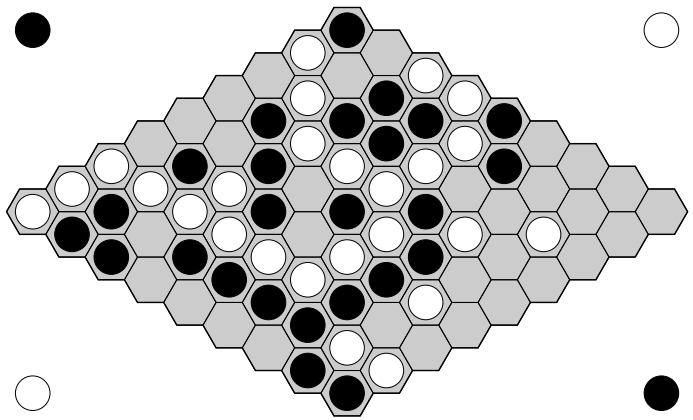
Hex, braids, the crossing rule, and XH-search

Philip Henderson

Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada

Joint work with Broderick Arneson and Prof. Ryan Hayward

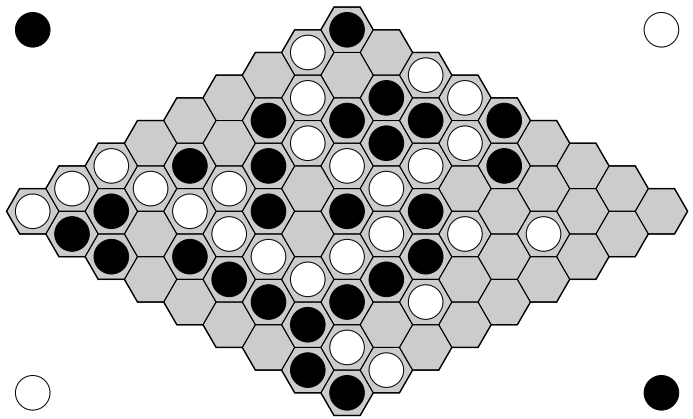
Hex Basics



Rules

- Two players alternate turns playing on any empty cell
- Stones are permanent (no moving or capturing)
- Goal is to connect your two sides of the board

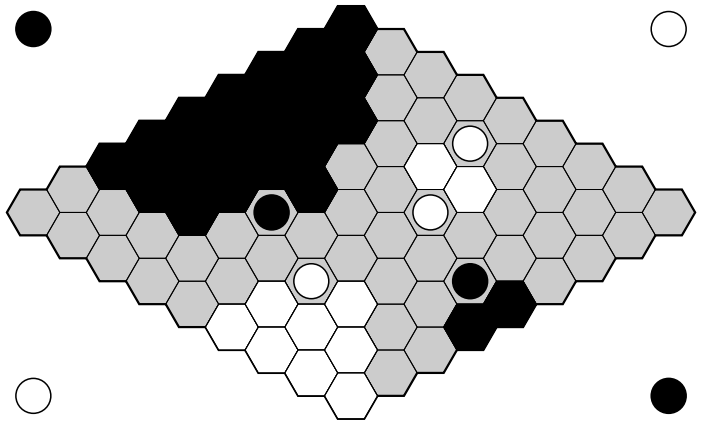
Hex Basics



Theoretical Properties

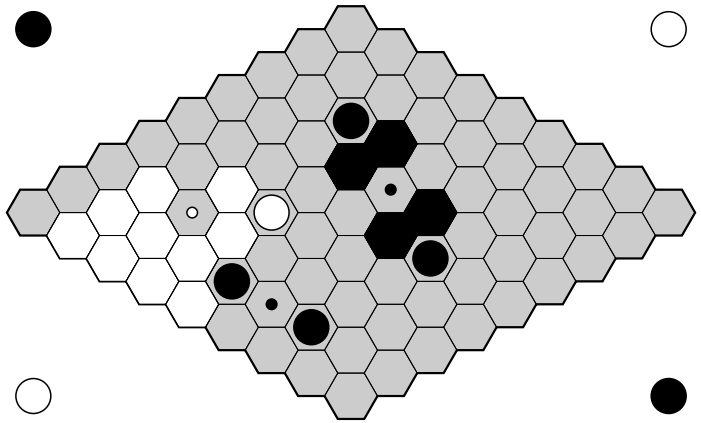
- An extra stone of your color is never a disadvantage
- Hex cannot end in a draw
- First-player win: strategy-stealing argument
- PSPACE-complete to determine winner in arbitrary position

Connection Strategies



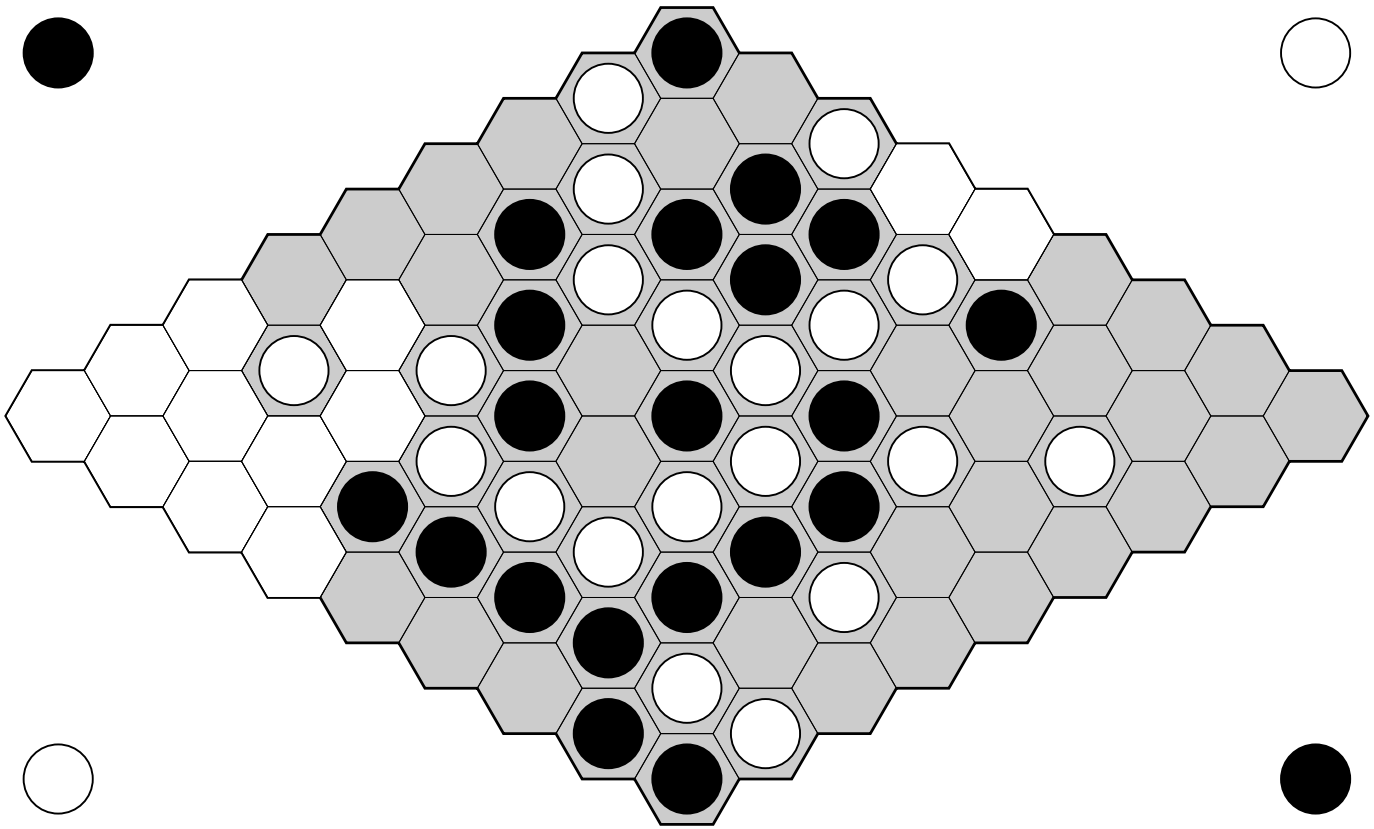
- Point: empty cell or chain of stones
- Virtual connection (VC): second-player strategy to connect two points
- Semi-connection (SC): first-player strategy to connect two points
- Carrier set: empty cells needed to carry out strategy
- Key: first move in SC

H-search

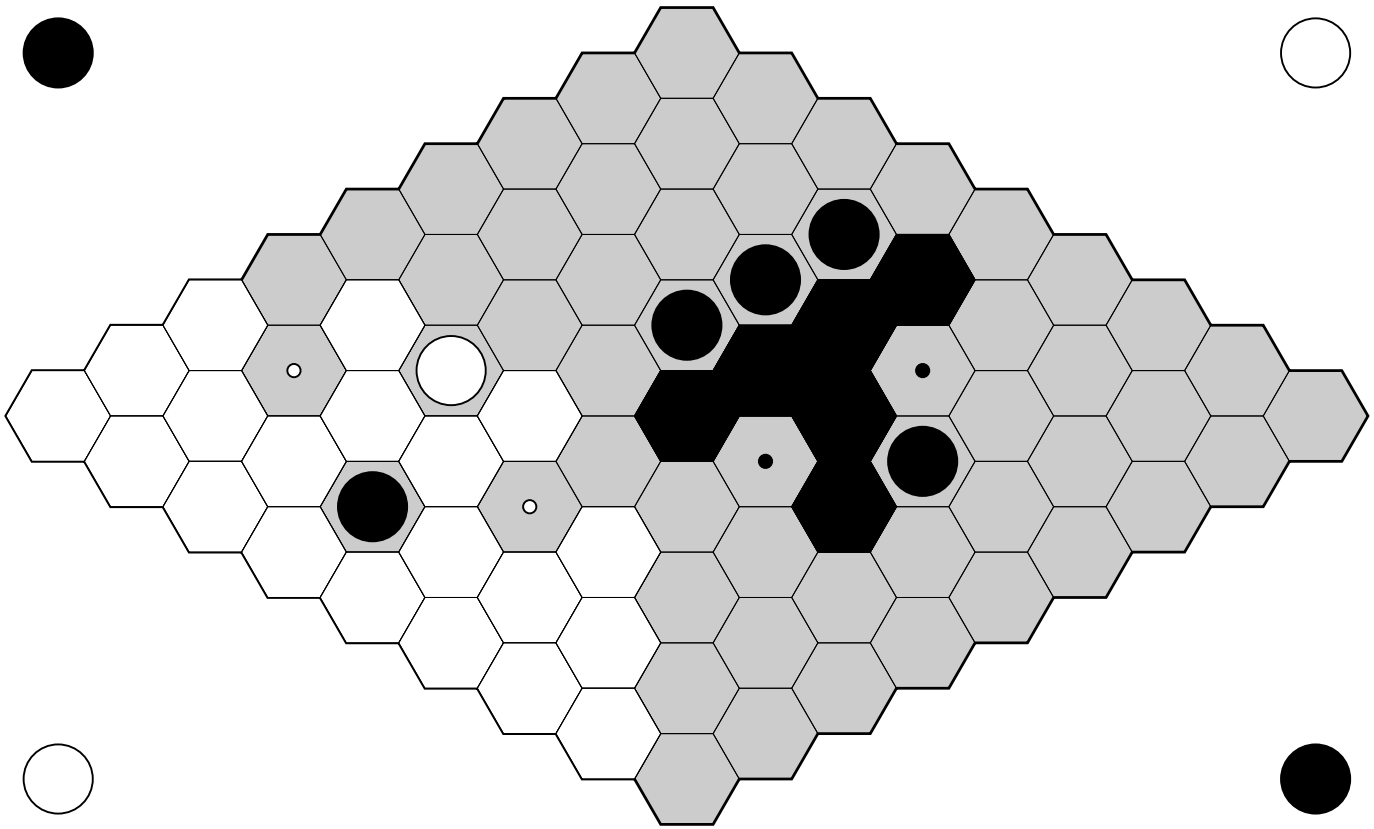


- Algorithm to identify VCs and SCs in a Hex position
- Builds connections in hierarchical fashion
- AND-rule: combines VCs in series
- OR-rule: combines SCs in parallel

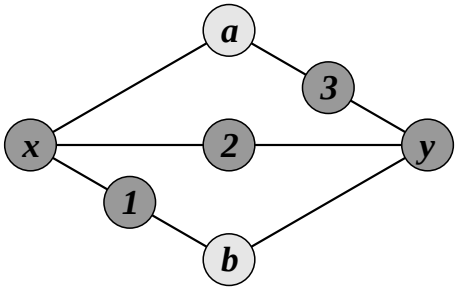
AND-Rule



OR-Rule

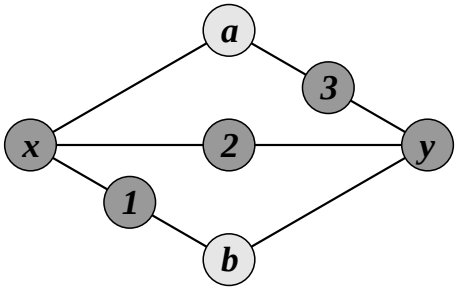


Braids



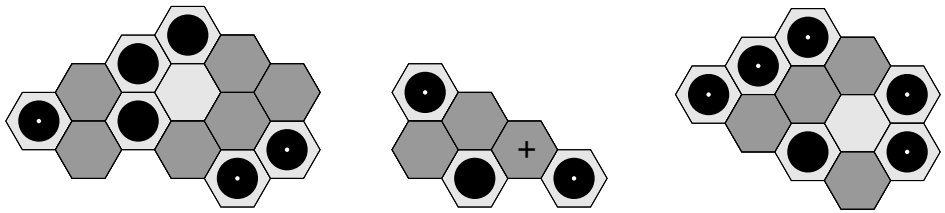
- H-search is incomplete
- Braids: SCs of above form
- Goal: extend H-search to identify braids

Braids



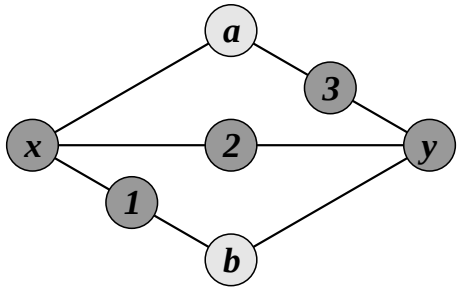
- Problem: Naive computation far too slow
- If endpoints are chains, then three SCs connect x to y
- Need to identify chains that partition connection strategies

Stepping Stones



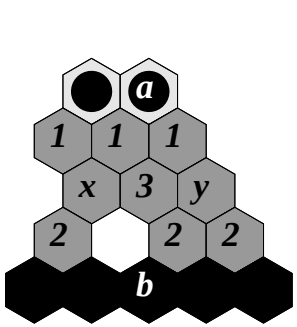
- Chains that partition a connection
- Compute recursively within H-search
- No change in computational complexity of H-search

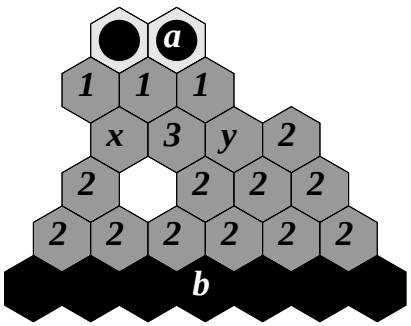
Crossing Rule



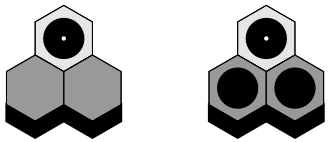
- **Crossing Rule:** Given two empty cells x and y with three pairwise disjoint SCs connecting them, such that two of the SCs have distinct stepping stones, then there exists an SC connecting these two stepping stones whose carrier is the union of the three SC carriers, and whose key can be either x or y.

Crossing Rule Examples



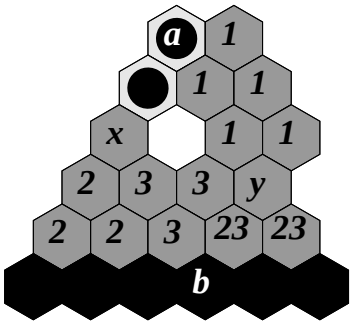


Strong Crossing Rule

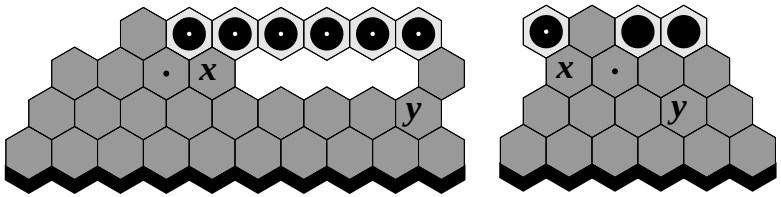


- Inferior cell analysis: can add stones without changing value of position
- If playing key x (or y) fills-in cells, SCs can overlap on that region
- **Strong Crossing Rule:** Apply edge bridge fill-in to crossing rule
- **XH-search:** H-search extended with the Strong Crossing Rule

Strong Crossing Rule Examples (1)



Strong Crossing Rule Examples (2)

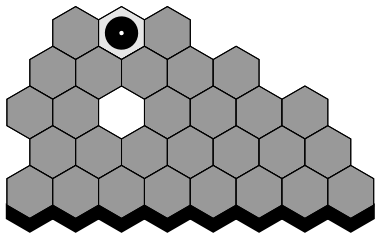


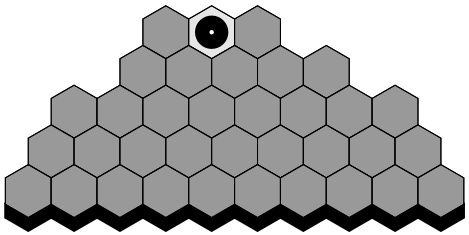
Analysis

Deduction rule	running time
AND rule	$n^3 l_V^2$
OR-3 rule	$n^2 l_S^3$
OR-4 rule	$n^2 l_S^4$
crossing rule	$n^2 l_S^3$
naive rule	$n^4 l_V^2 l_S^3$

- Stepping stones greatly reduce computational complexity
- Stepping stones only indicate existence of partition
- Crossing rule deduces orthogonal connections

Application





- XH-search still not complete
- Most new connections found near an edge
- XH-search 20% slower than H-search on average
- MoHex: 10% slower, small ELO gain, more efficient strength gain than rollouts

Summary

- Efficiently extended H-search with new deduction rule
- Identified method to combine inferior cell analysis with XH-search
- Open Question: Can the crossing rule be extended efficiently?

Any Questions?

Thanks to:

- NSERC and iCORE for funding
- Referees for helpful comments