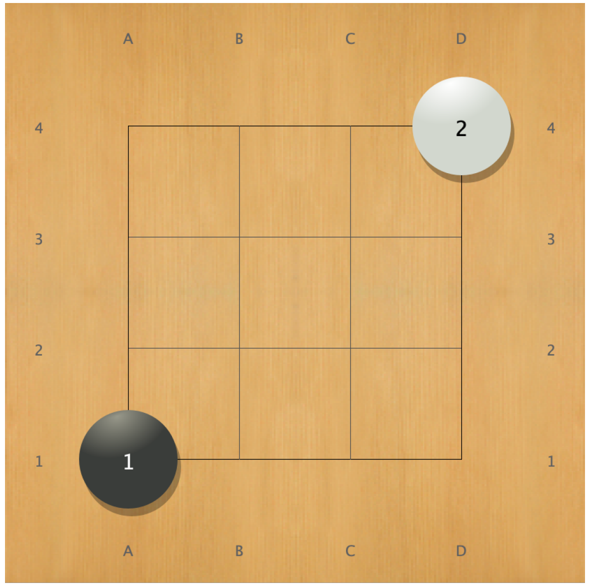
SBHSolver – Efficiency Gains

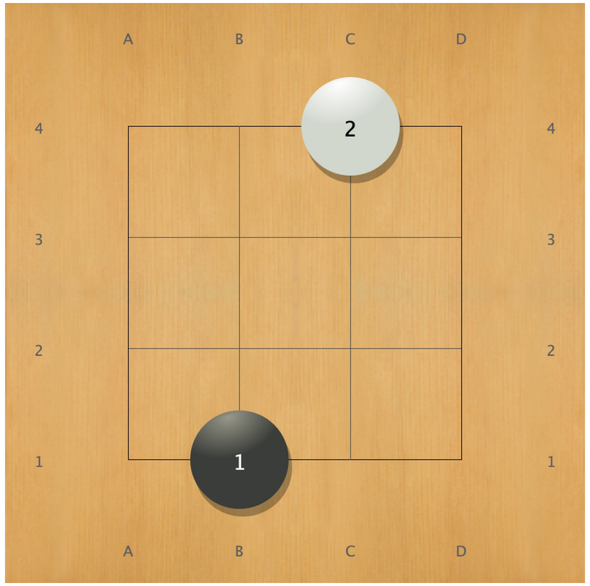


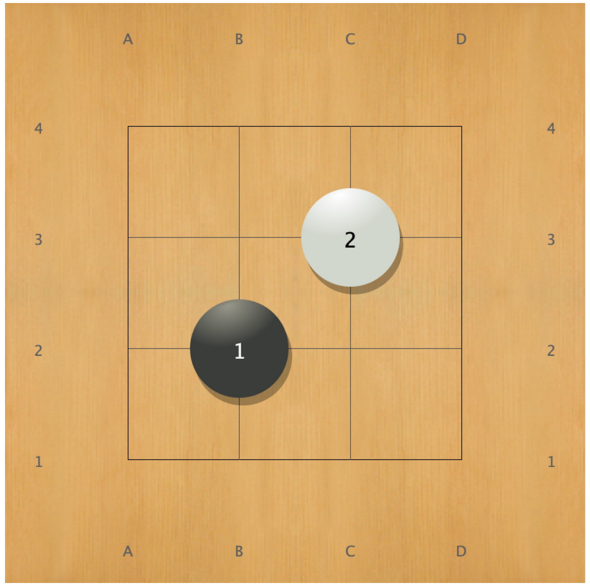


4x4 NoGo

- White wins
- Strategy: playing symmetrically
- It is the winning move for 85.3%

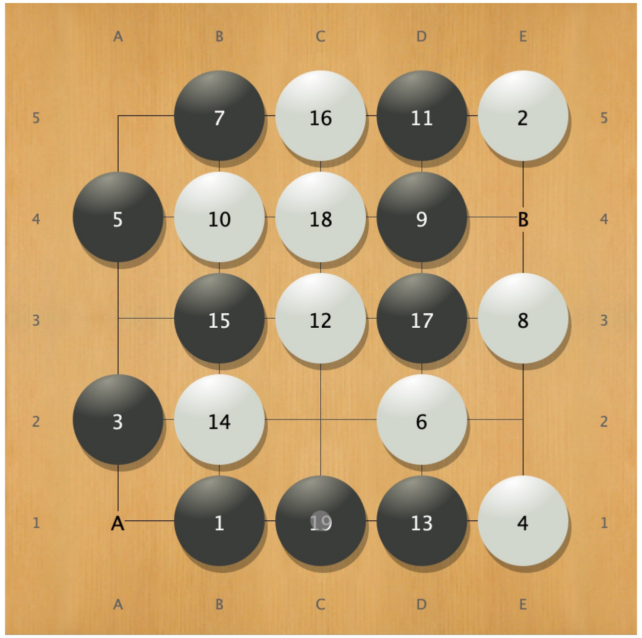






5x5 NoGo

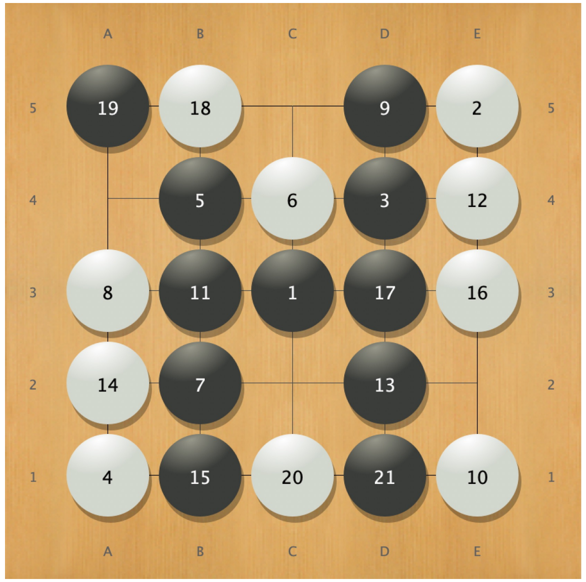
- Black wins
- Strategy: making eyes



- In solution of A1 opening,
- 45.2% of the endgame positions contain at least one eye

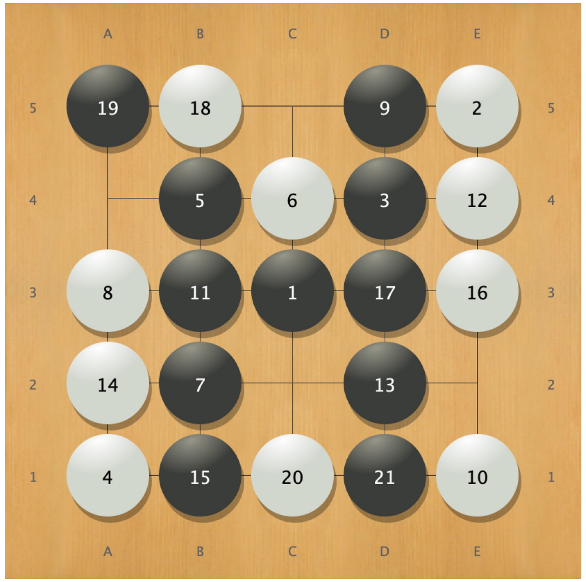
5x5 NoGo

- Black wins
- Another strategy: long strings to separate the opponent's stones



5x5 NoGo

- Black can win in 21 moves

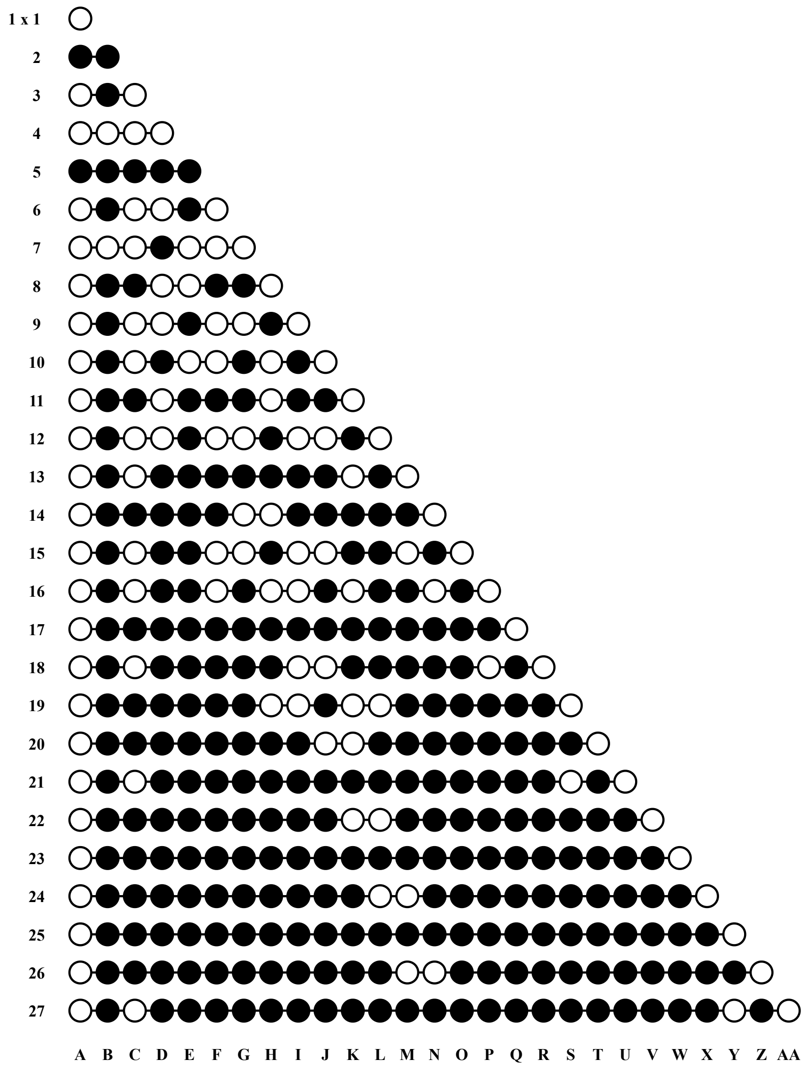


Solutions on Rectangular Boards

- Up to 27 points

Row \ Column	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	1	1	0	1	1	1	1	1	1	1	1	1
2		1	0	0	1	1	1	1	0	0	1	1	1
3			1	0	1	1	1	1	1				
4				0	1	1							
5					1								

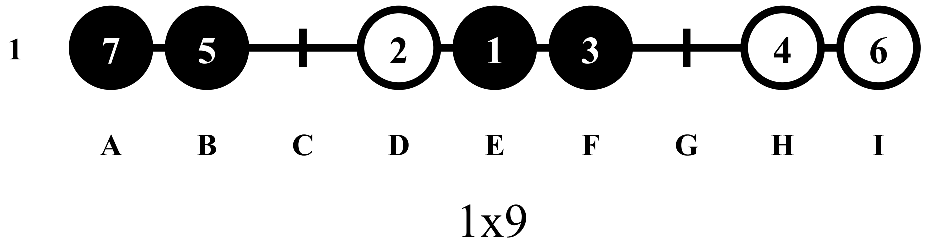
Solutions on Rectangular Boards

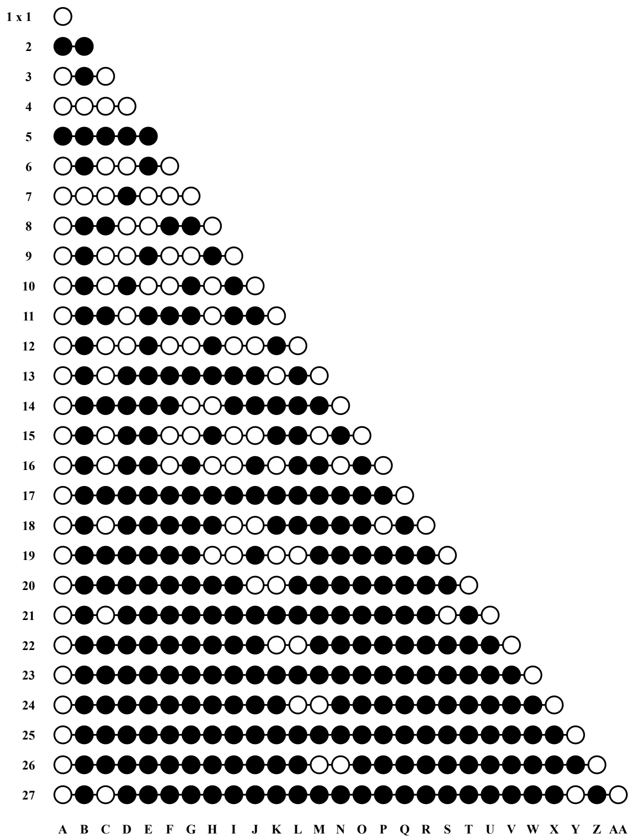


Solutions on Rectangular Boards

Theorem 1. In $1{\times}n$ NoGo, with odd $n > 1$, the opening move at the center point $(n+1)/2$ wins.

Proof sketch. Black wins by symmetric play.

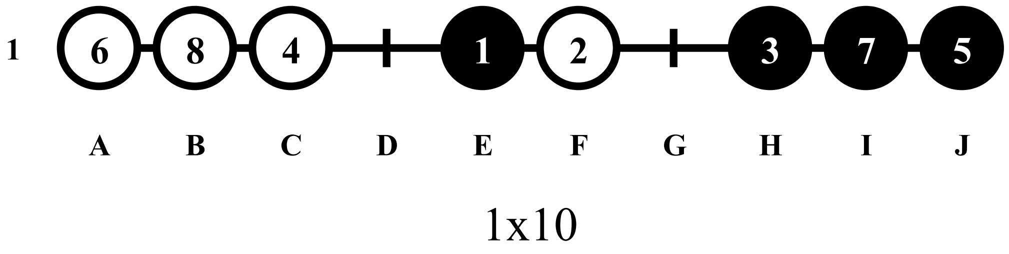


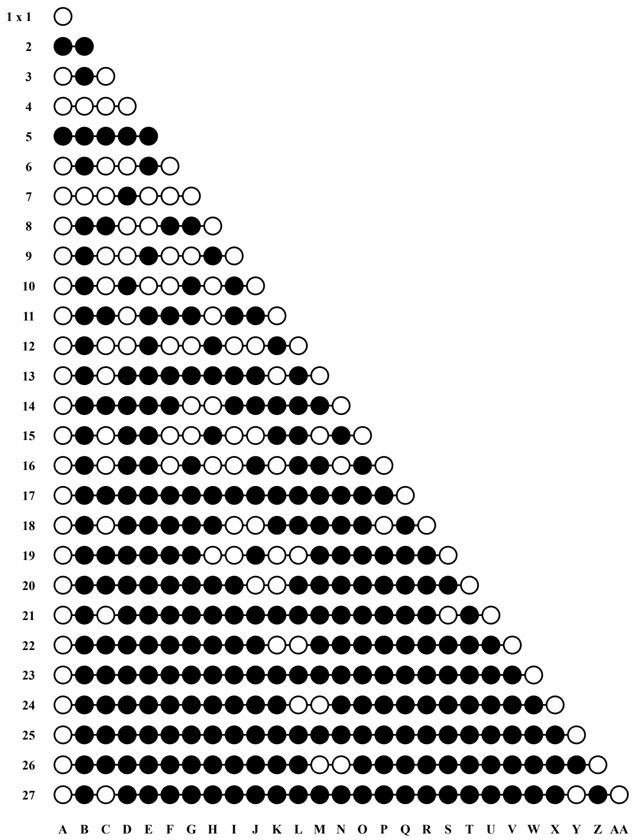


Solutions on Rectangular Boards

Theorem 2. In $1{\times}n$ NoGo, with even $n > 2$, the opening moves at the two middle points $n/2$ and $n/2 + 1$ lose.

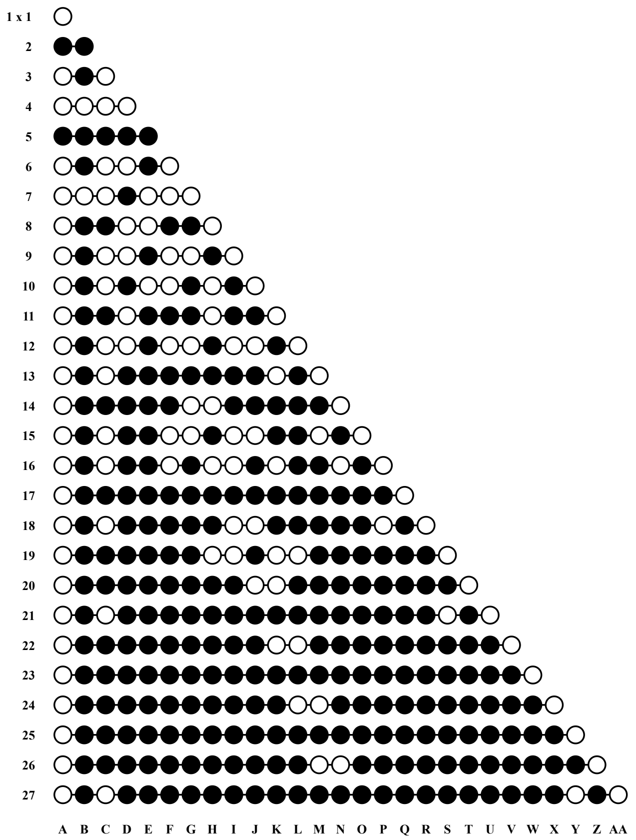
Proof sketch. White wins by symmetric play.





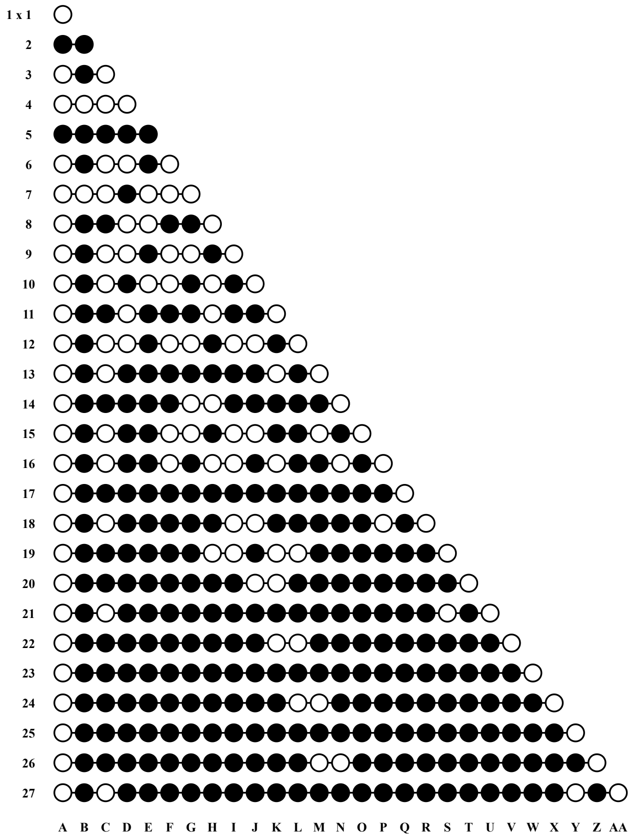
Solutions on Rectangular Boards

Conjecture 1. For $n > 5$, the opening moves at the two ends of the board are losing moves.



Solutions on Rectangular Boards

Conjecture 2. For $n > 7$, the opening moves at the points next to the two ends of the board are winning moves.



Conclusions on SBHSolver

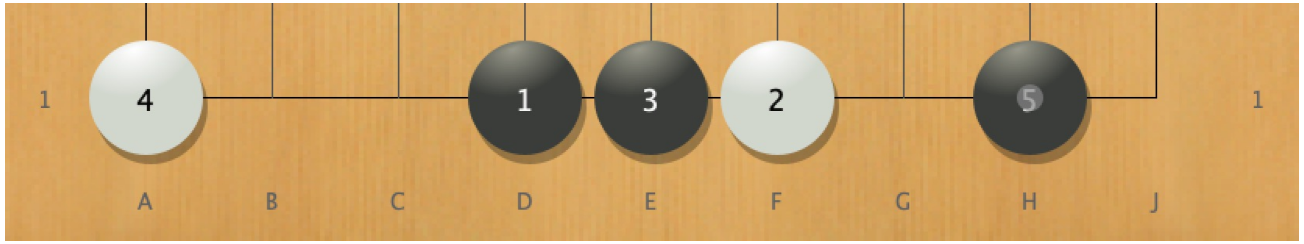
- Sorted Bucket Hash
- SBHSolver
- NoGo solutions up to 27 points
- Strategies on 4x4, 5x5, and general NoGo games
- Two theorems and two conjectures on Linear NoGo

CGTSolver

Linear NoGo

- NoGo played on $1 \times n$ boards
- `x` – black
- `o` – white
- `.` – empty

`o..xxo.x.`



CGT

- Outcome class

		Left playes next	
		wins	loses
Right playes next	wins	$\mathcal{N}$	$\mathcal{R}$
	loses	$\mathcal{L}$	$\mathcal{P}$

Black is *Left* player
White is *Right* player

CGT

- Equality

$G = H$ iff

$oc(G + X) = oc(H + X)$, for all X

CGT

- Inverse

$$G = \texttt{.x..o.x}$$
$$-G = \texttt{.o..x.o}$$

- $oc(G + -G) = \text{P}$
- G + −G = 0

A New Solver for Linear NoGo

- CGTSolver: Boolean Negamax + CGT

- Block Simplification & xo-Split

- Static evaluation
- Pre-computed database
- Play-In-The-Middle (PITM) heuristic
- Transposition table

Block Simplification

Theorem 1. A block of stones of the same color can be replaced by a single stone of that color.

`o..xxxo.xx. = o..xo.x.`

Proof sketch. The set of legal moves does not change.

xo-Split

Theorem 2. A Linear NoGo game can be split into two independent subgames at the boundary between two blocks of opposite colors.

```
o..xxxo.xx. = o..xxx + o.xx.
```

Proof sketch. A move played on the left does not affect the liberties, and the set of legal moves, on the right.

Cancellation of Subgames

`..x.oox..xo...xoo..oxxx.`

block simplification	`= ..x.ox..xo...xo..ox.`
xo-split	`= ..x.o + x..x + o...x + o..o + x.`
cancellation	`= ..x.o + o...x + x.`

CGTSolver

- Static evaluation
- Pre-computed database
- Play-In-The-Middle (PITM) heuristic
- Transposition table

Static Evaluation

$$\underbrace{\texttt{x..o.xo..ox.}}_{G} = \underbrace{\texttt{x..o.x}}_{H_1} + \underbrace{\texttt{o..o}}_{H_2} + \underbrace{\texttt{x.}}_{H_3}$$

OutcomeClass(H_1) = N   Next player wins

OutcomeClass(H_2) = R   White wins

OutcomeClass(H_3) = P   Previous player wins

N + R + P = N + R

If White goes next, White wins.

Reduced Position

Definition 1. A Linear NoGo position is called *reduced* if neither block simplification nor xo-split can be applied.

Pre-computed Database

Theorem 3. For all $n > 0$, there are 3^{n+1} distinct *reduced* positions with n empty points.

Proof sketch. A reduced position does not contain adjacent stones of the same color (block simplification) or opposite color (xo-split).

Reduced positions with 3 empty points: `_._._._`

```
...          .x..        o.x..       x.o.x.     o.o.o.x
```

Pre-computed Database

- Organized by layers from $n = 1$ to 15
- All reduced positions with up to 15 empty points
- $\sum_{n=1}^{15} 3^{n+1} = 64{,}570{,}077$ positions/entries
- <board, outcome class, pointer to *simplest equal game*>

Simplest Equal Games

Given a Linear NoGo position G, its simplest equal game $s(G)$ is defined to be the game that

(1) is equal to G, and

(2) appears earliest in the database.

`..o.x.o..x = ..`

G $s(G)$

- During search, replace G by $s(G)$.

Finding Simplest Equal Games

To find the simplest equal game of a position G,
for each game H_i before G in the database,
we sequentially test by search if $G - H_i = 0$.

PITM Heuristic

- Play-In-The-Middle
- Try moves closer to the middle of a subgame first.

- Move ordering:
- From largest to the smallest in length, using PITM in each subgame.

- Quickly break down a large game into smaller subgames.
- Increase the chance of database hits.

Transposition Table

- A node is $G_1 + G_2 + \cdots$
- Encode the board of each subgame G_i as a string
- Sort board strings
- Concatenate sorted strings with separator symbols in between

$s =$ . . | x . . . o | ...

- Zobrist hashing
- $R^B, R^W, R^E, R^{|}$
- $h(s) = \bigoplus_i R_i^{s_i}$

Black	$R^B{}_1$	$R^B{}_2$	$R^B{}_3$	**$R^B{}_4$**	$R^B{}_5$	$R^B{}_6$	$R^B{}_7$	$R^B{}_8$	$R^B{}_9$	...
White	$R^W{}_1$	$R^W{}_2$	$R^W{}_3$	$R^W{}_4$	$R^W{}_5$	$R^W{}_6$	$R^W{}_7$	**$R^W{}_8$**	$R^W{}_9$	...
Empty	**$R^E{}_1$**	**$R^E{}_2$**	$R^E{}_3$	$R^E{}_4$	**$R^E{}_5$**	**$R^E{}_6$**	**$R^E{}_7$**	$R^E{}_8$	$R^E{}_9$	...
Seperator	$R^{|}{}_1$	$R^{|}{}_2$	**$R^{|}{}_3$**	$R^{|}{}_4$	$R^{|}{}_5$	$R^{|}{}_6$	$R^{|}{}_7$	$R^{|}{}_8$	**$R^{|}{}_9$**	...

Solving Linear NoGo – New Results

- 12 new results – 1×28 to 1×39
- Black wins by the B1 opening move

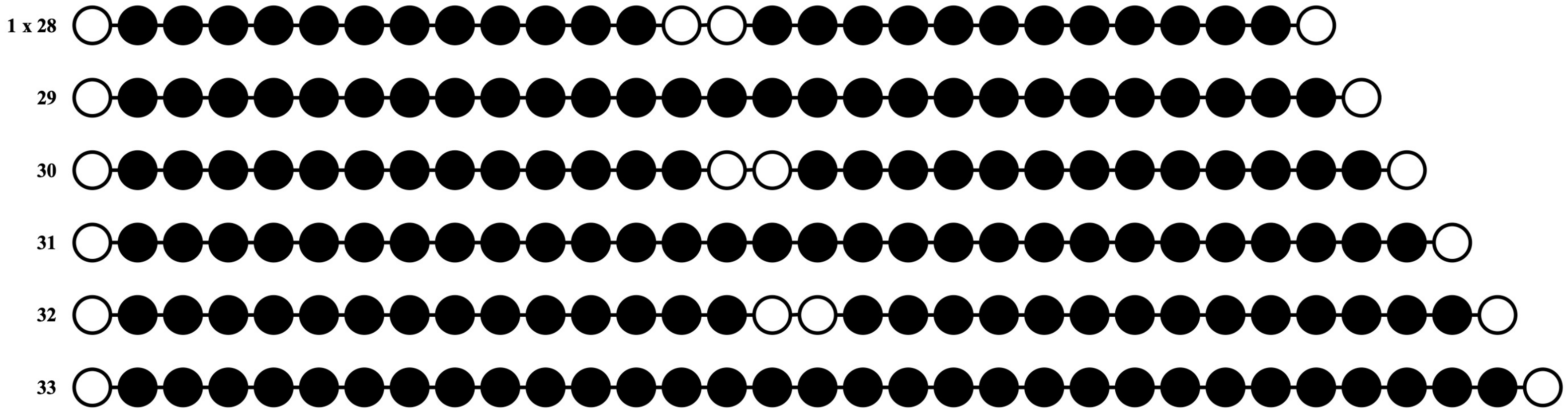
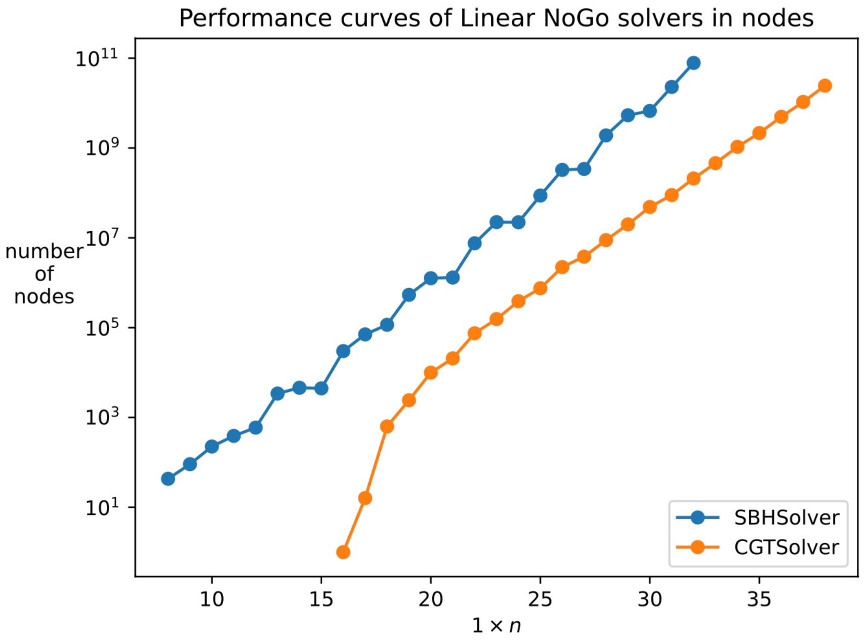
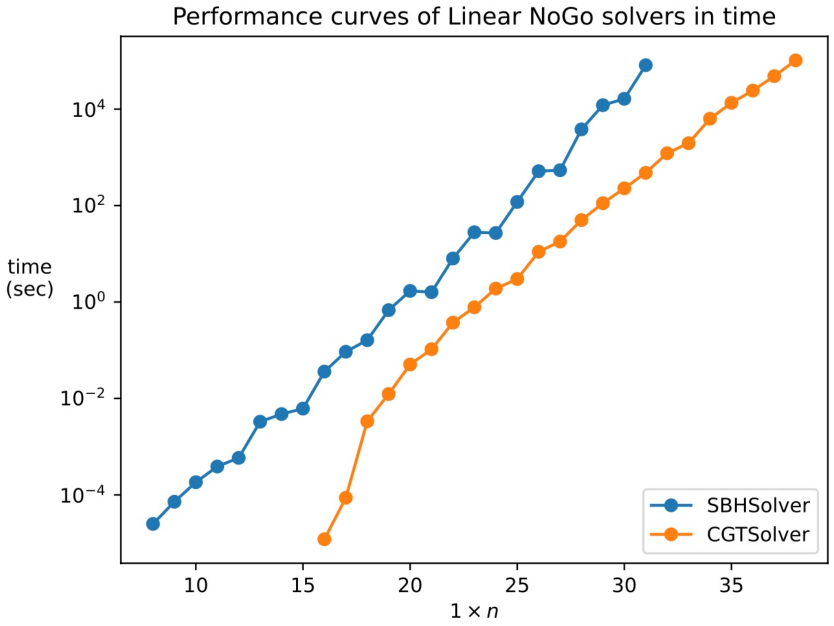


Fig. 3: The opening moves on empty $1 \times n$ NoGo with $28 \leq n \leq 33$. A black (white) stone indicates a winning (losing) opening move.

Solving Linear NoGo



(a) Node counts.



(b) Solving time.

Figure 1.2: Performance comparison of SBHSolver and CGTSolver for solving Linear NoGo with the B1 opening.

Ablation Study

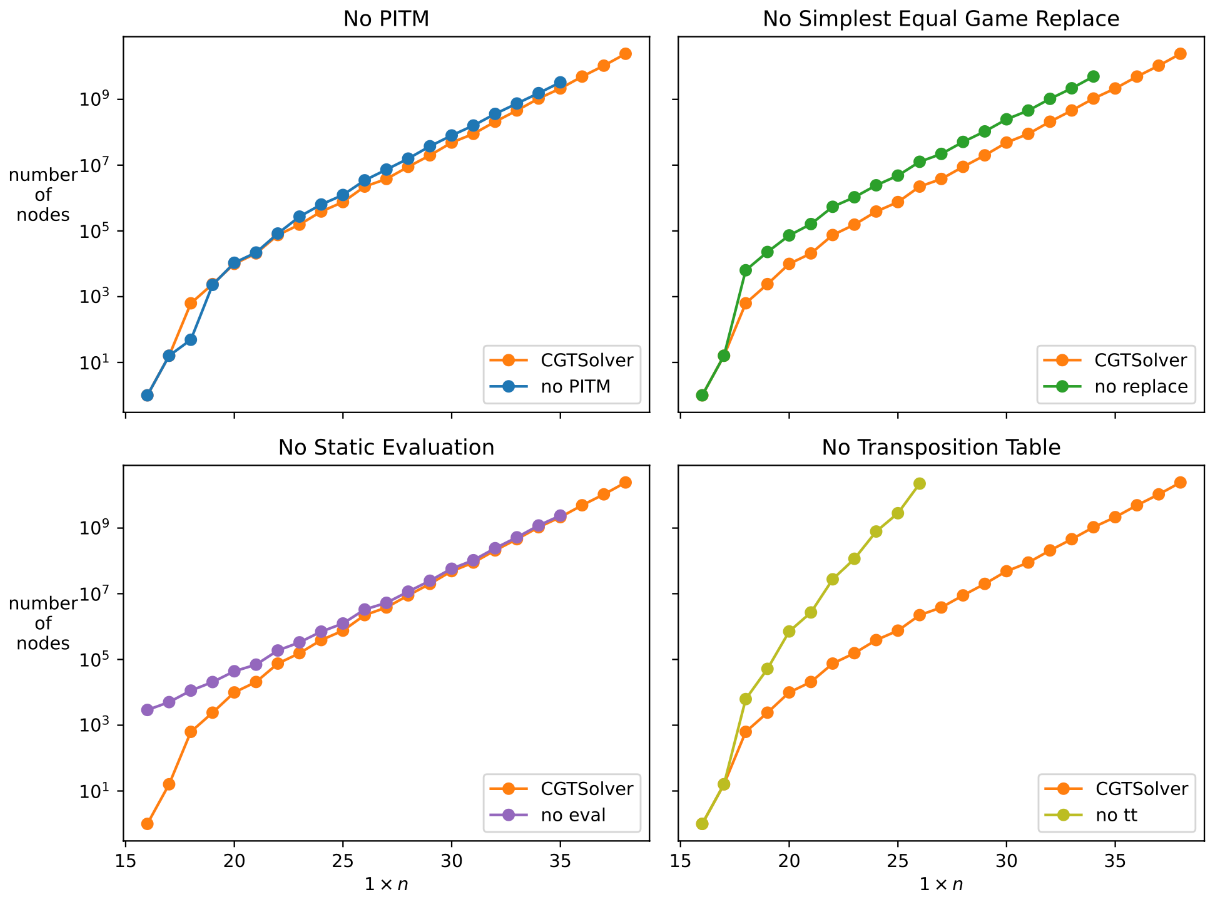


Figure 1.4: Ablation on components of CGTSolver: the PITM heuristic, replacement with simplest equal games, static evaluation, and transposition table (TT).

Statistics on Search Tree and Subgames

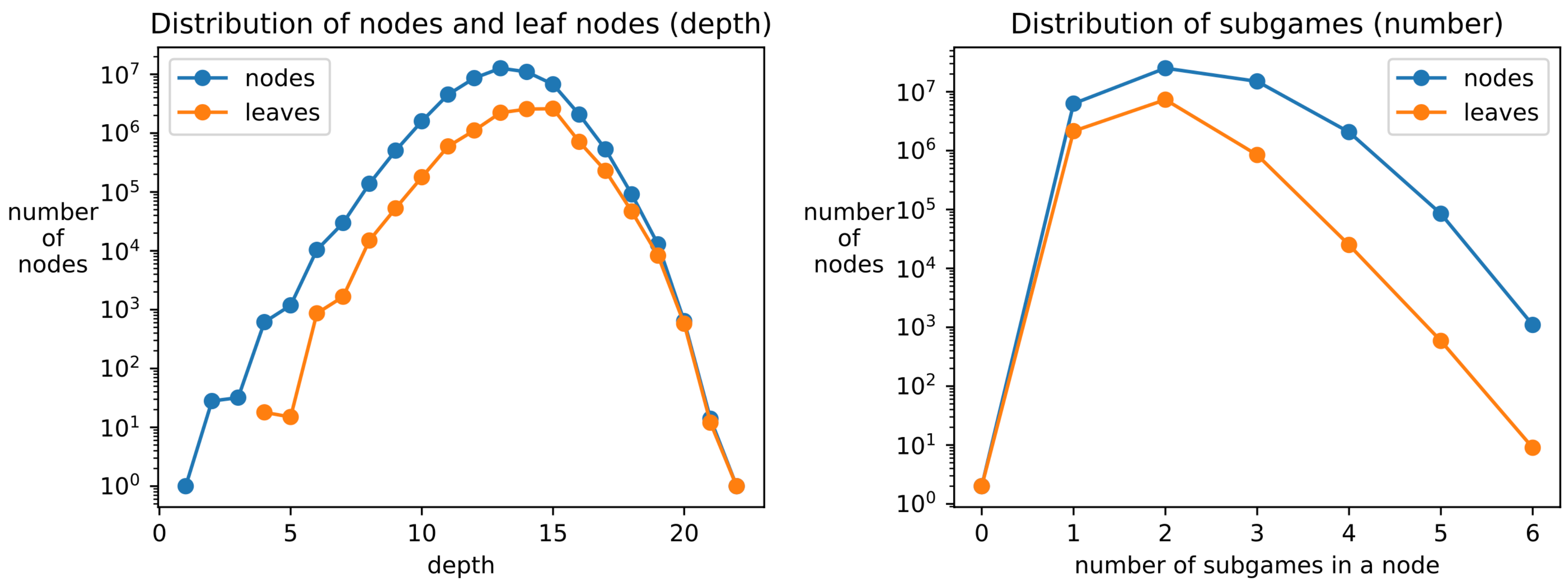


Fig. 5: Statistics in solving 1×30 NoGo with B1 opening: the number of nodes and leaf nodes across depths, the number of nodes and leaf nodes having different numbers of subgames, and the number of subgames of different sizes in all nodes

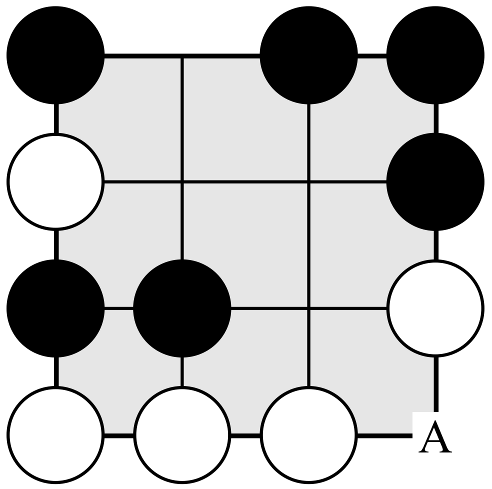
Avoid Eye-Filling Moves

Conjecture 1. In Linear NoGo, if the current player can win, and if there exist non-eye-filling legal moves, then at least one of those non-eye-filling moves is a winning move.

Remark. Complete database for $n \leq 25$. No counterexamples.

Avoid Eye-Filling Moves

- Conjecture does not hold for 2D NoGo.



Avoid Eye-Filling Moves

- It is still a good strategy.

Color	1-Go	Win by 1-Go	Eye	Win by Eye	
B	1212110	22912	431614	648	0.15%
W	1193208	24000	329480	1320	0.4%

Table 1.2: In 4×4 NoGo games, the number of game positions where there exist moves both players can play and moves only the current player can play (1-Go) or, more strictly, eye-filling (Eye); the number of positions that the current player **only** wins by playing those moves.

Incentives

- A measure of how much the player gains by making a move

- For Left (Black), $G_i^L - G$
- For Right (White), $G - G_i^R$

Incentives

Property 1. In Linear NoGo, there exist eye-filling moves that have an incentive worse than -1.

$$G = \texttt{.x..x.x.o.} = \{2* \mid 1*\}$$
$$G^L = \texttt{.x..xxx.o.} = \{1* \mid 0, *\}$$
$$G^L - G < -1$$

Incentives

Property 2. In Linear NoGo, there exist eye-filling moves that have an incentive better than -1.

$$G = \texttt{..x.x.o.x} = \{1* \mid *\}$$
$$G^L = \texttt{..xxx.o.x} = \{* \mid 0\}$$
$$G^L - G > -1$$

Incentives

Property 3. In Linear NoGo, there exist non-eye-filling moves that have an incentive of -1.

$$G = \texttt{x..x} = 1$$

$$G^L = \texttt{xx.x} = 0$$

$$G^L - G = -1$$

Incentives

Property 4. In Linear NoGo, there exist non-eye-filling moves that have an incentive worse than -1.

$$G = \texttt{.x.o..o..o} = \{0, \{1 \mid 0\} \parallel * \mid -1*\}$$
$$G^L = \texttt{.x.o.xo..o} = -1*$$
$$G^L - G < -1$$

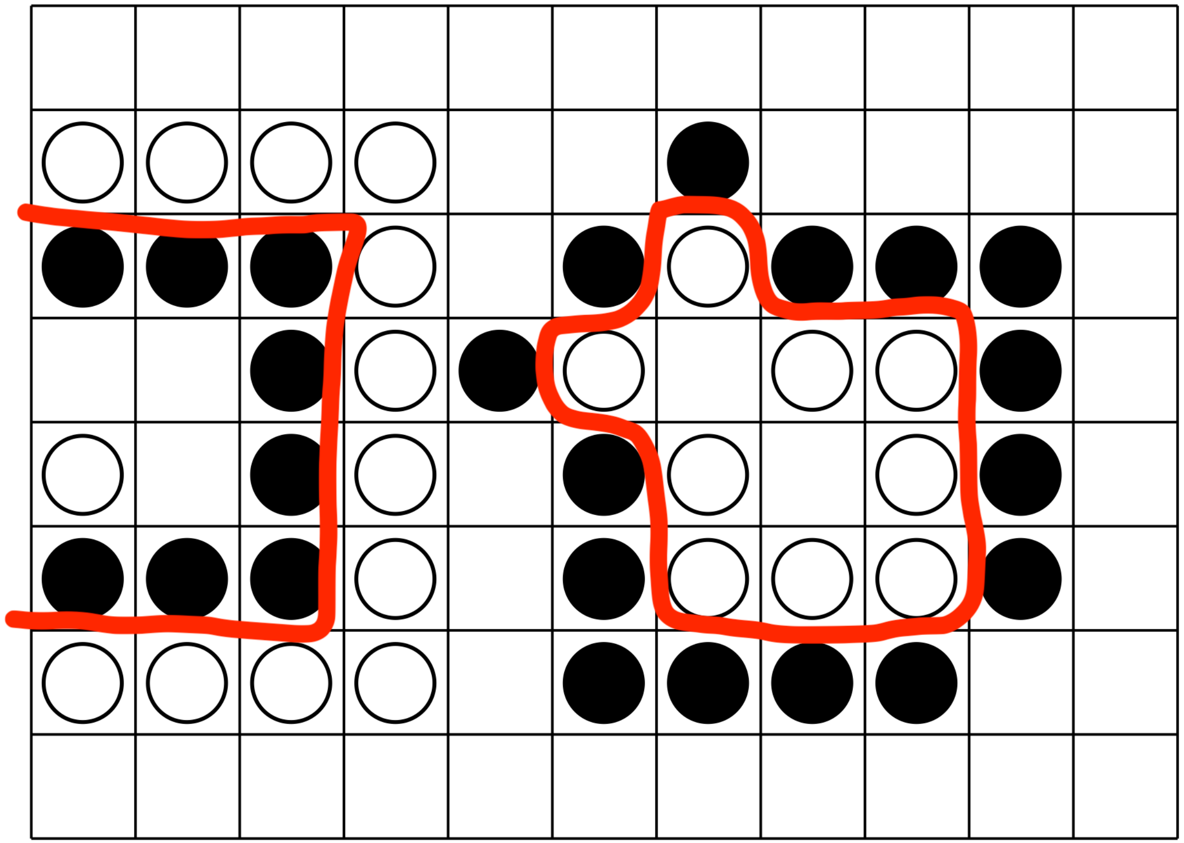
Conclusions on CGTSolver

- A CGT-enhanced solver
- Block simplification
- Split

- 12 new results on solving Linear NoGo: board size of 28 to 39
- Avoid Eye-Filling Moves
- Incentives

MCGS & 2D NoGo

xo-Split



Wall-Split

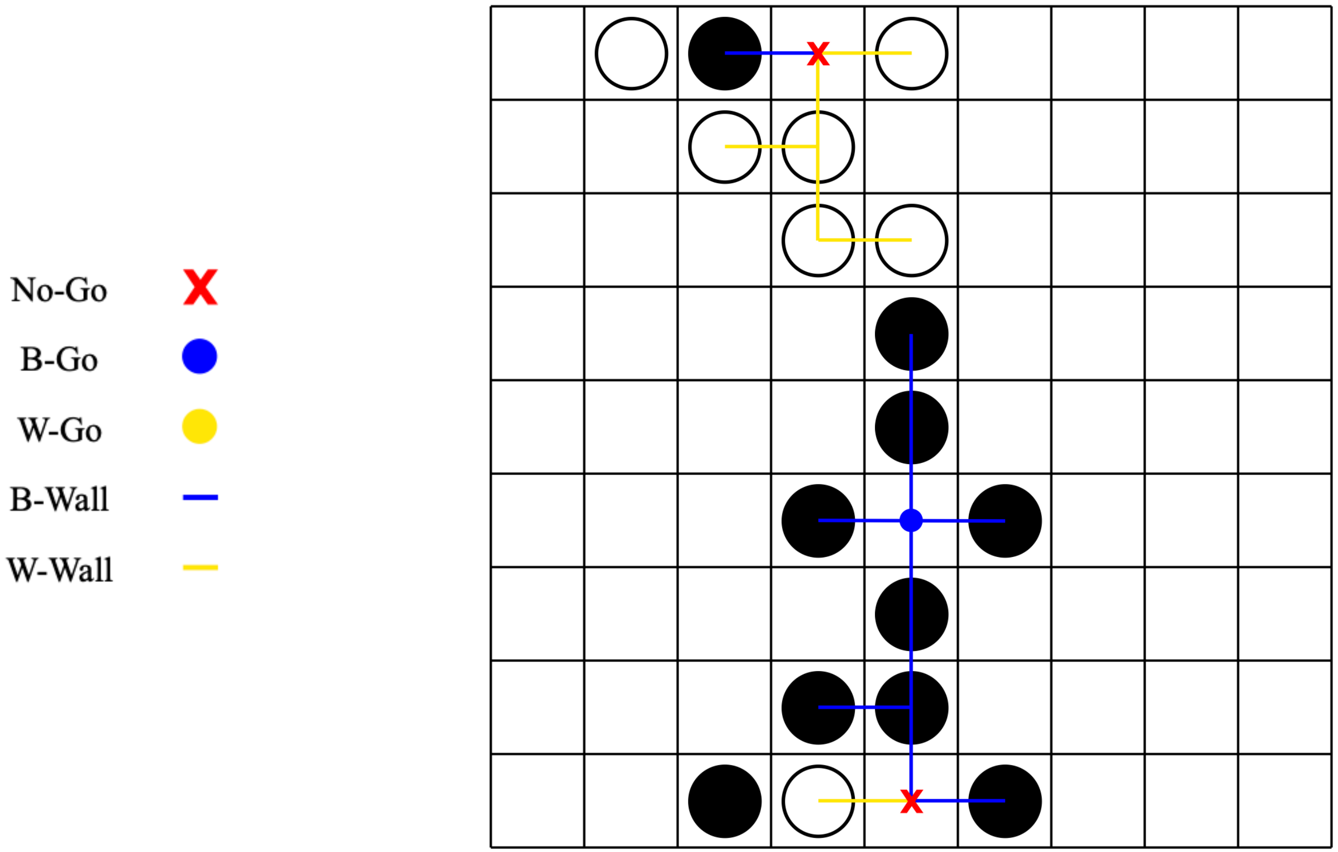
- Shan, Y.C.: Solving games and improving search performance with embedded combinatorial game knowledge. Ph.D. thesis, National Chiao Tung University (2013)

Wall-Split

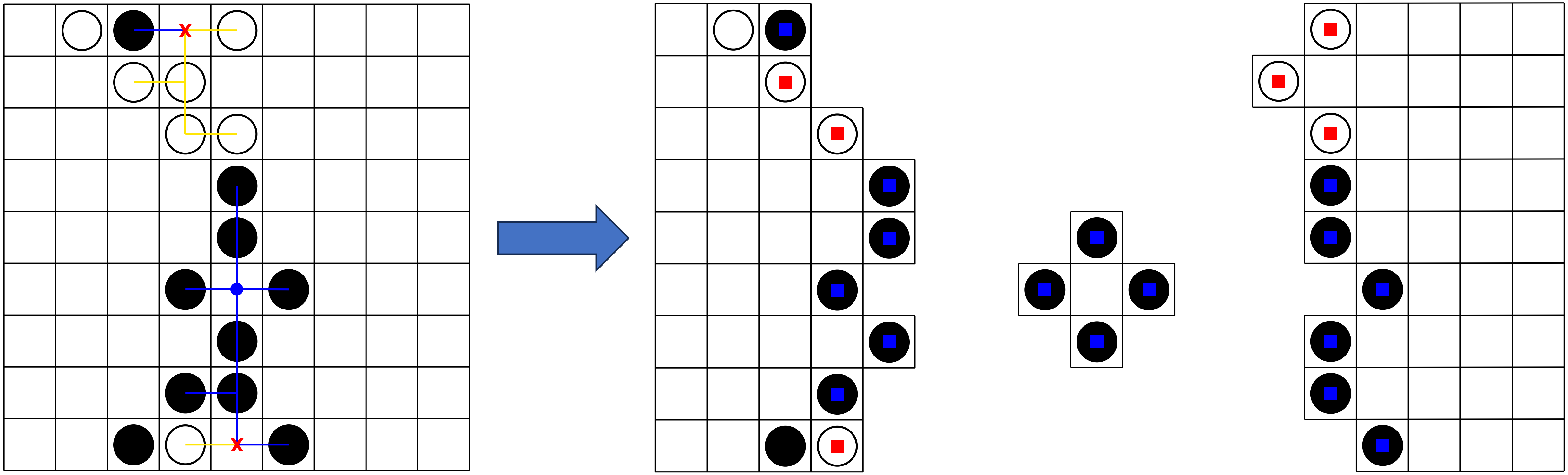
- No-Go: no player can play on
- 1-Go: only one player can play on
- B-Go: only Black can play on
- W-Go: only White can play on

- Wall: a set of stones of a color that can connect to a No-Go through stones or 1-Go’s of the same color.

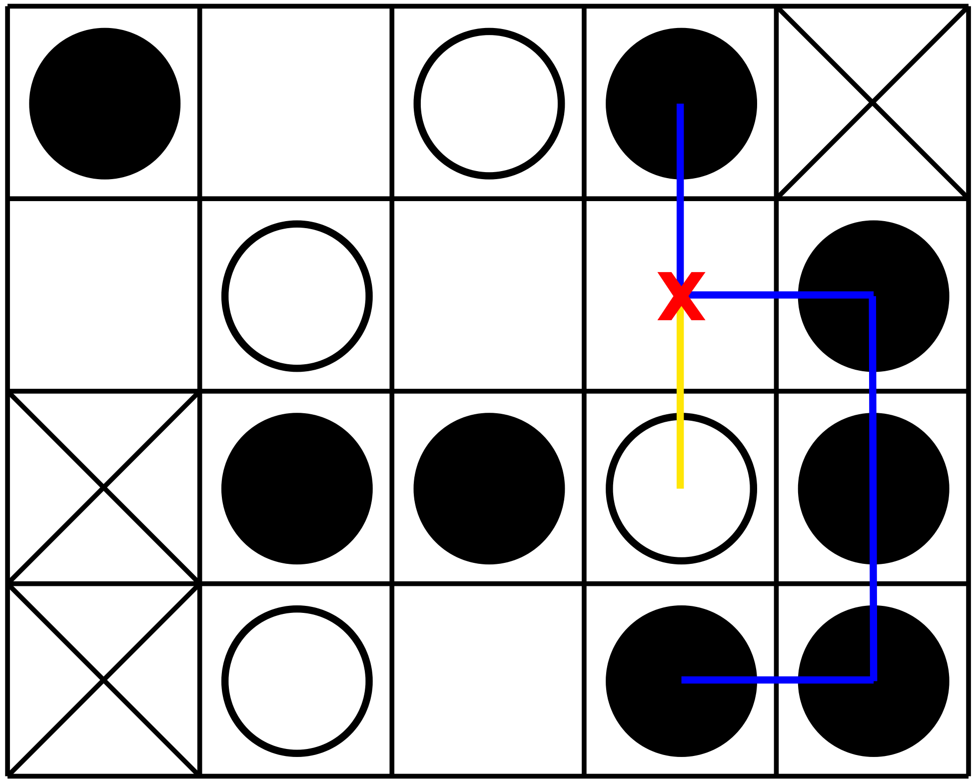
Wall-Split



Wall-Split



Wall-Split



Occurrences of Simplest Equal Games in DB

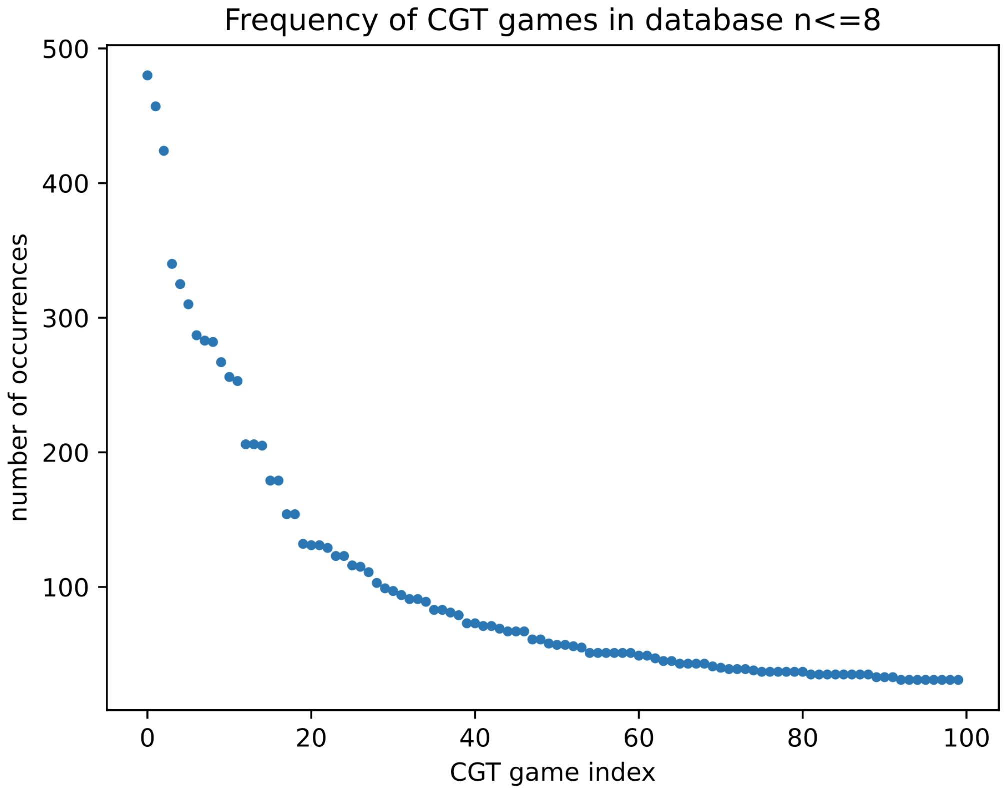


Fig. 6: The number of occurrences of the most frequent simplest equal games in the database.

idx	board	canonical form	occurrences
0	`.x.x.o.`	$1*$	481
1	`..`	$*$	457
2	`...`	± 1	424
3	`...x.o.`	$\pm(1*)$	341
4	`..x.x.`	$\{2\|1\}$	326
5	`.x.`	1	310
6	`.x.x.x.`	3	287
7	`.x.x.x.o.`	$2*$	284
8	`.`	0	282
9	`..x.`	$\{1\|0\}$	268
10	`.x.x.`	2	256

Table 1: The 11 most frequent simplest equal games in the $n \leq 8$ database.

*inverses removed

Simplest Equal Games

Due to computation constraints,

Full simplest equal games up to $n = 8$

Partial for $9 \leq n \leq 15$

idx	board	canonical form	occurrences
0	.x.x.o.	$1*$	481
1	..	$*$	457
2	...	± 1	424
3	...x.o.	$\pm(1*)$	341
4	..x.x.	$\{2\|1\}$	326
5	.x.	1	310
6	.x.x.x.	3	287
7	.x.x.x.o.	$2*$	284
8	.	0	282
9	..x.	$\{1\|0\}$	268
10	.x.x.	2	256

Table 1: The 11 most frequent simplest equal games in the $n \leq 8$ database.

Comparison with CGSuite

- CGSuite is not efficient for solving Linear NoGo
- Needs to compute canonical forms

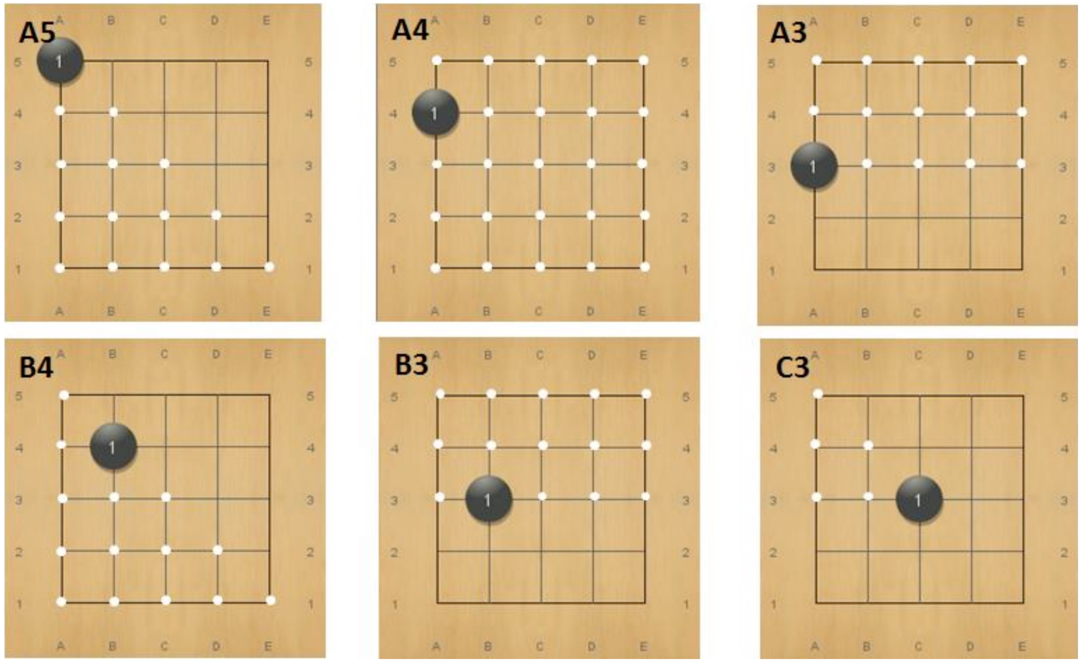
- 1×16 board has a massive canonical form with 1,201,194 stops!
- CGTSuite took over 8 minutes to solve.
- CGTSolver, without database, took less than 30 milliseconds.

Previous Work – NoGo

- She, P.: The design and study of NoGo program. Master's thesis, National Chiao Tung University (2013)
- Cazenave, T.: Monte carlo game solver. In: Monte Carlo Search, MCS 2020. Communications in Computer and Information Science, vol. 1379, pp. 56–70 (2021)

Previous Work

- Pohsuan She, 2013
- 5x5 NoGo
- All 6 openings are wins for Black



She, P.: The Design and Study of NoGo Program. Master’s thesis, National Chiao Tung University (2013)

Previous Work

- Tristan Cazenave, 2020
- Up to 25 points

	1	2	3	4	5	6	7	8	9	10
1	2	1	1	2	1	1	1	1	1	1
2	1	1	2	2	1	1	1	1	2	2
3	1	2	1	2	1	1	1	1		
4	2	2	2	2	1	1				
5	1	1	1	1	1					
6	1	1	1	1						
7	1	1	1							
8	1	1	1							
9	1	2								
10	1	2								

Table 3: Winner for Nogo boards of various sizes

Cazenave, T.: Monte carlo game solver.
In: *Monte Carlo Search, MCS 2020*. (2021)

Previous Work – CGT Solvers

- Müller, M.: Decomposition search: A combinatorial games approach to game tree search, with applications to solving **Go endgames**. In: IJCAI. (1999)
- Song, J., Müller, M.: An enhanced solver for the game of **Amazons**. IEEE Transactions on Computational Intelligence and AI in Games 7(1), 16–27 (2015).
- Uiterwijk, J., Griebel, J.: Combining combinatorial game theory with an α-β solver for **Clobber**: theory and experiments. In: BNAIC 2016. (2017)

Previous Work – NoGo

- Shan, Y.C.: Solving games and improving search performance with embedded combinatorial game knowledge. Ph.D. thesis, National Chiao Tung University (2013)